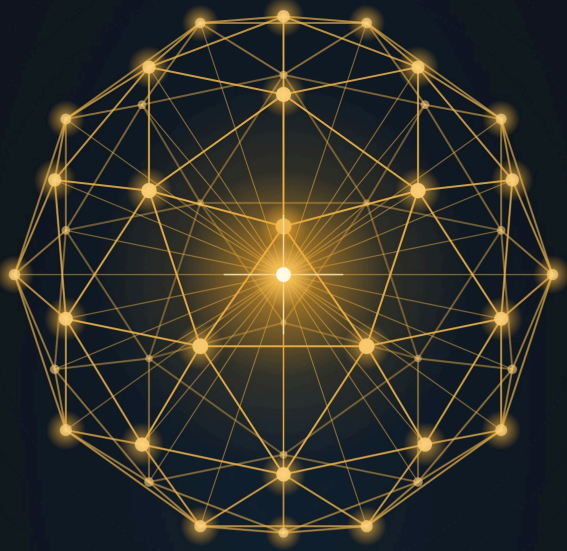
BUILDING THE CIVIC LAYER
OF A TRUSTWORTHY INTERNET

THE LAYERED WEB

THE DESIRABLE PROPERTIES OF A META-LAYER



FOREWORD BY VINT CERF

PEOPLE
Empowered communities

TRUST
Transparent & accountable

COORDINATION
Shared intelligence

REGENERATION
Sustainable futures

META-LAYER INITIATIVE

THE LAYERED WEB

THE DESIRABLE PROPERTIES OF A META-LAYER

With a foreword by Vint Cerf

VERSION 1.0

META-LAYER INITIATIVE

The Layered Web: The Desirable Properties of a Meta-Layer

Version 1.0 · November 13, 2026

Developed in the open by the Meta-Layer community through the Desirable Properties Challenge and its workgroups, and approved by the Desirable Properties steering committee.

The online edition: book.desirableproperties.org
The Desirable Properties Challenge: desirableproperties.org
Workgroups and drafts: interfacehub.net

© 2026 the contributors to the Desirable Properties of a Meta-Layer. This book is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0): you may share and adapt it for any purpose, provided you give appropriate credit and indicate if changes were made.

The chapters of this book are inscribed on Bitcoin as part of a BRC333 digital monument. The online edition is the living reference; this PDF is a snapshot of Version 1.0.

Contents

Foreword by Vint Cerf vii

About This Digital Monument xiii

Introduction 1

THE DESIRABLE PROPERTIES

DP1 Federated Authentication & Accountability 23

DP2 Participant Agency and Empowerment 63

DP3 Adaptive Governance Supporting an Exponentially Growing Community 91

DP4 Data Sovereignty and Privacy 113

DP5 Decentralized Namespace 167

DP6 Commerce 199

DP7 Simplicity and Interoperability 227

DP8 Collaborative Environment and Meta-Communities 253

DP9 Developer and Community Incentives 287

DP10 Education 315

DP11 Safe and Ethical AI 349

DP12 Community-based AI Governance 383

DP13 AI Containment 409

DP14 Trust and Transparency 433

DP15 Security and Provenance 453

DP16 Roadmap and Milestones 477

DP17 Financial Sustainability 503

DP18 Feedback Loops and Reputation 525

DP19 Amplifying Presence and Community Engagement 561

DP20 Community Ownership 595
DP21 Multi-modal 619
DP22 Civic Memory & Epistemic Continuity 653
DP23 Universal Participation & Linguistic Interoperability 695
DP24 Cross-Layer Resilience and Compromise Containment 755

Acknowledgements 785

Every chapter is also published online at book.desirableproperties.org, where it can be annotated and discussed.

Foreword

People who run around using phrases like "words of wisdom" always worry me. I have words, but whether they're wise or not is someone else's decision. What I am happy to share is a little of the experience of watching the Internet begin and then blossom, and to try to extract from that some insight into what makes infrastructure possible.

I think we have to agree that this proposal is indeed a broad kind of infrastructure. It is layered on top of the existing Internet, or whatever the Internet morphs into. When you create this sort of overlay, it's important that you use the words *agnostic*, or *technology agnostic*. The Internet Protocol layer is insensitive to the underlying transport, and that was by choice. We didn't care whether it was radio, optical fiber, a dedicated piece of copper or something else. That was helpful: when a new transport or transmission came along, we could simply sweep it in. The IP layer doesn't demand very much. It basically says: can you deliver this bundle of bits with some probability greater than zero?

So this new layer of functionality should be seen as infrastructure, with all of the implications of that.

One implication is standards. It would be hard to overemphasize their importance in an enterprise of this kind, because you want things to interwork, and not everyone who participates will be able to negotiate directly with every other party they might interact with. Standards create an interoperability that would be impossible to attain if you had to have pairwise agreements on how things work. They also allow parties to create competing products and services with consistent properties, so people have choice. And we want broad adoption, because without it, infrastructure

doesn't do very much good. Its benefits are directly related to how broadly it's adopted and how widely it's used.

Another lesson we learned very quickly is that there need to be self-sustaining business models. When we started, governments paid for all of it, and the only people allowed to use the Internet had some relationship to one of four agencies. In 1988 I requested permission to connect the MCI Mail commercial email system to the NSFNET backbone. That broke the acceptable use policy, and it was a very deliberate move on my part. In 1989 three commercial Internet service providers popped up. Some of those businesses didn't survive, but there was a business model to support the investment needed not only to spread the infrastructure but to enhance it. We will have to ask the same questions of this one.

I also want to emphasize simplicity. One thing we learned in the Internet Engineering Task Force is that if you're going to do something, try to do it only one way. As soon as you do it two ways, the two parties who want to communicate have to negotiate which way they have chosen. Minimizing complexity is our friend.

Because this infrastructure involves strong authentication and identity, and possibly a basis for transactions of real consequence, financial, medical and others, security, safety and strong authenticity are going to be essential, so that it is resistant to various and sundry forms of attack.

That concern is sharpened by artificial intelligence. My trivial example is the elevator bank where you choose your floor before you get in, and when the door opens there are no controls inside. You are completely dependent on someone else's software. That always makes me nervous, because I used to make a living writing software, or, to be really honest, writing bugs. If you worry about buggy software, you should doubly worry about buggy artificial intelligence, especially large language models that hallucinate. This infrastructure will invite the use of AI to take actions on behalf of users like you and me. That is not necessarily a bad thing; with populations declining or not growing in much of the world, there will be pressure on software to do what people once did. But that means reliability will be a high priority. Humans make mistakes. Software will make mistakes too. We should try very, very hard to reduce the likelihood of a mistake, especially where something is risky.

My two watchwords these days are accountability and agency. For many years I thought anonymity was a good thing, and I still believe that to be largely true. But the Internet and artificial intelligence have become amplifiers of everything, and the problem is the *everything* part: whether it's constructive or harmful, it gets amplified. People who cause a great deal of damage with these tools should be held accountable, and the only way to do that is to identify them. On the flip side, you don't want software, or other people, to take actions on your behalf that you didn't authorize. Strong authentication of you is in your best interest. If software is going to act for us, it must be very hard for it to exhibit bona fides implying that we authorized something we did not. This transitive authority, setting robots to work on our behalf but only doing things we've authorized, needs very careful design. And agency is about the tools we give you and the rest of the infrastructure in order to achieve those desirable properties: that limit the actions taken on your behalf to those you've actually authorized.

When Bob Kahn and I released the design of the Internet publicly in 1974, it was because we went through a little cycle. Who are our allies now? Who were our allies twenty-five years ago? A different list. Who will our allies be twenty-five years from now? We didn't know. So the only way to make sure they would have access was to release it publicly. People ask how we got away with that. We went through that analysis, we released it, and nobody noticed.

Nobody had a map of where this would go. The biggest surprise for me was when the first graphical web interface came along: the stunning fact that the Internet could be visual as well as textual. The next big surprise was the willingness of so many people to put information into the net with no pecuniary motivation. They simply wanted to know that the information they shared was useful to somebody else, and I had no anticipation of that. When the general population got on the Web, we saw a sea change in the diversity of online applications and content. Alongside the smartphone, we have seen a rapidly proliferating array of behaviors, content, incentives and side effects, most of which were not squarely on our radar fifty years ago.

There was no central authority deciding any of it. The Internet is a grand collaboration. People build pieces of it and connect them together on a voluntary basis, and they observe standards developed in a very open and coll-

aborative way. We designed it so that multiple parties could offer service. We didn't say anything about the business models, and we didn't say anything about who could connect to whom: find somebody to connect to, work out the terms, and if you follow the protocols, it will interwork.

What we did decide were its characteristics. It was not a requirement that the network support any particular application, and this is one of the big lessons for me: you don't want to over-specify a system like this. You give it some characteristics, like best-efforts communication, end-to-end recovery, some kind of flow control and the ability to route through multiple networks, but you don't assume any particular application. Because of that, people have been able to build virtually any application that can be carried as a chunk of digital bits. Every over-specification I can think of involves assumptions, and on a system with any longevity at all, those assumptions are guaranteed to be wrong. Leave a lot of room for other choices to be made, and you create a lot of flexibility. The secret sauce was making sure that anybody with a good idea could be heard and could participate.

We also left things out. Authentication is a place where we absolutely did not do very much in the original design, and it turns out that it is sometimes really important to know who you are talking to on the other end. Public key cryptography came too late for us to build into the standards; if it had been more mature, I would have liked to use it. There are lots and lots of policy issues now, of great importance, that were ignored in the Internet's early evolution.

That experience teaches me that when you're inventing something, you should try to think about what the consequences might be. The desirable properties are that thinking, done in advance: not a roadmap for what the next layer will become, which nobody can draw, but the characteristics it must have whatever it becomes.

Finally, context. I once asked a chatbot to write my obituary. It reported that Dr. Cerf had passed away, gave a date much too soon, which I didn't like very much, gave me credit for things other people did and gave them credit for things I did, and then invented family members I don't have. My cartoon answer for how that happens is that the system didn't know this paragraph was about Vint Cerf and that one was about somebody else. Until we figure out how to provide context, we are going to see these anomalies.

So, what needs to happen now is to get into the specifics, figure out what must happen before this can be implemented successfully, and ask ourselves the most important question: what could possibly go wrong? If we don't have answers to questions like that, we are not in a safe spot to implement something as significant as this infrastructure seems to be.

The Desirable Properties in this book are an attempt to answer that question. I hope you will help improve them.

Vint Cerf

About This Digital Monument

This edition of *The Layered Web: The Desirable Properties of a Meta-Layer*, Version 1.0, released on November 13, 2026, is not a conventional ebook. It is a digital monument – a civic record inscribed on Bitcoin and intended to be read by people who were not in the room when the work began, including generations not yet born.

This book exists to help define the foundational properties of a trustworthy Meta-Layer for the Internet – a global collaboration to build a human-aligned, decentralized, AI-assisted coordination layer that supports trust, agency, accountability, and human flourishing.



Figure: From today's web to the Meta-Layer. Physical Infrastructure and Internet Infrastructure carry packets and protocols. Websites & Applications are what most people experience as "the web." The Meta-Layer sits above that surface – a shared coordination layer for Trust · Context · Presence · Governance – where this book's Desirable Properties apply.

Most of us experience the Internet as pages, platforms, and feeds – a flat surface of sites and apps. The Meta-Layer is the space above that surface: not another website, but a shared digital atmosphere where context, presence, memory, and governance can persist across platforms. *The Layered Web* names the conditions that layer must satisfy if it is to serve people rather than extract from them.

That is how this monument will be used: as the open, inscribed edition of that inquiry – twenty-four Desirable Property chapters, front matter, and community patches – preserved onchain so the definitions, debates, and revisions remain visible long after any single site or sprint ends.

We are here to define the next level of the Internet. The properties refined in this challenge become the requirements and architectural decisions that guide how the Overweb – and the broader Meta-Layer – gets built.

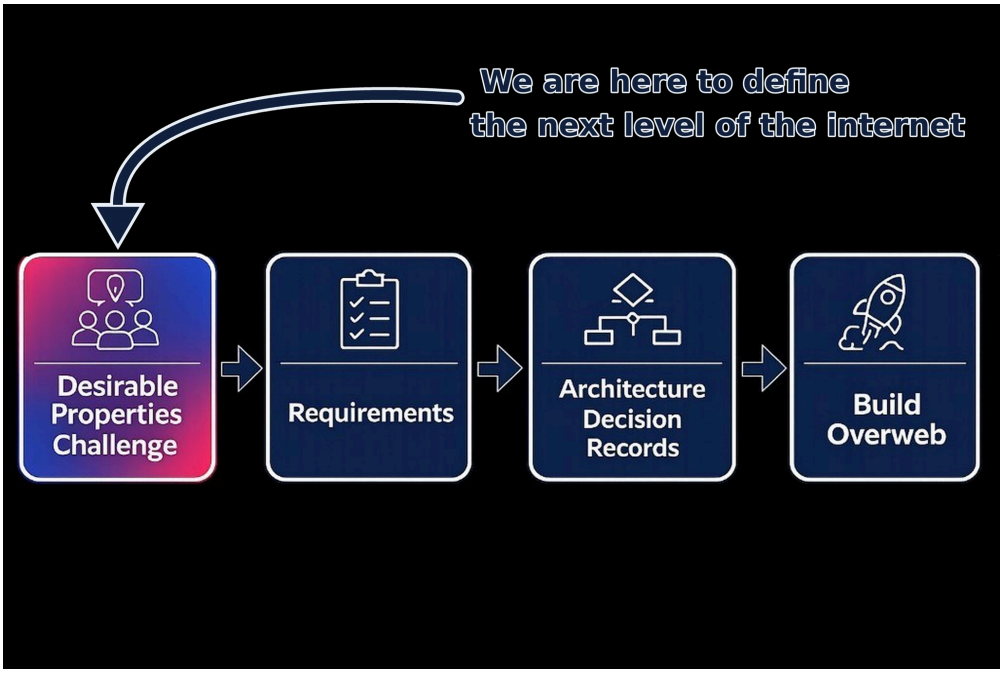


Figure: Desirable Properties Challenge → Requirements → Architecture Decision Records → Build Overweb. This inscribed edition captures Step 1: community-defined qualities a trustworthy Meta-Layer must possess before protocols are locked in.

Getting the "what" right is the community's first and most important architectural decision. The chapters here supply the vocabulary and criteria that later work will translate into formal requirements, recorded ADRs, and

eventually the Overweb itself – a coordination layer where trust, context, presence, and governance can persist across platforms rather than reset with every new app or feed.

A new information layer on Bitcoin

In late 2022, developers began exploring a practical way to store arbitrary content directly on Bitcoin. The breakthrough built on SegWit, an upgrade that added a witness field to each transaction – space that could carry data beyond simple payment instructions. In January 2023, Casey Rodarmor introduced the Ordinals protocol, which numbers individual satoshis and allows content to be written into that witness data and linked to a specific sat on the chain.

That combination opened an information layer on Bitcoin: a place where text, images, HTML, JSON, and other formats can become part of the ledger itself – not hosted on a server that can be switched off, but recorded in blocks that replicate across a global network.

What are Ordinals?

An ordinal (also called an inscription) is content embedded in a Bitcoin transaction and tied to a numbered satoshi. Each inscription receives a stable identifier derived from the transaction that created it. Indexers and readers (such as ordinals.com) resolve that identifier to the bytes stored onchain.

Ordinals turn Bitcoin from a ledger of value alone into a ledger of value and information. A chapter, a cover image, a configuration file, or an entire interactive application can exist as an inscription – readable for as long as the network is maintained and the data is mirrored by stewards who care about preservation.

Ordinals do not replace Bitcoin's role as money. They add a durable publishing surface – one suited to archives, monuments, and works meant to outlast any single company or domain name.

The BRC333 Digital Monument protocol

This book is built with the BRC333 protocol – specifically the BRC333-BOOK model for living documents, within the broader digital monument pattern for works intended as enduring public record.

A BRC333 monument is organized as a satplication: a graph of dedicated satoshis, each assigned a role in the work – configuration, sources inventory, layout shell, shared scripts, oracle data, and individual chapter rails. Content on a sat can be reinscribed when the community updates a chapter or policy, preserving history while allowing the monument to evolve in the open.

That structure is what lets this book behave like a living edition rather than a frozen PDF. The Desirable Properties Challenge, workgroups on interfacehub.net, and future reinscriptions can refine the text while the chain retains a trace of how the work changed over time.

Press `i` while viewing any chapter to inspect inscription metadata and explore the satplication graph that connects this monument's parts.

Intended to last

We inscribe these chapters, acknowledgements, and supporting artifacts onchain – and design them to remain readable through forthcoming preservation methods as indexing, mirroring, and cultural stewardship practices mature.

No storage system comes with absolute guarantees. Bitcoin's social layer must continue to honor and replicate the chain; indexers must continue to serve content; future readers may rely on methods we cannot yet name. What we can commit to now is intent: this monument is meant to be an enduring digital artifact, not ephemeral web copy.

We expect future generations to encounter this work – to judge what proved durable, improve what proved incomplete, and continue the inquiry into desirable properties before immutable protocols settle the shape of the Meta-Layer.

Explore further

- Desirable Properties Challenge: desirableproperties.org
- Open collaboration: interfacehub.net
- BRC333 protocol: brc333.xyz

INTRODUCTION

The Desirable Properties of a Meta-Layer

Purpose of This Chapter

This opening chapter frames the Desirable Properties (DPs) as a shared inquiry into what distinguishes a layered Web from today's Web. When we refer to the Meta-Layer, we mean a coherent layer of community-governed overlays that enable people and agents to safely meet, interact, and collaborate with greater trust, consent, and context. The Meta-Layer can be understood as a distributed protocol for collaboration and interaction around digital assets and overlays, rather than a single platform or service. Because such a layer must be built deliberately and stewarded over time, a properties-based approach is appropriate. The Meta-Layer represents a qualitative shift in coordination, and its most significant upside is not infrastructure alone, but the possibility of collective intelligence at scale.

This chapter is intentionally non-normative, but also precise in terms of desired outcomes. It provides orientation, metaphor, and context rather than requirements or specifications. It also situates the Desirable Properties work within a longer intellectual lineage concerned with augmenting human and collective intelligence.

1. Opening the Horizon: Engelbart and the Possibility of Collective Intelligence

Douglas Engelbart is one of the most consequential and yet persistently misunderstood figures in the history of modern computing. He was a systems thinker whose work anticipated the web, collaborative computing,

and what we now call collective intelligence. Yet he is most commonly remembered for inventing the mouse, a reduction that is akin to crediting the inventor of the automobile with designing only the steering wheel.

This narrowing of Engelbart's legacy obscures the true ambition of his work. Engelbart was concerned with how humanity might improve its collective ability to understand complex problems, make better decisions, and respond intelligently to accelerating change. He believed that our greatest challenges were not technical in isolation, but cognitive and organizational, and that our tools needed to be designed explicitly to augment human and collective intelligence.

The fact that we tend to pigeonhole Engelbart as a historical footnote rather than recognize him as one of the most important thinkers of the modern era says less about his relevance and more about our own difficulty in holding long-horizon visions. At a moment when societies face increasing complexity, fragmentation, and coordination failure, Engelbart's insights are urgently contemporary. To revisit his thinking now is to tap a remarkable source of intellectual guidance at a moment of intellectual and social vertigo. This framing is essential for understanding why the Desirable Properties approach begins with questions of capability and conditions, rather than immediately with technologies or standards.

1.1 The Mother of All Demos as a Mind-Opening Event

In December 1968, Douglas Engelbart did not introduce a product. Rather, he set out a horizon of expectations and possibilities.

What later became known as the "Mother of All Demos" was remarkable because it revealed an integrated system for augmenting human intellect. Engelbart demonstrated how tools, interfaces, language, and shared context could combine to radically expand what groups of people could do together.

The power of the demonstration lay in its narrative coherence. The interactions that Engelbart modeled in the presentation allowed people to see a future that had previously been unimaginable. By providing a tangible example of a *way* of interacting with technology, Engelbart accelerated collective understanding far more effectively than abstract theory or incremental deployment ever could.

The thinking required for building the Meta-Layer is like Engelbart's demonstration: rather than offering a finished system, we model concrete interactions and connective possibilities. The purpose of this exercise is to help people grasp what becomes achievable when coordination, trust, and intelligence are treated as shared conditions rather than isolated features. At the highest level, the "idea" of the Meta-Layer is the idea that these three qualities – coordination, trust, and intelligence – can become an instrumental part of our digital landscape.

1.2 Augmenting Human and Collective Intelligence

Engelbart's central insight was that intelligence is not solely an individual trait. It is shaped by the tools people use, the symbols they share, the processes they follow, and the environments in which they collaborate.

He understood augmentation as a systemic challenge. Improving keyboards or screens was insufficient if the surrounding social and cognitive structures remained unchanged. True augmentation requires co-evolving tools, practices, and culture to increase a group's ability to understand complex problems, make better decisions, and adapt over time.

Crucially, Engelbart focused on augmentation rather than automation. The goal was not to replace human judgment, but to amplify it. This distinction remains essential as artificial intelligence becomes more powerful.

1.3 From Engelbart's Vision to the Meta-Layer

The Meta-Layer can be understood as the inevitable extension of Engelbart's work. Engelbart's later writings make clear that he was primarily concerned with what he called the *augmentation system*: an evolving co-development of humans, tools, language, methods, and organizational structures designed to raise a group's ability to deal with complex, urgent problems. Like Francis Bacon's dream of a *New Organon* that would help fire a revolution in the sciences, Engelbart was pointing to a novel way of creating, growing, and sharing knowledge. This vision also resonates strongly with Teilhard de Chardin's concept of the Noosphere: a shared sphere of collective cognition and intelligence emerging through human cooperation and meaning-making.

In Engelbart's vision, improvement does not come from isolated innovation, but from increasing the *collective capability* of a community to understand its situation, coordinate action, and improve its own improvement processes. His focus on improving the process of improvement, sometimes described by Engelbart as bootstrapping, is a direct intellectual precursor to the Meta-Layer's emphasis on durable context, coordination, and learning over time. He emphasized shared conceptual frameworks, explicit linking of knowledge, traceable reasoning, and the ability for groups to build upon prior work rather than constantly starting over.

Unlike cybernetics, which seeks to automate the creation of knowledge, an Engelbart-inspired approach seeks to enlarge human capacity by making the creation of knowledge more intentional. The Engelbartian approach enables "many think at once" rather than an automated "no one thinking at all."

The Meta-Layer extends this vision to the scale and conditions of the contemporary internet. Where Engelbart focused on organizations and research communities, the Meta-Layer applies the same principles to a plural, decentralized digital environment composed of many communities, agents, and institutions operating simultaneously.

Persistent context, shared memory, accountable participation, and coordinated action are prerequisites for collective intelligence in such environments.

For example, consider a researcher examining a claim in a climate paper. In a Meta-Layer environment, highlighting that claim could surface prior critiques from other labs, the funding provenance of the study, a live thread of domain experts discussing its implications, and even a machine-readable summary of areas of consensus and disagreement. This is what "persistent context" and "shared memory" mean in practice.

These are precisely the conditions Engelbart identified as necessary for augmentation, but which today's Web largely fails to provide at scale. By contrast, encountering such a claim today typically requires opening multiple tabs, manually searching for sources, and reconstructing context in isolation. The Meta-Layer seeks to make these conditions visible, composable,

governable, and coordinative across communities without centralizing control.

In this sense, the Meta-Layer is best understood as an augmentation environment rather than a platform. It aims to raise the baseline capability of networked society itself, enabling people and communities to think together more effectively, reason across longer horizons, and coordinate responses intelligently to accelerating complexity – much as Engelbart sought to do, but now at planetary scale.

2. The Desirable Properties Challenge

2.1 Origins of the Challenge

As the Web evolved from an academic network into a planetary-scale social and economic substrate, it became increasingly clear that incremental fixes would not be sufficient to address its emerging failures. A father of the Internet, Vint Cerf articulated this challenge succinctly: if we are building a new layer on top of the Web, we must first be able to describe what makes it fundamentally different.

This question is deceptively difficult. New systems are often defined by their features or protocols, but doing so too early tends to lock in assumptions, centralize power, and narrow the design space. The Desirable Properties framework emerged as a way to resist that trap by focusing instead on the conditions that must hold for a Meta-Layer to function responsibly and at scale.

2.2 Why Properties Precede Protocols

Desirable Properties articulate what must be true without prescribing how it must be built. This distinguishes them from both requirements and principles. Requirements tend to harden prematurely into constraints, while principles often remain too abstract to guide design tradeoffs.

By contrast, Desirable Properties describe conditions that shape what becomes possible. This allows multiple implementations, governance models, and technical approaches to coexist while remaining aligned around shared outcomes. It also allows the framework to remain durable across rapid technological change.

Viewed as a whole, the Desirable Properties function as a map of dependency and emergence. Foundational properties establish the minimum conditions for safe participation. Intermediate properties enable coordination across difference, scale, and time. Higher-order properties support collective intelligence, regenerative value creation, and adaptive governance.

By grounding the Meta-Layer in properties rather than features, this work prioritizes long-term legitimacy, adaptability, and trust over short-term optimization. It treats the Meta-Layer as an evolving environment rather than a finished product.

2.3 Vint Cerf's Challenge

On September 16, 2024, the Meta-Layer Initiative convened a kickoff meeting to frame the work ahead. During this meeting, Vint Cerf articulated a clear and consequential challenge: if the community believes a new layer is emerging above today's Web, it must be able to explain what makes that layer meaningfully different in principled terms.

Cerf emphasized that the goal was not to immediately define architectures or standards, but to write an essay identifying the *desirable properties* of such a layer. These properties would serve as a way to reason about design space, tradeoffs, and legitimacy before locking in technical decisions. He stressed that this approach mirrors how the early Internet benefited from clear guiding principles long before formal protocols stabilized.

As a concrete starting point, Cerf suggested **Federated Strong Authentication** as an initial property worthy of exploration. His reasoning was that trust, identity, and accountability are foundational to any higher-order coordination. Without mechanisms that allow participants to reliably know who or what they are interacting with – across organizational and jurisdictional boundaries – more advanced forms of coordination, governance, and collective intelligence become fragile or impossible.

Importantly, the emphasis was on *federation*, not centralization. Cerf underscored that no single authority should control identity or authentication for the Meta-Layer. Instead, multiple trusted entities should be able to participate, allowing identity and trust to be established in a plural, interoperable manner consistent with the Web's decentralized ethos.

He further noted that such authentication mechanisms would need to evolve alongside new realities, including automated agents and AI systems, making clear differentiation, accountability, and binding between actors and actions increasingly important.

This kickoff reframed the work ahead. It also echoed foundational ideas from the early Internet, where layered design, federation, and trust-by-design were treated as enabling assumptions rather than afterthoughts. Rather than asking how to build a Meta-Layer, the community was challenged to first articulate the conditions under which such a layer could be trustworthy, scalable, and legitimate. The Desirable Properties effort emerged directly from this moment, with DP1 intentionally addressing the trust and authentication questions raised at the outset.

3. Two Calls for Input and a Shared Inquiry

During the first year following Cerf's challenge, the Desirable Properties work was shaped through two coordinated calls for input. These calls were designed to surface both lived experience and forward-looking concerns, recognizing that the Meta-Layer must respond to real failures while anticipating future ones.

The first call engaged communities through the People Centered Internet, the Bridgit.io network, and aligned partners. Participants included organizers, builders, and stewards already navigating trust breakdowns, abuse dynamics, and governance stress. Their input emphasized legitimacy, accountability, and the need for durable coordination mechanisms.

A second, broader call was conducted through the first generation Meta-Layer Initiative site and public channels, with AI assistance used to synthesize themes and lower barriers to participation. This surfaced concerns about AI agents, automation at scale, and new forms of manipulation and coordination failure.

To support transparency and shared sensemaking, a dedicated Desirable Properties web application aggregated submissions across both calls. Patterns emerged clearly: trust failures are systemic, coordination problems span social and technical domains, and any viable Meta-Layer must address human and AI participation together.

4. From a Single-Cell Organism to a Multilevel One

4.1 Biological Metaphor for Digital Systems

The transition from a single-cell organism to a multilevel organism offers a useful metaphor for understanding what is happening as the Web scales beyond its original design assumptions.

Single-cell organisms are highly autonomous. Each unit contains everything it needs to survive and act, and coordination is minimal because scale is limited. This simplicity is powerful at small sizes, but it does not scale. As complexity increases, single-cell systems quickly encounter limits in adaptability, resilience, and coordination.

Multilevel organisms emerge when scale demands more than autonomy alone can provide. Cells differentiate, specialize, and coordinate through shared structures and signaling systems. Control is not centralized in a single cell, but distributed across layers, enabling the organism to act coherently while remaining adaptable.

Digital systems follow a similar trajectory. Early web architectures privileged independence: pages, sites, and platforms operated largely as self-contained units. This worked when scale was limited and interactions were sparse. At planetary scale, however, autonomy without coordination produces fragmentation, brittleness, and systemic failure.

The Meta-Layer reflects a transition toward a multilevel digital organism: an environment where local autonomy is preserved, but higher-order coordination becomes possible through shared conditions, interfaces, and governance primitives.

4.2 Scale, Differentiation, and Collective Capability

Scale fundamentally changes the nature of coordination. As participation grows, it becomes impossible for all actors to relate to one another directly or to rely on informal norms alone. New roles, functions, and governance structures emerge as adaptive responses to complexity.

Single-level systems depend on uniformity and direct interaction. At small scales, this works well. At larger scales, it collapses. Differentiation bec-

omes necessary so that different functions can be handled by different actors, and so that coordination does not overwhelm autonomy.

In biological systems, this differentiation gives rise to tissues, organs, and regulatory systems. In social and digital systems, it gives rise to roles, institutions, interfaces, and governance mechanisms. These structures are signals that the system has crossed a threshold where higher-order coordination is required. The resulting ensemble can be understood as an autopoietic system, a concept from evolutionary biology and social systems theory that describes how complex systems maintain their own distinctive equilibria in response to new constraints imposed by both the environment and new states within the system.

At internet scale, simplistic models of coordination collapse. Differentiation becomes unavoidable, governance emerges, and collective capability increases. The Meta-Layer acknowledges this reality and seeks to provide shared conditions under which differentiation and coordination can occur without centralization.

4.3 Complexity, Structure, and the Expansion of Degrees of Freedom

What appears, at first glance, as increasing complexity is in fact an expansion of **degrees of freedom**. Single-cell systems feel simple because they have few possible configurations, but they are also brittle. They can adapt only within a narrow range before failing.

Multilevel organisms introduce structure, differentiation, and constraint to enlarge the possibility space. By organizing complexity into layers, roles, and relationships, they unlock vastly more ways to respond to change, absorb shocks, and coordinate action. Freedom in such systems does not come from the absence of structure, but from having *enough structure* to support many possible paths forward.

The same principle applies to digital society. Low-complexity environments appear permissive, but they severely constrain what people can do individually and what groups can do together. They limit shared reality and understanding, collapse learning across time, and force coordination into brittle, short-term patterns.

Engelbart recognized this inversion early: environments which feel simple often impoverish collective capability, while deliberately structured environments enable groups to think, learn, and coordinate in ways that would otherwise be impossible. Freedom, in this sense, is not the absence of constraints, but the presence of enabling ones.

The Meta-Layer is designed to expand this possibility space. By introducing layers, context, relational structure, and interface-level governance, it increases the degrees of freedom available to people and communities. These expanded degrees of freedom are what make coordination through complexity possible, and they set the stage for the emergence of collective intelligence.

5. From a Flat Web to a Layered World

5.1 Earth Metaphor for Digital Systems

For much of human history, people understood the Earth as flat. Over time, this view expanded to recognize a spherical planet, and later a far richer reality: a planet composed of layers, including its core, strata, soils, oceans, atmosphere, and beyond.

The Web follows a similar pattern. While it is often treated as a flat surface of pages and links, it is better understood as a layered environment with distinct functional and contextual strata.

Today's Web largely operates at the surface layer. Just as biological organisms rely on internal layers to regulate, coordinate, and sustain life, layered digital systems rely on higher-order contexts to regulate interaction, meaning, and governance. The Meta-Layer is a shared digital atmosphere that sustains crucial resources such as context, presence, memory, and meaning. While these terms may seem abstract, they acquire specific meaning in the context of specific types of digital interactions, especially those that benefit from distributed social interaction. In this sense, the Meta-Layer functions as an extension of the digital noosphere, surrounding existing content without replacing it.

5.2 Context as Infrastructure (From Contextlessness to Context)

The most underappreciated limitation of today's Web is not scale, speed, or access, but structural contextlessness. The consequences of structural contextlessness shape how trust, governance, and accountability fail across today's Web.

Each interaction largely resets. For example, a thoughtful discussion thread is flattened into isolated replies as it moves between platforms, or is revisited weeks later with no shared memory. Identity fragments across platforms when the same person must repeatedly reintroduce themselves, rebuild credibility, and renegotiate trust in each new context. Meaning seems bound to particular statements in digital locations, when in fact it is subject to negotiation across different actors and interests.

In a contextless environment, trust is fragile. Governance decisions are made without shared memory, and incidents that test accountability are fragmented in ways that make it difficult or impossible to examine full chains of causation, contributing actors, or systemic incentives. Coordination remains shallow because participants lack the durable context needed to reason collectively about patterns, intent, or history. The challenge of understanding and improving the internet increasingly resembles the problem of reducing airline or spaceflight accidents: meaningful progress depends on deep contextual knowledge of chains of causation, decisions, and actors, without which systemic weaknesses remain invisible.

The Meta-Layer introduces context as infrastructure. By allowing shared context to persist above individual pages, platforms, or interactions, it creates the conditions for memory, learning, and continuity. Context in this sense is not content. It is the surrounding fabric that allows content to be interpreted, evaluated, and acted upon coherently over time.

One way to understand durable context is as a *context package*: a well-formed, referencable body of accumulated information about participation, decisions, and meaning with provenance. Such packages are not static records. They can evolve, be tracked over time, and be selectively linked to other contexts through bridges. This allows understanding to travel without being flattened, and governance to operate on shared memory rather than isolated events.

The shift from contextlessness to structured context is foundational. It transforms trust from a one-time or binary judgment into a condition that accumulates through observable patterns of participation over time. Governance can then rely on shared memory and precedent, rather than being forced to re-adjudicate identity, intent, and legitimacy from scratch with each new incident.

5.3 Interface-Level Governance

When context persists, governance necessarily moves closer to where participation actually occurs. In the Meta-Layer, governance is not confined to backend policy documents or invisible moderation systems. It is enacted at the interface level, where people encounter rules, constraints, signals, and affordances directly.

Interface-level governance makes actors and actions visible and so accountable to reasoned communal judgment. Participants can see how trust is earned, how decisions are made, and how boundaries are enforced. Rules become contextual rather than universal, and enforcement can be proportional rather than blunt.

By operating at the interface level, the Meta-Layer does not impose governance from above. Instead, it provides the civic substrate through which legitimate governance can emerge, be contested, and adapt over time.

5.4 Connection, Relation, and Bridging as Enabling Conditions

Context does not arise automatically. It is produced through **connection and relation**: the explicit linking of people, agents, content, actions, and communities in ways that make meaning transferable rather than isolated.

The Meta-Layer therefore treats connection as an enabling condition. Relationships between entities are made explicit, durable, and governable. This includes relationships such as authorship, response, endorsement, disagreement, stewardship, reuse, and moderation.

Connecting information plays a critical role in this process. Facts or statements might be treated as atomic entities, but they have no real consequences – no real meaning – until they are gathered into meaningful groupings such as narratives or theories. This is how knowledge surfaces. Significant work is required to connect contexts across boundaries, such as

between communities, domains, disciplines, or moments in time. This connecting of information is interpretive and *civic labor* that translates meaning so it can be understood and acted upon elsewhere without being flattened or distorted.

As connections accumulate, they form a contextual graph: a living structure that links what is related, relevant, or consequential. This graph does not require global completeness. It grows locally, is composed selectively, and is governed contextually. Its value lies not in total connectivity, but in *appropriate relation*.

By enabling selective connection, durable relation, and meaningful connections, the Meta-Layer creates the conditions under which context can persist, trust can travel, and collective intelligence can emerge across scales.

6. From Low Coordination to Coordination through Complexity

6.1 The Web as a Low-Coordination Environment

Many of the most persistent failures of the modern internet are failures of coordination. The Web evolved as an environment where actors optimize locally, with no shared reputation across platforms, no portable moderation signals, and no persistent memory of behavior across contexts. Feedback loops are weak or fragmented, with enforcement and learning reset at each platform boundary, and there is little capacity for collective constraint beyond isolated, platform-specific rules.

In such environments, what is often described as the "tragedy of the commons" or the "tyranny of the commons" emerges naturally. Individually rational actions, when aggregated at scale, produce collectively destructive outcomes. Attention is overextracted through engagement-maximizing algorithms, trust is depleted as signals are gamed or siloed, discourse degrades under incentive structures that reward amplification over accuracy, and long-term value is sacrificed for short-term gain.

As Liv Boeree and others have articulated through the metaphor of Moloch, destructive equilibria arise when systems lack mechanisms for aligning in-

al boundaries, or coordinate restraint over time. In this sense, a modern Moloch is the absence of infrastructure for shared memory, portable identity, and cross-context governance.

On today's Web, platforms are siloed, governance resets at every boundary, and accountability signals do not travel with actors. A user banned in one context can reappear in another, if not the same one, with impunity. A reputation built in one space is invisible in another. Even well-intentioned actors are pushed toward extractive or adversarial behavior because the environment provides no durable way to coordinate restraint, stewardship, or long-horizon thinking.

6.2 The Meta-Layer as a Coordination Shift

The Meta-Layer represents a shift from low-coordination environments to coordination through complexity. Rather than assuming independence by default, it introduces shared conditions, such as portable identity, persistent reputation, and interface-level enforcement, that allow coordination to emerge wherever complexity demands it, without requiring constant negotiation or centralized control.

By introducing durable context, such as shared memory of actions and provenance, visible relationships, such as clear signals of who is acting and under what constraints, and interface-level governance, such as zone-based rules enforced through overlays at the point of interaction, the Meta-Layer expands the space of possible collective interactions and outcomes. Actors are no longer forced into destructive equilibria by default. Instead, they gain additional degrees of freedom: the ability to carry identity and accountability across contexts, apply different trust thresholds in different zones, require proof-of-humanity for amplification, and coordinate behavior across time horizons.

This increase in degrees of freedom is critical. Low-coordination systems appear simple, but they severely constrain what groups can do together. Coordination through complexity enables new forms of cooperation that were previously infeasible, such as cross-platform moderation signals, community-defined rules and trust zones, and shared governance over AI agents and automated behavior. It allows societies to move beyond false

tradeoffs between speed and legitimacy, autonomy and cooperation, competition and care.

This does not eliminate conflict or competition. It makes them governable. Competing actors can operate under shared rules, visible constraints, and contestable processes rather than opaque incentives and fragmented enforcement. Planetary-scale challenges, from information integrity to ecological sustainability, cannot be addressed through isolated action alone. They require environments that support coordination across difference, uncertainty, and scale, with mechanisms for shared memory, attribution, and adaptive governance.

In this sense, the Meta-Layer is more than an enhancement to the Web. It is an attempt to change the equilibrium conditions under which digital society operates, replacing coordination failure with an expanded possibility space in which collective intelligence, grounded in accountable participation and shared context, can emerge at scale.

7. From Individual Intelligence to Collective Intelligence

7.1 Why Collective Intelligence Is Systematically Undervalued

Collective intelligence compounds slowly but decisively through improved decision quality over time. Unlike individual productivity gains, which are immediately visible and easily measured, collective intelligence expresses itself as an expansion of **degrees of freedom**: more options available, more perspectives integrated, and more resilient paths forward.

Because these gains are diffuse, delayed, and often invisible at first, systems tend to underinvest in the conditions that enable them. Yet it is precisely this expanded possibility space – the ability to respond creatively rather than reactively – that determines long-term success in complex environments.

7.2 The Meta-Layer as an Environment for Collective Intelligence

Collective intelligence does not emerge automatically from connection or scale. It requires an environment in which knowledge can accumulate, disagreement can be processed productively, and decisions can be made with legitimacy over time.

The Meta-Layer provides such an environment by establishing persistent context, shared memory, and legible governance as baseline conditions. Rather than the normative model of ideal participation proposed by philosopher Jürgen Habermas, the Meta-Layer offers user-curated onramps and offramps for debate and deliberation. Persistent context allows participants to understand not just isolated statements, but patterns of behavior, contribution, and intent. Shared memory ensures that learning is not lost at every boundary, enabling communities to adapt rather than repeat failures. Legitimate governance provides a basis for resolving conflict and making binding decisions without resorting to force or fragmentation.

Together, these conditions transform intelligence from an individual attribute into a collective capability. Groups gain the ability to reason across time horizons, integrate diverse perspectives, and coordinate action in the face of uncertainty. The Meta-Layer does not guarantee wise outcomes, but it expands the space in which wiser outcomes become possible.

7.3 Human–AI Collective Intelligence

Artificial intelligence introduces a powerful new dimension to collective intelligence, but only if it is integrated as an augmenting force rather than a substitutive one. In the Meta-Layer, AI systems are treated as cognitive amplifiers: tools that extend perception, pattern recognition, and simulation capacity without displacing human judgment or legitimacy.

When AI is positioned as a decision authority, it collapses degrees of freedom by obscuring reasoning, concentrating power, and weakening accountability. By contrast, when AI operates within a context-rich, governable environment, it can support collective intelligence by surfacing insights, exploring alternatives, and revealing tradeoffs that humans alone might miss.

The Meta-Layer therefore emphasizes clear binding between AI agents, their operators, and the contexts in which they act by forcing explicit understanding of how agents “bridge” different parts of the knowledge and actor base. AI participation is differentiated from human participation, subject to asymmetric constraints, and embedded within interface-level governance. This ensures that human values, responsibility, and contestability remain central, even as cognitive capacity is amplified.

In this configuration, human–AI collective intelligence becomes a force multiplier for coordination through complexity rather than a source of new systemic risk.

8. New Forms of Value Creation

By expanding the conditions for trust, coordination, and collective intelligence, the Meta-Layer enables new forms of value creation that are poorly supported by today's Web. These forms of value are real and often challenging to measure. They already exist in fragmented, informal, or fragile ways. The Meta-Layer can make them legible, durable, and rewardable.

8.1 From Attention Extraction to Contextual Value

Today's Web primarily rewards attention, immediacy, and amplification. This creates incentives for sensationalism, polarization, and short-term engagement at the expense of accuracy, care, coordination, and long-term usefulness.

The Meta-Layer shifts value creation toward context. Contributions gain value not only from how widely they spread, but from how they are situated: who made them, how they relate to prior work, how they are interpreted over time, and how they support collective understanding. Context becomes a value multiplier.

8.2 Stewardship, Sensemaking, and Care as First-Class Value

Much of the most important work on the internet today is invisible or unrewarded. This includes moderation, curation, synthesis, translation, conflict resolution, and long-term stewardship of knowledge and communities.

By making relationships, provenance, and contribution histories explicit, the Meta-Layer enables these forms of labor to be recognized and valued. Communities can choose to reward not just original creation, but the ongoing work that keeps shared environments intelligible, trustworthy, and resilient.

8.3 Living Knowledge and Archival Revaluation

In a context-rich environment, archival content does not decay into irrelevance. Instead, it can be reactivated as part of a living knowledge system.

Past contributions gain new meaning when they are linked, contextualized, and reinterpreted in light of current questions.

This revaluation of archival material supports entirely new knowledge economies. Research, documentation, and historical record-keeping can become durable assets rather than sunk costs. Value can compound over time through reuse, reinterpretation, and integration.

8.4 Regenerative and Non-Extractive Value Models

Because the Meta-Layer supports long-horizon coordination, it enables value models that align individual incentives with collective wellbeing. Extractive dynamics that degrade trust and shared resources become less attractive when communities can coordinate restraint, reward care, and enforce norms transparently.

These dynamics open pathways for regenerative economies in which value creation strengthens the underlying social and informational commons rather than depleting it. In this sense, the Meta-Layer does not merely enable new markets. It enables healthier ones, grounded in the same expansion of collective capability that Engelbart envisioned decades earlier.

9. What the Desirable Properties Reveal

Taken together, the Desirable Properties reveal a structural theory of how value, legitimacy, and intelligence are created in a layered digital environment.

Each Desirable Property identifies a condition that must hold for participation, coordination, and intelligence to scale responsibly. These conditions are not independent. They form an interlocking system in which higher-order capabilities depend on the integrity of more fundamental ones. Trust, identity, accountability, context, and governance are not optional enhancements. They are enabling constraints that determine what kinds of coordination are possible at all.

This framing also clarifies why partial solutions fail. Weakness in foundational properties propagates upward, undermining higher-order capabilities regardless of how sophisticated later mechanisms appear. Conversely, strong foundations dramatically expand the space of viable outcomes, all-

owing advanced coordination to emerge without centralization or coercion.

In this sense, the Desirable Properties reflect a way of reasoning about system design. They help communities, builders, and policymakers explore where failures originate, which constraints are enabling rather than limiting, and how intentional structure can unlock greater degrees of freedom over time.

10. How to Read the DP Chapters

Each Desirable Property focuses on a single condition required for the Meta-Layer to function as intended. Together, they form a layered set of lenses for reasoning about system design, governance, and participation.

10.1 Properties as Conditions, Not Requirements

The DP chapters are not specifications, standards, or compliance checklists. They describe conditions that must plausibly hold for higher-order coordination and collective intelligence to emerge. Different communities may realize these conditions in different ways, and partial or experimental implementations are expected. Each chapter includes a Minimum Alignment section; it is binding only in the chapters that state it in capitalized requirement keywords, and elsewhere it is an assessment aid.

10.2 Dependencies and Ordering

The sequence of the Desirable Properties is currently provisional. DP1 (Federated Authentication & Accountability) grew from Vint Cerf's suggestion of federated strong authentication as an intentionally foundational starting point, while subsequent properties reflect an exploratory, community-informed working set that may be re-ordered or re-categorized as understanding improves.

Even so, dependencies still matter. Some properties function as preconditions for safe participation. Weaknesses at the foundation can propagate upward. Readers are encouraged to pay attention not only to each property in isolation, but to the emerging dependency structure across properties as the set evolves.

Viewed as a whole, the Desirable Properties function as a map of dependency and emergence even as the working set and ordering remain provisional.

10.3 Design Lenses, Not Prescriptions

Each DP chapter can be read as a design lens. It offers a way to interrogate systems, proposals, or communities by asking: does this environment actually support this condition? Where does it fall short, and what tradeoffs are being made?

This framing is intentionally flexible. It supports critique, comparison, and adaptation rather than enforcing uniform solutions.

10.4 ML-Drafts as a Living Process

The DP chapters are published as ML-Drafts to signal that they are works in progress. They are meant to evolve through critique, experimentation, and lived experience. Promotion toward more formal consensus documents, such as ML-RFCs, is expected to occur only where rough consensus emerges over time.

Different audiences may engage with these drafts in different ways. Policymakers may use them to reason about institutional safeguards and public interest. Builders may use them as architectural and governance lenses. Community stewards may use them to articulate norms and expectations. This diversity of use is intentional and reflects the Meta-Layer's commitment to pluralism and federation.

11. Why Begin with Trust

The decision to begin the Desirable Properties with trust – in the form of DP1: Federated Authentication & Accountability – is not arbitrary. Trust is the condition upon which all other forms of coordination depend. Without it, higher-order properties cannot function, no matter how sophisticated their design.

11.1 Trust as a Prerequisite for Coordination

Coordination requires participants to act with some confidence that others will behave in intelligible and bounded ways. This confidence does not

ly agreement or harmony. It implies legibility, accountability, and the expectation that actions are meaningfully attributable to agents.

In the absence of trust, coordination collapses into defensive behavior. Participants withhold information, avoid long-horizon commitments, and default to adversarial strategies. Even well-designed systems degrade under these conditions, because actors are forced to optimize for short-term self-protection rather than shared outcomes.

Trust, in this sense, is essential to coordination. It echoes early Internet assumptions about layered trust, legibility of endpoints, and the expectation of accountable participation that made open networks viable in the first place. It is the structural precondition that makes coordination feasible at all.

11.2 Why Other Properties Depend on DP1

Every subsequent Desirable Property assumes some baseline level of trust. Context cannot accumulate if identities are unstable. Governance cannot function if actions are not attributable. Collective intelligence cannot emerge if participants cannot rely on shared norms, memory, and accountability.

DP1 addresses these foundational requirements directly. By establishing conditions for identity, accountability, and legitimacy, it creates the substrate upon which other properties can operate. Weakness at this layer propagates upward, undermining more advanced capabilities regardless of their technical sophistication.

This dependency structure explains why trust must be addressed first. It is not because trust is sufficient on its own, but because it is necessary for everything that follows.

11.3 Trust as an Enabling Condition, Not an Outcome

Trust is often framed as something to be achieved after coordination succeeds, or as a reward for good behavior. In the Meta-Layer, trust is understood differently. It is a freedom-enabling condition that expands the space of possible actions.

When trust is present, actors gain additional degrees of freedom. They can take risks, delegate responsibility, share partial information, and coordinate over longer time horizons. These behaviors are prerequisites for learning, adaptation, and collective intelligence.

When trust is absent, systems collapse into defensive simplicity. The range of viable actions narrows, coordination becomes brittle, and participants are pushed toward short-term, zero-sum strategies. Complexity becomes unmanageable when the conditions required to navigate it are missing.

Beginning with trust reflects this reality. Trust is the condition that enables coordination through complexity to occur at all. Without it, higher-order collective intelligence remains structurally unreachable, regardless of intent or effort.

12. Looking Ahead

The Meta-Layer is not a destination, but an evolving civic substrate. The Desirable Properties provide a shared vocabulary for navigating this evolution without collapsing into premature standardization or fragmentation.

The work that follows should be understood as an invitation rather than a conclusion. Each Desirable Property represents a condition that must be stewarded over time, not merely implemented once. As technologies, actors, and social norms evolve, these properties will need to be interpreted, tested, and refined in practice.

What follows begins with trust, not because trust is sufficient on its own, but because without it, higher-order coordination is not possible. The chapters that follow explore how additional properties build upon that foundation to support a trustworthy, adaptive, and collectively intelligent digital future. In this sense, the Desirable Properties are less a blueprint than a shared responsibility for shaping the conditions under which the Meta-Layer can evolve with purpose and integrity.

Read the Introduction online →

DESIRABLE PROPERTY 1

DP1

Federated Authentication & Accountability

The Conditions for Trust – strong, federated authentication and decentralized accountability are the foundations for real trust, safe interaction, and meaningful community.



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp01

1. Purpose of This Draft

This draft articulates **Desirable Property 1 (DP1)** as a foundational condition for trust in the Meta-Layer. It expands DP1 beyond federated authentication to encompass accountability, adaptive intelligence integration, and foresight-driven governance.

DP1 responds to multiple, overlapping needs:

- The need for decentralized, federated identity without single points of control
- The need for durable accountability without mandatory real-world identity
- The need to govern both human and AI agents coherently
- The need to anticipate predictable abuse and governance failure modes

This draft is intended to guide implementation, governance design, and future ML-RFC development.

2. Problem Statement

Why identity alone is not enough.

For most of the web's history, trust has been treated as a byproduct of identity. If a participant could be authenticated, logged in, or verified, trust was assumed to follow. This assumption no longer holds.

At contemporary scale, identity has become cheap to generate, easy to discard, and increasingly decoupled from responsibility. As a result, systems optimized around login and verification routinely fail to protect participants, communities, and institutions from predictable harm.

DP1 begins from a different premise: **trust is not something identity produces on its own**. Trust emerges only when identity is paired with accountability, memory, and governance that operate coherently at the point of interaction.

2.1 The Limits of Login-Centric Trust

Login-centric trust models focus on answering a narrow question: who is allowed to enter a system. They do not meaningfully address what happens

after entry.

Across platforms and applications, this has produced a recurring pattern:

- Verification is treated as a proxy for good faith
- Identity checks are decoupled from behavior over time
- Enforcement resets when identities are abandoned or recreated
- Predators, scammers, fake accounts, and autonomous agents are able to exploit accumulated trust within an ecosystem before detection or response

Even strong authentication does not prevent abuse when actions are not durably bound to accountable actors. A verified account can still mislead, manipulate, impersonate, or cause harm if there is no persistent relationship between identity and responsibility.

2.2 Structural Failure Modes in Today's Web

Several structural conditions compound these weaknesses:

- **Context collapse**: Behavior in one space rarely follows a participant into another
- **No shared memory**: Harm accumulates, but accountability does not
- **Reactive moderation**: Intervention occurs after damage is done
- **Synthetic scale:** Bots, sockpuppets, and automated agents operate faster than human oversight

These are not edge cases. They are systemic properties of the current web. As documented in Meta-Layer research, even the largest platforms remove billions of fake or abusive accounts annually, without meaningfully reducing the underlying incentives or recurrence of abuse.

2.3 DP1 as a Shift in Framing

DP1 reframes the problem of trust along three axes:

- From **identity** to **accountability**
- From **platforms** to **zones**
- From **enforcement** to **conditions**

Rather than asking only who someone is, DP1 asks under what conditions participation is allowed, how actions are attributed, and how trust evolves over time.

3. Threats and Failure Modes

DP1 is not defined in opposition to any single class of actor. Instead, it responds to recurring failure modes that reliably emerge in large-scale, low-friction digital systems.

Two surfaces are in scope. The first is **adversarial**: classes of actor whose behavior reliably degrades trust wherever identity is cheap and accountability is discontinuous. The second is **structural**: the pressures catalogued under Section 3.4 below, which erode trust even where no actor intends harm. DP1 treats both as design inputs, because a trust system that defends only against deliberate attackers will still fail under ordinary growth logic.

3.1 Scammers

Scammers exploit environments where identity is inexpensive and disposable. Common characteristics include:

- Rapid account creation and abandonment
- Cross-context exploitation of trust signals
- Asymmetric incentives favoring deception

DP1 does not attempt to eliminate scams entirely. Instead, it raises their cost by binding actions to accountable agents, preserving memory across contexts, and enabling communities to escalate trust requirements where appropriate.

3.2 Serial Predators and Repeat Abusers

Serial abuse often persists not because it is invisible, but because it is fragmented. When identities reset after enforcement, harm becomes distributed across communities without a durable record.

DP1 addresses this pattern by supporting persistent pseudonymous identity, zone-scoped accountability, and governance mechanisms that allow communities to respond to patterns of harm without resorting to exposure, vigilantism, or centralized surveillance.

3.3 Impersonators (Human and AI)

Advances in generative systems have dramatically lowered the cost of impersonation. Voice, image, and text synthesis now allow both humans and AI systems to convincingly misrepresent identity and intent.

DP1 counters impersonation by binding content and actions to verifiable agents, clearly differentiating between human and AI actors, and surfacing provenance signals directly at the interface layer.

3.4 Other Drivers (Equally Important)

In addition to explicit abuse, DP1 responds to broader systemic pressures that erode trust even in the absence of malicious intent:

- **Scalable agents operating without visible constraints.** As automation and AI agents scale, they can overwhelm human participation, distort consensus, and accumulate influence faster than human governance processes can respond. Without clear accountability, visibility, and rate limits, both human-operated and autonomous agents can unintentionally or deliberately reshape an ecosystem's trust dynamics.
- **Governance capture and conflicts of interest.** Trust systems are vulnerable when those who set or enforce rules have misaligned incentives. Concentrated power, opaque decision-making, or financial dependencies can lead to selective enforcement, uneven accountability, or loss of legitimacy. Over time, this erodes community confidence even if formal rules remain unchanged.
- **Economic models that reward engagement regardless of harm.** Many digital systems optimize for growth, virality, or attention without regard for downstream effects. When visibility, rewards, or influence are tied solely to engagement metrics, deceptive, polarizing, or manipulative behavior is systematically advantaged over constructive participation.

These pressures are structural rather than incidental. They interact and compound one another, producing environments where abuse, manipulation, and trust erosion become predictable outcomes. DP1 is designed to address their combined effects by reshaping incentives, accountability, and governance conditions, rather than treating each pressure in isolation. Taken together, these pressures make clear that trust cannot be repaired

solely through backend policy or platform moderation, but must be enacted visibly and continuously at the interface level, where participation, amplification, and accountability actually occur.

3.5 Identity-Layer Failure Modes

The mechanisms defined later in this draft each guard against a named failure mode. Collected here, they form the adversarial checklist against which a DP1 implementation should be evaluated:

- **Identity fragmentation**: the same participant appears as unrelated actors across systems, breaking incentives, governance, and trust
- **Identity replay**: actions or credentials are reused across systems to gain unearned trust, rewards, or access
- **Identity laundering**: responsibility is shifted across actors, tools, or delegates to evade accountability
- **Sybil saturation**: large numbers of coordinated identities overwhelm governance, incentives, or visibility systems
- **Semantic drift**: identity signals are assumed to carry the same meaning across contexts where they do not
- **Lineage loss**: identity history cannot be reconstructed, enabling impersonation or erasure of harmful behavior
- **Responsibility laundering**: automated outputs circulate without a traceable, contestable responsible entity
- **Governance drift**: adaptive or automated systems reshape trust conditions without visible human ratification

Each of these is addressed in the sections that follow, and each reappears as a testable condition under Section 19.

4. Core Principle

Identity is not merely descriptive at the interface layer; it is the basis for enforceability, continuity, and accountable participation across all Meta-Layer interactions.

Identity is the enforcement boundary of the Meta-Layer. If identity cannot persist across context, delegation, and scale, trust collapses into

simulation.

DP1 establishes identity as an enforceable, continuous, and context-bound substrate for all higher-order trust, governance, and interaction within the Meta-Layer.

Trust in the Meta-Layer emerges when identity, accountability, learning, and foresight are bound together at the interface level.

This principle has several direct implications, each of which is essential to sustaining trust at scale:

- **Identity is plural and contextual, not singular or global.** Participants may operate under different identities in different zones, allowing communities to set appropriate norms without forcing a single global identity model. This preserves inclusion, safety, and local governance autonomy.
- **Accountability attaches to actions, not just names.** Trust depends on the ability to evaluate behavior over time. Binding actions to accountable agents ensures that responsibility persists even when identities are pseudonymous or federated.
- **Memory is preserved without requiring mass surveillance.** Durable, attributable records allow communities to learn from past behavior and prevent repeat abuse, while avoiding continuous monitoring or centralized data collection.
- **Governance adapts over time, but remains human-ratified.** Trust systems must evolve in response to new threats and conditions, yet retain human oversight so that changes remain legitimate, explainable, and aligned with community values.

DP1 does not promise perfect safety or universal trust. Instead, it defines the minimum conditions under which trust can form, persist, and be repaired in complex, multi-actor environments.

5. Primary Mechanisms and Structural Conditions

DP1 is enacted through a small number of composable mechanism families. None of them is sufficient alone; trust emerges from their combination, and

a system that implements some while omitting others will exhibit the failure modes named above.

The mechanism families are:

- **Federated strong authentication** establishes the entry condition. It ensures that access to shared spaces is neither trivially cheap nor monopolized by a single identity authority, while leaving trust decisions to the zone.
- **Sociotechnical zones** translate governance principles into concrete, composable participation rules that operate at the interface, where interaction and amplification actually occur.
- **The identity system layer** carries continuity, integrity, non-transferability, sybil resistance, cross-context semantics, and lineage across environments and time, so that identity remains interpretable as it moves.
- **Action-bound accountability** attaches responsibility to actions rather than names, supported by pseudonymity with responsibility, sealed memory with editability windows, and an explicit trust lifecycle covering escalation, revocation, and recovery.
- **Contestability and due process** make the accountability system itself accountable, through explainability, appeals, and human review at consequential thresholds.
- **Agent classification and asymmetric constraint** keep human, AI, and hybrid participation legible and bounded, and bind automated outputs to responsible entities.
- **Adaptive intelligence under human ratification** allows trust conditions to learn from changing behavior without transferring authority to opaque systems.

These families operate as a stack of conditions rather than a sequence of checks. Authentication without accountability produces verified abuse. Accountability without contestability produces captured enforcement. Zones without continuity produce fragmentation. Adaptation without ratification produces governance drift. The structural condition DP1 imposes is therefore *coherence*: each mechanism must remain legible and enforceable

at the point of interaction, and must degrade visibly rather than silently when a boundary is crossed.

The sections that follow define each family in depth.

6. Federated Strong Authentication (Entry Condition)

Federated strong authentication establishes the baseline condition for participation in the Meta-Layer. Its purpose is not to define trust, but to ensure that entry into shared spaces is not trivially exploitable or monopolized by a single identity authority.

DP1 treats authentication as an **entry condition**, not as a guarantee of trustworthiness or good behavior. Strong authentication reduces frictionless abuse, but only when paired with downstream accountability, memory, and governance does it meaningfully contribute to trust.

6.1 Federation as a Baseline Requirement

The Meta-Layer supports federation across multiple identity and authentication systems, including traditional SSO providers, wallets, and emerging credential frameworks. This plural approach ensures:

- No single provider controls access to the Meta-Layer
- Participants can authenticate using systems appropriate to their context
- Communities can adopt stronger or lighter requirements without fragmenting the ecosystem

Federation is essential to resilience. Centralized identity systems concentrate power and risk, while federated systems distribute trust and reduce systemic failure modes.

6.2 User-Held Keys and Credentials

Where possible, participants hold their own keys or retain meaningful control over credentials. In cases where custodial systems are used, consent and revocability remain core requirements.

User-held credentials support:

- Participant agency and exit
- Reduced platform lock-in

- Durable accountability across contexts

6.3 Authentication Is Not Authorization

Authentication answers the question of *who may enter*. It does not determine *what that participant may do*, *what they may access*, or *how much trust they are afforded*.

All authorization, trust thresholds, and participation rules are defined at the zone level. This separation prevents overloading identity systems with governance logic and keeps trust decisions contextual, transparent, and adaptable.

Authentication and public identification are different acts. A person or device should be able to prove what an interaction requires without making unrelated identity attributes public or linking other personas. A SIM authenticating to a network illustrates the distinction, not anonymity from the network operator. Anonymous participation remains subject to the zone-specific boundaries in Sections 7 and 18.

7. Sociotechnical Zones as Trust Contexts

Sociotechnical zones are the primary mechanism by which trust conditions are enacted at the interface level. Zones translate abstract governance principles into concrete participation rules that operate where interaction, amplification, and accountability actually occur.

Rather than relying on backend policy enforcement or platform-level moderation alone, zones make trust visible, enforceable, and configurable within the lived experience of participants.

7.1 Definition of Sociotechnical Zones

Zones combine technical requirements and social norms to define the conditions under which participation is permitted.

Each zone specifies:

- Acceptable risk levels
- Required proofs or credentials
- Accountability expectations

- Governance and escalation pathways

Communities and applications choose which zones they operate within, allowing trust conditions to vary without fragmenting the underlying Meta-Layer.

7.2 Zone-Specific Access Paradigms

Zones represent **orthogonal and composable trust constraints**. Real-world environments typically operate under multiple zones simultaneously, reflecting layered social, legal, and safety requirements.

Open and Identity-Light Zones

- Tokenless open zones
- Pseudonymous zones

Credential and Federation-Based Zones

- Credential-gated zones
- Federated authentication zones

Safety and Constraint-Oriented Zones

- Age-gated zones
- Human-only zones
- High-trust or safety-critical zones

By composing zones, communities can precisely calibrate participation conditions without defaulting to global restrictions.

7.3 Compatibility and Exclusion by Design

Zones enforce explicit compatibility requirements. Participation is limited to actors who can meet the defined conditions, making boundaries legible rather than implicit.

This design:

- Prevents silent exclusion or shadow banning
- Makes trust thresholds understandable and contestable
- Allows communities to defend against abuse without universal surveillance

7.4 Proof of Humanity as a Zone-Scoped Basis for Participation

Proof of humanity refers to mechanisms that allow a participant to demonstrate that they are a **unique human actor**, without necessarily revealing their real-world identity.

Within DP1, proof of humanity is treated as a **foundational system capability** that must be available across the Meta-Layer, even though its enforcement is zone-scoped and community-defined. This capability is critical at the interface level, where rewards, visibility, and reputation are allocated and where synthetic scale can otherwise distort outcomes.

Some communities may choose to make proof of humanity the basis for participation itself. Others apply it selectively to specific functions such as rewards, governance, rate-limited actions, reputation amplification, or access to safety-critical spaces.

Key principles include:

- Proof of humanity is available system-wide, but enforced at the zone level
- Communities decide when and how proof of humanity is required, including making it a prerequisite for participation, rewards, governance, or amplification
- Proof of unique humanity is commonly required to unlock rewards, boost virality, or affect transferable reputation, where synthetic scale would otherwise overwhelm human participation
- Multiple proof mechanisms may coexist, be combined, or be phased out over time
- Proof of humanity does not imply real-name identity or permanent disclosure
- Requirements may be stricter in high-trust, safety-critical, or resource-allocation contexts

By treating proof of humanity as an enduring and adaptable capability rather than a fixed mechanism, DP1 enables long-term defense against synthetic scale and impersonation while preserving pluralism, pseudonymity, and local governance autonomy.

8. Identity System Layer: Continuity, Integrity, and Adversarial Resilience

Beyond authentication and zone-scoped participation, DP1 requires a coherent identity system layer that persists across environments, interactions, and time. This layer is the enforcement boundary of the Meta-Layer. If identity cannot maintain continuity under scale, delegation, and interoperability, trust collapses into simulation.

The identity system layer ensures that actions, reputation, and accountability remain meaningfully bound to agents even as they move across zones, tools, and contexts.

8.1 Identity Continuity Across Systems

Identity must persist across platforms, zones, and applications without fragmenting into unrelated entities.

Continuity requires:

- Stable identifiers or linkable identity references
- Preservation of accountability history across environments
- Explicit signaling when identity properties degrade or reset

A failure mode is **identity fragmentation**, where the same participant appears as unrelated actors across systems, breaking incentives, governance, and trust.

8.2 Identity Integrity and Anti-Replay Guarantees

Identity-bound actions must not be duplicable across systems without attribution.

This requires:

- Binding actions to identity with verifiable provenance
- Preventing replay of actions or credentials across contexts
- Ensuring that contributions cannot be re-used to extract value multiple times

A failure mode is **identity replay**, where actions or credentials are reused across systems to gain unearned trust, rewards, or access.

8.3 Identity Non-Transferability and Delegation Boundaries

Identity must not be freely transferable in ways that detach responsibility from the original actor.

Delegation is permitted, but must be:

- Explicit
- Scoped
- Attributable

This ensures that actions taken by agents, tools, or collaborators remain traceable to accountable principals.

Accountability can be transitive rather than global. When Alice works with Bob and Bob brings in Carol to help with Alice's task, Alice need not authenticate Carol: she authenticates Bob in whatever way she chooses and holds Bob responsible for Carol's use of her resources. Trust, identity, and accountability then flow along chains of delegated authorization, which scales to remote and open collaboration where requiring every party to authenticate every other does not.

A failure mode is **identity laundering**, where responsibility is shifted across actors to evade accountability.

8.4 Resistance to Sybil and Coordinated Identity Attacks

Identity systems must make large-scale duplication, coordination, or synthetic amplification detectable, constrained, or economically costly.

Mechanisms may include:

- Proof of humanity where appropriate
- Behavioral analysis and rate limits
- Reputation weighting and trust decay

A failure mode is **sybil saturation**, where large numbers of coordinated identities overwhelm governance, incentives, or visibility systems.

8.5 Cross-Zone Identity Semantics and Degradation

Identity does not carry identical meaning across all zones.

Systems must:

- Signal when identity guarantees change across contexts
- Prevent misinterpretation of reputation or credentials
- Allow communities to define how identity signals are interpreted locally

A failure mode is **semantic drift**, where identity signals are incorrectly assumed to carry the same meaning across different contexts.

8.6 Identity Memory and Lineage

Identity must retain a reconstructable history of actions, credentials, and governance interactions over time.

Lineage enables:

- Attribution of contributions
- Detection of behavioral patterns
- Accountability across time and context

Breaks in lineage must be treated as risk signals rather than neutral events.

A failure mode is **lineage loss**, where identity history cannot be reconstructed, enabling impersonation or evasion.

This identity system layer does not require centralization or global identity unification. It requires coherence. Identity must remain usable, accountable, and interpretable across the Meta-Layer without collapsing into surveillance or fragmentation.

9. Accountability as a First-Class Property

Accountability is the core mechanism through which trust becomes durable in the Meta-Layer. While authentication governs entry, accountability governs behavior over time. Without it, trust signals decay, abuse repeats, and governance loses legitimacy.

DP1 treats accountability as a first-class property that operates continuously at the interface level, binding actors to their actions in ways that are visible, attributable, and contestable, without requiring real-world identity disclosure.

9.1 Action-Bound Accountability

In the Meta-Layer, accountability attaches to actions, not merely to identities. Every meaningful action, such as posting content, issuing judgments, triggering automation, or influencing visibility, is bound to an accountable agent identifier.

This ensures that:

- Actions have clear provenance
- Responsibility persists across time and context
- Trust assessments can be based on behavior, not credentials alone

Action-bound accountability allows communities to reason about patterns of conduct without collapsing participation into real-name systems or centralized surveillance.

9.2 Pseudonymity with Responsibility

DP1 explicitly supports pseudonymous participation, recognizing its importance for safety, expression, and inclusion. Pseudonymity, however, does not imply anonymity from accountability.

Persistent pseudonymous identities allow participants to:

- Build reputation over time
- Be held responsible for repeated behavior
- Participate across zones without exposing real-world identity

Communities may permit multiple personas per participant, subject to local rules, provided that accountability requirements are met. This balances flexibility with responsibility, enabling participation without enabling evasion.

Anonymity from other participants can coexist with extensive visibility to an operator or institution. Identity-system review should distinguish public disclosure, operator observation, and inferred linkage rather than treating a pseudonym as evidence that all three are controlled.

9.3 Sealed Memory and Editability Windows

To balance forgiveness, accuracy, and integrity, DP1 supports time-bound editability followed by sealing.

Participants may edit or retract contributions within community-defined windows. After this period, contributions become sealed: immutable, attributable, and part of the shared civic memory.

Sealed memory:

- Prevents retroactive manipulation
- Preserves historical context
- Enables learning from past behavior

Communities may determine whether edit histories are retained, visible, or restricted, but the existence of durable memory is essential for trust to accumulate.

9.4 Trust Lifecycle, Revocation, and Recovery

Trust in the Meta-Layer is not binary. It evolves.

Zones define explicit conditions for:

- **Escalation and intervention**: graduated responses to harmful or destabilizing behavior
- **Temporary suspension or restriction**
- **Revocation of access or privileges**
- **Reinstatement or recovery**

Revocation is zone-scoped by default, avoiding unnecessary global punishment. Memory persists across decisions.

10. Contestability, Appeals, and Due Process

For accountability systems to be trusted, they must themselves be accountable. DP1 therefore treats contestability and due process as essential trust infrastructure, not optional governance overhead.

Participants must be able to understand, challenge, and appeal decisions that materially affect their participation, visibility, reputation, or access.

Key principles include:

- **Transparency**: Decisions affecting trust or access must be explainable and grounded in visible rules or signals.
- **Contestability**: Participants must have mechanisms to dispute actions taken against them.
- **Human Oversight**: Escalation thresholds require human review, particularly where consequences are significant.
- **Explainable Automation**: AI-assisted flagging or enforcement must be intelligible to affected parties.

Appeals processes reinforce legitimacy. They help communities detect governance failure, correct errors, and adapt rules over time.

By embedding contestability directly into trust systems, DP1 ensures that accountability strengthens trust rather than undermining it.

In summary:

- Accountability systems must be challengeable and auditable
- Participants must be able to contest decisions affecting access, reputation, or visibility
- Escalation thresholds require human review
- AI-assisted flagging and enforcement must be explainable
- Appeals are treated as trust infrastructure, not optional overhead

11. Human and AI Agents Under DP1

DP1 treats both human and artificial agents as first-class participants in the Meta-Layer, while recognizing that they differ fundamentally in capacity, scale, intent, and risk profile. Trust cannot be sustained if these differences are ignored, obscured, or flattened.

The goal of DP1 is not to exclude AI agents categorically, but to ensure that their participation is **legible, bounded, and accountable** in ways that preserve human agency and community governance.

11.1 Agent Classification and Visibility

An *agent* refers to any actor capable of taking actions that affect shared environments, visibility, reputation, or outcomes within the Meta-Layer.

DP1 requires clear classification between:

- Human agents
- AI or automated agents
- Hybrid or assisted agents, where human intent is mediated by automation

This classification must be **visible at the interface level**, allowing participants to understand whether they are interacting with a human, an AI system, or a combination of both. Hidden or ambiguous agent identity erodes trust and enables manipulation.

11.2 Symmetric Accountability, Asymmetric Constraints

DP1 applies accountability symmetrically: all agents are accountable for their actions. However, constraints are applied asymmetrically, reflecting differences in scale, speed, and potential impact.

For example:

- AI agents may be subject to stricter rate limits, scope restrictions, or amplification caps
- Certain zones may restrict participation to human agents only
- Higher proof thresholds may apply where AI activity could distort consensus, rewards, or governance

This approach avoids both extremes: granting AI agents unchecked parity with humans, or exempting them from accountability altogether.

11.3 Binding AI Outputs to Responsible Entities

AI agents do not operate independently of human or institutional responsibility. DP1 therefore requires that AI outputs be bound to a responsible entity, such as:

- The operator deploying the agent
- The organization maintaining it

- A community governance structure authorizing its use

In high-trust or safety-critical zones, anonymous autonomous agents are not permitted. Responsibility must be traceable, contestable, and enforceable.

These bindings also apply to advanced AI and complex systems managing critical infrastructure, including physical or hybrid systems. Greater capability does not remove the need for an identifiable responsible entity.

By binding AI behavior to accountable entities, DP1 prevents responsibility laundering while enabling beneficial automation under governed conditions.

11.4 Contextual Agent Registration

Agents acting on behalf of people, services, or other agents should be identifiable in the context where they act. Registration can use mechanisms analogous to software identification to connect the acting agent, its mandate, and its responsible entity. It should not require a single central registry or indiscriminate public disclosure of the represented participant.

12. Adaptive Intelligence Integration (RLADP)

Static trust and governance systems degrade over time. Incentives shift, adversaries adapt, and behaviors drift. DP1 therefore anticipates the need for adaptive intelligence to support, but not replace, human and community governance.

12.1 Why Static Governance Fails

At scale, purely static rules and manual moderation encounter predictable limits:

- Human moderators cannot match the speed or volume of adversarial behavior
- Rules ossify and become misaligned with lived practice
- Bad actors learn to game fixed thresholds and heuristics

Without adaptation, governance systems either become overly permissive or increasingly brittle.

12.2 RLADP as Advisory Infrastructure

DP1 envisions adaptive intelligence, including reinforcement learning and approximate dynamic programming (RLADP), as **advisory infrastructure**.

Adaptive systems may:

- Detect emerging patterns of abuse or manipulation
- Surface signals about shifting norms or risk profiles
- Propose adjustments to thresholds, friction, or zone parameters

They may not:

- Unilaterally change rules
- Impose sanctions without human ratification
- Operate as opaque or unchallengeable authorities

12.3 Transparency and Auditability

All adaptive processes must be observable and auditable. Communities must be able to understand:

- What signals are being used
- How recommendations are generated
- What effects adaptations have produced

This visibility is essential to preventing hidden governance drift and maintaining legitimacy.

12.4 Human and Community Ratification

Adaptive intelligence proposes; humans decide.

Material changes to trust conditions, enforcement thresholds, or governance rules require explicit human or community ratification, using processes appropriate to the zone.

By constraining adaptive intelligence within transparent, ratified loops, DP1 enables learning without surrendering agency or accountability.

13. Governance, Accountability, and Agency Surfaces

DP1 is not satisfied by identity architecture alone. The conditions for trust must be *experienced* as navigable surfaces, or they remain private properties of a backend that participants are asked to take on faith. A system can bind every action to an accountable agent and still fail DP1 if participants cannot see the binding, cannot tell which zone rules apply to them, and cannot challenge a decision that changes their standing.

Participants must be able to:

- see which zone they are currently participating in, what proofs it requires, and which of those proofs they currently satisfy
- see whether the actor they are interacting with is classified as human, AI, or hybrid, and which entity is responsible for that actor's behavior
- inspect what is attributed to their own agent identifier, and within what editability window a contribution remains revisable before it is sealed
- understand their current trust state in a zone, including any active restriction, its stated basis, its duration, and the recovery pathway
- initiate an appeal against a decision affecting access, visibility, or reputation, and observe its status rather than waiting on an invisible queue
- understand when identity continuity is about to degrade – on export, migration, fork, or persona separation – before the degradation occurs

Communities must be able to:

- define, publish, and revise the zone conditions under which they operate, including proof requirements and escalation thresholds
- see who currently holds enforcement authority in the zone, under what mandate, and for how long
- audit whether enforcement actions were executed as ratified, and whether adaptive recommendations were adopted, modified, or rejected
- review aggregate accountability outcomes – appeal volumes, reversal rates, escalation frequency – without exposing individual participants
- preserve governance memory of why a trust rule exists, what incident prompted it, and what effect it produced

- fork zone rules, or exit, when legitimacy breaks down, without stripping members of identity continuity

Example: A participant restricted from amplifying content in a high-trust zone sees the specific rule invoked, the signal that triggered it, whether a human ratified the action, the date the restriction lapses, and a single path to contest it. The community, separately, can see how often that rule fires and how often contests succeed.

Without these surfaces, accountability becomes indistinguishable from arbitrary enforcement. The failure mode is **opaque legitimacy**, where a system's trust decisions may in fact be sound but cannot be verified as such, and therefore accumulate distrust at the same rate as capture would.

14. Incentives and Power Analysis

Identity systems are never neutral infrastructure. Whoever controls issuance, verification, revocation, or resolution of identity holds structural power over participation itself, and that power is valuable enough that it will be sought whether or not the formal rules permit it.

DP1 therefore treats incentive structure as part of the trust surface, not as context outside it.

Several pressures recur:

- **Verification as a rent.** Where a single provider becomes the de facto path to credible participation, verification becomes a tollgate. Federation is not merely a resilience property; it is an anti-rent property, ensuring that no issuer can price access to the Meta-Layer.
- **Engagement-weighted visibility.** Where reach is allocated by engagement alone, synthetic scale is economically rational. Proof-of-humanity gating for rewards and amplification is therefore an incentive intervention, not only a security control.
- **Enforcement asymmetry.** Those who set rules, enforce them, and benefit from their outcomes are frequently the same actors. DP1 requires that these relationships be visible (see Section 16.2), because conflicts of interest degrade legitimacy even absent bad intent.

- **Reset economics.** When identity is cheap to abandon, sanctions become a cost of doing business rather than a deterrent. Persistent identity, lineage, and privilege loss on reset change the arithmetic of abuse.
- **Data leverage from accountability.** Accountability infrastructure generates exactly the records that surveillance economies value. DP1's insistence on pseudonymity, zone-scoping, and non-propagation by default is an incentive constraint on the operators of trust systems themselves.
- **Automation subsidy.** Automated agents can absorb friction that would deter humans. Asymmetric rate, scope, and amplification constraints prevent capital from purchasing disproportionate presence.

Example: A platform requires proof of humanity for rewards but exempts partner integrations to protect a revenue relationship. No rule changed, yet the incentive to route abuse through partner surfaces was created.

Why this matters: A trust system that does not act on incentives will be routed around by them. DP1 therefore expects that materially relevant incentives – who funds enforcement, who profits from visibility, who benefits from a given verification path – be legible to communities, and that zones retain the ability to constrain those incentives locally.

15. Community Signals Informing DP1

DP1 reflects recurring themes from community submissions, workshops, and discussions across the Meta-Layer initiative.

While individual inputs vary, several consistent signals emerge:

- **A strong preference for pseudonymity with accountability, rather than forced real-name identity.** Communities repeatedly emphasize the need to separate accountability from exposure. Participants want the ability to speak, contribute, and organize without tying activity to real-world identity, while still ensuring that actions carry responsibility over time. This reflects lived experience in environments where real-name policies create safety risks, suppress participation, or concentrate power, without reliably preventing abuse.

- **Frustration with repeat abusers enabled by identity resets and fragmented enforcement.** Many communities report that harmful behavior persists not because it goes unnoticed, but because enforcement lacks continuity. When identities can be cheaply abandoned and re-created, sanctions lose meaning and abuse becomes a cost of doing business. This signal directly informs DP1's emphasis on persistent, pseudonymous identity and durable memory.
- **Concern about synthetic scale, including bots and AI agents overwhelming human participation.** Participants consistently describe environments where automated or semi-automated agents dominate visibility, rewards, or discourse. Even benign automation can distort outcomes when scale is unchecked. This concern motivates proof-of-humanity capabilities, asymmetric constraints for AI agents, and explicit limits on amplification and rate.
- **Distrust of opaque or centralized moderation and fear of governance capture.** Communities express low confidence in trust systems where rules are enforced invisibly or controlled by unaccountable actors. Perceived capture, selective enforcement, or undisclosed incentives erode legitimacy even when formal policies appear sound. This drives DP1's requirements for transparency, contestability, and visible governance at the interface level.
- **Desire for preventative and restorative approaches rather than purely punitive systems.** Many submissions emphasize that punishment alone does not build trust. Communities want mechanisms that prevent harm upstream, allow for learning and repair, and support reintegration where appropriate. This signal underlies DP1's focus on graduated responses, recovery pathways, and governance that evolves through foresight rather than crisis.

These signals reinforce the core framing of DP1: trust must be designed as a set of conditions that balance agency, safety, and legitimacy, rather than imposed through static rules or centralized control. DP1 is therefore best understood not as a single solution, but as a shared response to patterns of failure repeatedly identified by communities operating at scale.

16. Foresight and Failure Design

DP1 treats foresight not as speculation, but as a core governance discipline. Large-scale sociotechnical systems fail in recognizable ways. When trust systems are designed only for normal operation, they become brittle under stress, capture, or adversarial pressure.

Minefield thinking refers to the practice of deliberately anticipating where incentives, power, and scale are likely to produce failure, and designing safeguards in advance rather than reacting after harm has occurred.

16.1 Governance as Anticipatory Design

Most trust failures are not surprises. They arise from known dynamics such as incentive misalignment, asymmetric power, scale effects, and adversarial learning.

DP1 therefore treats governance as an anticipatory design problem. Communities are encouraged to:

- **Identify foreseeable abuse and failure modes.** Rather than assuming good-faith participation as a default, communities are encouraged to explicitly map how their systems could be exploited, stressed, or captured at scale. This includes considering adversarial behavior, incentive misalignment, power concentration, and unintended consequences of well-meaning rules.
- **Encode preventative friction rather than relying solely on punishment.** Preventative friction includes rate limits, proof thresholds, graduated permissions, and contextual checks that slow or deter harmful behavior before it escalates. This reduces reliance on after-the-fact enforcement, which is often costly, contentious, and insufficient to prevent harm.
- **Periodically reassess assumptions as conditions change.** Trust systems operate in dynamic environments. Communities are encouraged to revisit governance assumptions as participation grows, technologies evolve, or incentives shift, ensuring that rules remain aligned with lived practice rather than ossifying over time.

This approach shifts governance from reactive moderation to continuous risk management.

16.2 Conflict of Interest (COI) Visibility

Trust erodes when participants cannot see whose interests shape rules and enforcement. DP1 requires that material conflicts of interest be surfaced structurally rather than assumed away.

This includes visibility into:

- **Funding sources and economic incentives.** Communities benefit from understanding who funds infrastructure, moderation, or tooling, and how revenue models or token incentives may shape decision-making. Visibility into economic incentives helps participants evaluate whether rules are aligned with collective goals or subtly optimized for growth, extraction, or control.
- **Governance authority and decision rights.** Trust depends on knowing who has the power to set rules, enforce them, and change them over time. Clear articulation of decision rights allows participants to assess legitimacy, understand escalation pathways, and distinguish community governance from operator discretion.
- **Relationships between operators, enforcers, and beneficiaries.** When the same actors design rules, enforce them, and benefit from their outcomes, conflicts of interest can arise even without malicious intent. Making these relationships explicit allows communities to surface bias, challenge capture, and adjust governance structures before trust erodes.

By making incentives legible, communities can better assess legitimacy, detect capture early, and sustain confidence in governance over time.

16.3 Governance Pre-Mortems

DP1 encourages communities to conduct periodic governance pre-mortems: structured exercises that ask how current rules or systems might fail under plausible future conditions.

Pre-mortems may examine:

- **How rules could be gamed at scale.** Communities are encouraged to consider how well-intentioned rules might be exploited when participation grows, automation increases, or incentives shift. This

includes identifying loopholes, edge cases, or feedback loops that could advantage bad-faith actors.

- **Where enforcement might become selective or biased.** Pre-mortems surface the risk that enforcement could drift toward favoritism, uneven application, or disproportionate impact on certain groups. Making these risks explicit allows communities to design checks, audits, or appeals in advance.
- **How new technologies or actors could distort participation.** Emerging tools, AI capabilities, or new classes of participants may change how power and influence are exercised. Pre-mortems help communities anticipate these shifts rather than reacting after harm occurs.

The goal is not prediction, but preparedness. Pre-mortems create shared awareness of fragility, normalize course correction, and reduce the social and political cost of adaptation.

16.4 Exit, Fork, and Kill Switches

No governance system should assume its own permanence. DP1 treats exit as a safety feature rather than a failure, recognizing that the ability to leave or disengage is essential to legitimacy.

Communities and participants should have:

- **Clear paths to exit without losing identity or accountability continuity.** Participants should be able to leave a space without being erased or forced to abandon their history, allowing accountability and learning to persist across contexts.
- **The ability to fork governance or norms when consensus breaks down.** When irreconcilable differences emerge, forking allows communities to diverge without coercion, preserving agency while limiting destructive conflict.
- **Emergency mechanisms to pause or disable systems causing systemic harm.** Kill switches or pauses provide a last-resort safeguard against cascading failure, runaway automation, or captured governance.

These safeguards limit the blast radius of governance failure, reduce incentives for capture, and make participation safer by design.

16.5 Cross-Zone Failure Containment

In a multi-zone environment, failures should be contained by default. DP1 assumes that trust loss, enforcement actions, and reputational signals are local unless explicitly propagated.

Communities define:

- **When signals remain zone-scoped.** Localizing consequences prevents minor or context-specific failures from unfairly affecting participation elsewhere.
- **When and how signals may propagate across zones.** Communities may choose to share certain signals across zones where risks overlap, but such propagation should be deliberate, transparent, and governed.
- **What thresholds justify broader impact.** Explicit thresholds help distinguish between isolated incidents and systemic harm, enabling proportional response.

This containment prevents cascading harm while preserving the ability to respond proportionally to serious or systemic abuse.

Failure is expected. Silent failure is not. A DP1-aligned system is one in which the degradation of trust conditions – a broken lineage, an unratified adaptation, an unresolved appeal backlog, a zone quietly loosening its proof requirements – is observable while it is still correctable.

17. Relationship to Other Desirable Properties

DP1 is foundational and cross-cutting. Many other Desirable Properties depend directly on the conditions it establishes.

In particular:

- **Properties related to safety, harm reduction, and abuse prevention** rely on durable accountability, shared memory, and zone-scoped enforcement to function effectively. Without persistent attribution and the ability to recognize patterns of behavior over time, safety mechanisms degrade into reactive moderation that fails to prevent repeat harm or coordinated abuse.

- **Properties concerning agency, consent, and autonomy** depend on federated identity, user-held credentials, and contestable governance so that participants can meaningfully choose how they engage, under what conditions, and with which authorities. Without these foundations, consent becomes nominal, exit becomes costly, and power asymmetries harden.
- **Properties addressing AI participation, automation, and alignment** require clear agent differentiation, asymmetric constraints, and binding accountability to prevent synthetic scale from overwhelming human judgment or distorting collective outcomes. DP1 establishes the conditions under which AI systems can participate without eroding trust or legitimacy.
- **Properties focused on collective intelligence, coordination, or governance** assume the existence of trustworthy participation, legible authority, and adaptive learning loops. Collective sensemaking and coordination cannot emerge where participants doubt who is acting, how decisions are made, or whether systems can learn from failure without capture.

More specifically:

- **DP2** supplies the levers that accountable actors hold. DP1 without DP2 produces accountability without agency, which is surveillance; DP2 without DP1 produces agency without attribution, which is impunity.
- **DP3** governs how trust conditions themselves change. Zone rules, escalation thresholds, and enforcement mandates are governance artifacts subject to DP3's tiering, revocability, and memory requirements.
- **DP4** bounds what the accountability layer may retain, infer, or transfer. Identity lineage must be reconstructable without becoming a surveillance corpus.
- **DP5** provides the namespace substrate. Persistent identifiers, personal identifiers, and controller bindings are what make identity continuity addressable rather than notional.
- **DP6** and **DP9** depend on anti-replay and sybil resistance, because value transfer and contribution rewards are the surfaces on which duplicated

identity pays best.

- **DP7** determines whether identity guarantees survive movement between systems, or degrade silently at integration boundaries.
- **DP11**, **DP12**, and **DP13** extend agent classification and asymmetric constraint into disclosure, policy execution, and containment for automated actors.
- **DP14** and **DP15** carry provenance and transparency for the records on which accountability rests.
- **DP20** determines who ultimately owns and can fork the identity and enforcement infrastructure, which is the final backstop against capture.

Weakness or ambiguity in DP1 propagates upward, undermining the effectiveness of other properties. Conversely, a strong DP1 enables the Meta-Layer to support more advanced coordination, safety, and governance capabilities without reverting to centralized control.

18. Non-Goals and Explicit Boundaries

DP1 deliberately defines the *conditions* for trust rather than attempting to solve all problems associated with identity, abuse, or governance on the Internet. Explicitly stating non-goals is essential to prevent scope creep, misinterpretation, and inappropriate application of this property.

DP1 does **not** attempt to:

- **Enforce real-name policies globally.** DP1 explicitly rejects the assumption that real-world identity disclosure is a prerequisite for trust. Mandatory real-name systems often increase risk, suppress participation, and centralize power without reliably preventing abuse.
- **Eliminate all abuse or deception.** No trust system can guarantee perfect safety. DP1 focuses on raising the cost of harm, preserving accountability, and enabling learning and repair, rather than promising total prevention.
- **Centralize identity, moderation, or governance.** DP1 is incompatible with architectures that concentrate authority in a single platform, provider, or enforcement body. Trust is treated as a plural, zone-scoped property rather than a global control mechanism.

- **Replace legal systems or law enforcement.** DP1 operates at the interface and governance layer. It does not supersede legal processes, nor does it attempt to adjudicate crimes or enforce jurisdictional law.

DP1 also does not:

- **Mandate a single proof-of-humanity mechanism.** The capability must be available system-wide; the mechanism remains a community choice, and multiple mechanisms may coexist or be superseded.
- **Establish a global reputation score.** Trust signals are zone-scoped by default. Cross-zone propagation is an explicit, governed act, not an emergent property of participation.
- **Prohibit anonymity.** Unattributed action may exist. What DP1 constrains is its ability to affect shared resources, rewards, governance, or amplification.
- **Treat delegation as a transfer of responsibility.** Agents, tools, and collaborators may act on a participant's behalf, but the accountable principal does not change by virtue of delegation.
- **Guarantee that identity guarantees survive every boundary.** Where continuity, provenance, or trust semantics degrade, DP1 requires that the degradation be signaled, not that it never occur.

Failure mode: **overreach**, where DP1 is invoked to justify mandatory identification, global enforcement, or surveillance infrastructure that its own non-goals exclude.

By naming these boundaries explicitly, DP1 remains adaptable across cultures, legal regimes, and communities, while resisting overreach or misuse.

19. Minimum DP1 Alignment

Minimum alignment is not a feature checklist. It is the threshold at which an identity system can be considered **enforceable, portable, and resistant to trivial abuse**.

A system that does not meet these conditions may function, but it cannot reliably sustain trust under scale, automation, or adversarial pressure.

At minimum, a system claiming alignment with DP1 must satisfy the following **irreducible conditions**:

19.1 Persistent Identity Continuity

- Actions MUST be bound to a persistent agent identifier that survives across sessions and contexts
- Identity resets MUST be rate-limited, detectable, or carry loss of accumulated privileges
- Systems MUST signal when identity continuity is broken or degraded

Failure mode: identity reset cycles that enable repeated exploitation of incentives and governance.

19.2 Verifiable Action Attribution

- All meaningful actions (content, transactions, moderation, automation) MUST be attributable to an agent
- Attribution MUST include sufficient provenance to audit origin and context
- Anonymous or unattributed actions MAY exist, but MUST be constrained from affecting shared resources or governance

Failure mode: untraceable actions that erode accountability and enable manipulation.

19.3 Anti-Replay and Non-Duplication Guarantees

- Identity-bound actions and credentials MUST NOT be reusable across systems without explicit lineage
- Systems MUST detect or prevent replay of contributions, credentials, or proofs
- Cross-system transfers MUST preserve attribution or clearly signal loss of guarantees

Failure mode: replay attacks that extract duplicate rewards, access, or influence.

19.4 Sybil Resistance Under Scale

- Systems MUST impose friction or constraints that make large-scale identity duplication costly or ineffective
- Mechanisms MAY include proof of humanity, rate limits, staking, or reputation weighting
- Systems MUST remain functional under coordinated identity attacks

Failure mode: sybil saturation overwhelming incentives, governance, or visibility.

19.5 Zone-Scoped Enforcement and Revocation

- Enforcement actions MUST be defined and executed within explicit zones
- Revocation, restriction, and recovery pathways MUST be visible and rule-based
- Global or opaque enforcement MUST NOT be the default

Failure mode: arbitrary or centralized enforcement that undermines legitimacy.

19.6 Human and AI Agent Differentiation

- Systems MUST visibly distinguish between human, AI, and hybrid agents at the interface level
- Automated agents MUST be subject to appropriate constraints (rate, scope, amplification)
- AI outputs MUST be bound to a responsible entity

Failure mode: indistinguishable agents enabling manipulation, impersonation, and synthetic dominance.

19.7 Identity Lineage and Memory

- Systems MUST maintain reconstructable identity lineage for actions, credentials, and governance events
- Breaks in lineage MUST be treated as risk signals, not neutral transitions
- Historical records MUST be tamper-resistant after defined edit windows

Failure mode: lineage loss enabling impersonation, laundering, or erasure of harmful behavior.

19.8 Contestability and Due Process Baseline

- Participants MUST have access to mechanisms for contesting enforcement actions
- Appeals pathways MUST exist for decisions affecting access, visibility, or reputation
- High-impact decisions MUST include human review or ratification

Failure mode: unchallengeable systems that degrade into opaque or captured governance.

These conditions define the **minimum viable enforcement layer for identity** in the Meta-Layer.

Partial implementations that omit continuity, attribution, anti-replay guarantees, or sybil resistance SHOULD NOT be considered aligned with DP1, regardless of authentication strength or interface design.

20. Open Questions and Future Work

DP1 establishes foundational conditions for trust, but it does not resolve all questions required for long-term interoperability, standardization, and global deployment. The following areas are intentionally left open for further research, experimentation, and community deliberation.

- **Standardization of proof-of-humanity mechanisms.** While DP1 requires the availability of proof of unique humanity as a system capability, it does not prescribe specific mechanisms. Multiple decentralized approaches already exist, such as Fractal ID and other proof-of-humanity systems, each with different tradeoffs around privacy, accessibility, cost, and resistance to gaming. Communities may choose the mechanisms that best fit their norms and risk profiles. Over time, however, it is likely that one or a small number of widely trusted proof-of-humanity systems will emerge for use at the Meta-Layer or Overweb level, providing a common baseline that communities are encouraged, but not required, to adopt. Open questions include how such proofs can remain privacy-

preserving, globally accessible, and adaptable over time, as well as how multiple proof systems might interoperate, be bridged, or be composed.

- **Cross-zone reputation portability.** DP1 assumes that accountability and trust signals are zone-scoped by default. Further work is needed to determine when and how reputation, sanctions, or trust signals should move across zones without creating unjust spillover effects or de facto global scoring systems.
- **Liability and responsibility models for autonomous agents.** As AI agents become more capable and autonomous, clearer models are needed for assigning responsibility across operators, deployers, tool providers, and communities. DP1 establishes binding to responsible entities, but does not yet resolve how liability should be apportioned in complex, multi-actor systems. Records should preserve a path to responsible human and organizational actors rather than treating the machine artifact as a substitute for responsibility; legal allocation remains subject to the applicable framework.
- **Thresholds for escalation across zones.** Communities will need shared patterns for deciding when local failures warrant broader response. This includes defining proportional thresholds, evidentiary standards, and governance processes for cross-zone escalation without undermining local autonomy.
- **Meta-Layer coordination mechanisms.** DP1 implies the need for shared coordination mechanisms capable of maintaining coherent context across human participants, AI agents, content objects, and communities. Such mechanisms would support converging profiles, scoped capabilities, and durable accountability without requiring centralized control. Emerging approaches to structured context exchange and agent coordination, such as Model Context Protocols (MCP), may offer useful design patterns for this layer when generalized beyond model runtime to sociotechnical actors. The design of any such Meta-Layer coordination protocol remains an open area of research and standardization. A minimal protocol for personal and corporate agents should specify observable traces, exchange formats, and distributed authentication across websites, interfaces, and independently operated agents (DP7, Section 12.9.1, Agent interaction contracts).

Further questions concern how a trust system is measured and how its guarantees behave under pressure:

- how to measure trust-system health beyond enforcement counts, for example appeal reversal rates, time-to-resolution, and the frequency of unratified adaptive changes
- how to represent identity continuity degradation in interfaces that ordinary participants can interpret at the moment it matters
- how to support legitimate persona separation without enabling evasion, given that both look similar from the outside
- how to preserve accountability lineage while honoring data minimization and deletion expectations under DP4
- how to sustain proof-of-humanity accessibility for participants without documents, stable devices, or reliable connectivity

These questions are not gaps in DP1, but signals of where future ML-Drafts and ML-RFCs may be required as the Meta-Layer matures.

21. Path Toward ML-RFC

This ML-Draft is intended as exploratory scaffolding rather than a finalized specification. Progression toward an ML-RFC should be guided by rough consensus, iterative refinement, and practical validation.

Key steps toward ML-RFC status include:

- **Soliciting broad community review and critique.** Feedback from implementers, civil society, governance practitioners, and researchers is essential to test assumptions and surface edge cases.
- **Identifying points of rough consensus.** Not all aspects of DP1 must be settled to advance. Emphasis should be placed on stabilizing core invariants such as action-bound accountability, zone-scoped trust, and contestability.
- **Clarifying implementation invariants.** Future drafts should distinguish clearly between invariant requirements and flexible design space, reducing ambiguity for builders while preserving pluralism.
- **Separating exploratory elements from normative commitments.** Concepts such as coordination protocols or adaptive intelligence should

mature through dedicated drafts before being incorporated normatively.

- **Promoting stable elements to ML-RFC status.** Once sufficient consensus and operational understanding exist, portions of DP1 may be advanced as ML-RFCs to serve as durable reference points for the Meta-Layer ecosystem.

Graduation criteria SHOULD include demonstrable evidence, not description alone:

- conformance tests for each condition under Section 19, including adversarial scenarios for replay, sybil saturation, and lineage loss
- reference schemas for agent identifiers, attribution records, zone profiles, and enforcement receipts
- evidence that identity continuity survives migration into at least one independent system, or that degradation is signaled where it does not
- evidence that appeals resolve within stated timelines and that reversal outcomes are observable
- evidence that adaptive recommendations were ratified, modified, or rejected by humans with a visible record

This progression reflects the Meta-Layer's commitment to transparency, accountability, and participatory standards development.

22. Closing Orientation

DP1 is the claim that trust can be built without requiring people to surrender who they are.

It refuses the two familiar bargains of the current web. The first offers safety in exchange for identification, and delivers surveillance. The second offers freedom in exchange for accountability, and delivers impunity. DP1 asserts that neither trade is necessary: identity can be plural and pseudonymous while actions remain durably attributable, and communities can enforce their own conditions without a central authority defining who counts.

When DP1 is strong, participation carries weight. Reputation accumulates because it cannot be trivially reset. Automation is welcome because it is legible and bounded. Enforcement is credible because it can be contested.

Communities can set their own thresholds because zones make difference composable rather than fragmenting.

When DP1 is weak, everything above it becomes performance. Governance votes that anyone can flood. Agency controls over flows no one can attribute. Ethical AI commitments with no responsible entity behind an output. Commerce where the same identity collects twice.

DP1 defines the conditions under which trust can emerge. Without it, the Meta-Layer becomes another surface. With it, the Meta-Layer becomes a place.

Continue the conversation on DP1 online →

DESIRABLE PROPERTY 2

DP2

Participant Agency and Empowerment

Power to the Participant – the Meta-Layer puts participants, not platforms, in control of how they show up, interact, and shape their online experience.



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp02

1. Purpose of This Draft

This draft articulates Desirable Property 2 (DP2) as the Meta-Layer's commitment that participants can meaningfully steer their digital lives. Beyond authentication (DP1) and governance (DP3), DP2 establishes that people and accountable agents hold real, usable power over presence, data flows, automation, and the conditions under which they are seen, acted upon, and counted.

DP2 responds to recurring failures of the contemporary web:

- **Agency theater**: settings and consent flows that do not change outcomes
- **Asymmetric literacy**: systems legible only to specialists while obligations bind everyone
- **Structural dependency**: exit and portability exist nominally but are costly or illusory
- **Delegated opacity**: agents act on a participant's behalf without durable, legible control

This draft guides implementation, governance design, and future ML-RFC development. It is exploratory scaffolding, not a finalized specification.

2. Problem Statement

Why "control" without capability fails.

For decades, platforms have described participants as "in control" while reserving decisive power for operators, opaque ranking systems, and unbounded automation. The result is not merely dissatisfaction; it is predictable harm: manipulation, lock-in, surveillance-by-default, and governance that responds to scale by narrowing what ordinary people can do or understand.

DP2 begins from a different premise: **agency is not a feeling; it is a property of systems**. A Meta-Layer earns the label human-first only if participants can **observe, redirect, and withdraw** from the forces that shape their experience – within the same zones where accountability (DP1) is enforced.

2.1 Agency vs. authorization

Authorization answers what a token allows. Agency answers whether a participant can **shape outcomes**: defaults, reach, automation, data use, and the rules that allocate visibility and risk.

Systems that conflate "logged in" with "empowered" routinely:

- Bundle consequential defaults behind "agree to continue"
- Hide material changes behind versioned policies
- Route impactful decisions to models or pipelines participants cannot inspect

DP2 separates authentication and authorization (DP1) from **participant-directed configuration of the lived interface**.

2.2 Empowerment as distributed capability

Empowerment is **capability + legibility + recourse**:

- **Capability**: participants can change outcomes (not just preferences)
- **Legibility**: participants can see how systems act on their behalf
- **Recourse**: participants can contest, reverse, or exit

A system lacking any one of these is not empowering, regardless of interface polish.

3. Threats and Failure Modes

Agency fails in patterned ways. The failure modes below are named throughout this draft as consequences of specific mechanism gaps; collected here, they form the adversarial model against which a DP2 implementation should be tested. None of them requires a malicious operator. Most emerge from ordinary optimization pressure, interface convenience, or the accumulation of integrations over time.

3.1 Agency theater

Interfaces expose settings that do not alter execution. The participant experiences choice; the system behaves identically.

Example: A "limit data use for personalization" toggle changes a display preference but not the ranking pipeline that consumes the same signals.

Why this matters: Simulated control is worse than absent control, because it suppresses the demand for real control.

3.2 Delegation drift

Agents act beyond the scope they were granted, either by accumulating permissions incrementally or by interpreting a mandate expansively.

Example: An assistant granted read access to a calendar begins composing and sending replies on the participant's behalf after a capability update.

Why this matters: Scope that is not continuously enforced is not scope; it is a suggestion.

3.3 Consent decay

Permissions persist beyond the participant's awareness, becoming "zombie consent" that no one remembers granting and no interface surfaces.

Example: A third-party integration authorized years earlier retains ongoing access with no expiry, review prompt, or visible record.

Why this matters: Consent that cannot be recalled cannot be revoked in any meaningful sense.

3.4 Coercive configuration

Defaults, flow design, and asymmetric friction steer participants toward choices that disadvantage them.

Example: Signup completes in one tap; cancellation requires locating a buried page, confirming twice, and waiting for an email.

Why this matters: Where entry and exit have unequal friction, the system has taken a position against the participant.

3.5 Automation overrun

Agent activity exceeds human capacity to observe, intervene, or revoke, nullifying agency without ever formally removing it.

Example: Delegated agents transact hundreds of times per hour; the participant's kill switch works, but only after the consequential actions have already settled.

Why this matters: Oversight that cannot keep pace with execution is not oversight.

3.6 Exit obstruction and degraded export

Leaving is technically permitted but practically infeasible, or the exported artifact is unusable elsewhere.

Example: An archive arrives containing content but no thread structure, permissions history, or stable identifiers.

Why this matters: Exit is the backstop for every other agency guarantee. When it fails, all remaining controls are granted at the operator's discretion.

3.7 Interop deception

Portability or integration is advertised, but core agency properties are silently lost in transit.

Example: A migration preserves posts but drops delegation states, so agents authorized in the source system silently gain or lose authority in the destination.

Why this matters: Undisclosed degradation is indistinguishable from misrepresentation.

3.8 Agency fragmentation and semantic drift

Participant choices do not persist across systems, or the same permission is interpreted differently in a new context.

Example: A "no automated action without confirmation" preference is honored in one tool and treated as advisory by a downstream integration.

Why this matters: Agency that stops at a system boundary is agency the participant cannot rely on.

3.9 Agency opacity

Participants cannot reconstruct what was done on their behalf, when, or under what authority.

Example: An automated decision changed a participant's visibility, but no log, summary, or attribution is available to review or contest.

Why this matters: Without auditability there is no recourse, and without recourse there is no agency.

3.10 Capture and collective overreach

Community-level mechanisms either concentrate power in a small group or suppress legitimate individual choices without pathways for challenge.

Example: A stewardship role accumulates permanent authority; or a zone-level setting silently overrides members' individual privacy configurations.

Why this matters: Collective agency must enhance individual agency, not substitute for it.

4. Core Principle

Agency is the ability to change outcomes, not merely configure preferences. Systems that do not preserve participant intent across automation, delegation, and scale do not provide agency.

Participant agency in the Meta-Layer is the combination of meaningful defaults, legible automation, durable delegation controls, and practical exit – enacted at the interface where people actually live.

Implications:

- Defaults favor the participant where stakes are asymmetric (data, reach, automation, payments), with zone-level calibration
- Automation is always **scoped**: time-bounded, purpose-limited, revocable, and attributable (DP1)
- Core agency paths (privacy, notifications, delegation, export, exit) remain reachable without specialized training

- **Collective agency is first-class**: communities can set norms and enforce them without stripping member autonomy (handoff to DP3, DP18–DP20)

5. Primary Mechanisms and Structural Conditions

DP2 is enacted through mechanism families that together convert stated control into enforceable control. Each addresses a distinct point at which agency is typically lost.

- **Presence and visibility controls** treat identity plurality and reach as configurable objects rather than fixed profile properties, so that pseudonymous participation is not defeated by accidental correlation or unsolicited amplification.
- **Default and friction design** assigns normative weight to what the system does when the participant does nothing, and distinguishes protective friction from friction engineered to obstruct understanding or exit.
- **The agency system layer** carries continuity, delegation integrity, consent durability, anti-coercion, cross-system semantics, and auditability across environments and time, so that participant intent survives movement.
- **Agency substrates** – purpose-limited processing, the agent delegation graph, and human-in-the-loop gradients – bind data use and automation to declared scope with a reachable interrupt.
- **Portability and exit guarantees** make leaving feasible in human time, with truthful disclosure of what is preserved, what degrades, and what cannot transfer.
- **Collective agency tools** allow communities to set and enforce norms through governed processes without silently nullifying individual controls.

The structural condition uniting these is *enforceability under movement and scale*. A control that holds in a single interface but fails on delegation, integration, or migration is not an agency mechanism; it is a display. DP2 therefore requires that every control either persist across a boundary or signal, at the boundary, that it no longer holds.

Several tensions are inherent to this design and are addressed rather than resolved, among them usability against configurability, automation against

control, and safety against paternalism. These are treated first, because every mechanism that follows is a position taken within them.

6. Tensions and Tradeoffs

6.1 Usability vs. complexity

Agency introduces configuration surfaces that can overwhelm. Hiding them removes control. DP2 requires **graduated disclosure**: simple defaults that are safe, with deeper controls accessible without specialized expertise.

6.2 Automation vs. control

Automation reduces effort but can displace agency. Participants must be able to answer:

- What did the agent infer?
- What authority does it have?
- How do I stop it?

DP2 requires **visible delegation scopes, renewal, and revocation** aligned with accountable binding (DP1).

6.3 Power-law attention markets

Even fair rules can reproduce inequality when attention is the currency. DP2 does not promise equal outcomes; it guarantees **equal access to the levers** that govern one's participation and visibility within a zone, and **transparent disclosure** when algorithmic allocation is in play (touchpoint DP14).

6.4 Safety vs. patronizing lockdown

Safety work can slide into infantilizing participants. DP2 pairs with DP1 to require that constraints be **proportionate, explainable, and contestable**, with pathways for competent self-determination inside high-trust zones.

7. Presence, Identity Plurality, and the Right to Shape Visibility

DP2 treats presence as something participants **sculpt**, not merely a profile object.

7.1 Plural identities, singular accountability

Participants may present differently across zones (DP1). Agency requires **per-zone controls** for visibility, linkage, and discoverability so pseudonymous participation is not undermined by accidental correlation.

7.2 Reach and amplification as explicit objects

When systems can amplify (boost, recommend, cross-post), amplification settings are **agency-bearing surfaces**: who may amplify me, under what proofs, with what caps? This is where DP2 meets DP1's asymmetric constraints for AI scale.

8. Defaults, Friction, and "Reasonable Participant" Design

8.1 Dangerous defaults are a governance bug

DP2 assigns normative weight to default selection: the burden of proof lies on whoever proposes a default that increases extraction, surveillance, or irreversible commitment.

8.2 Friction as protection, not punishment

Strategic friction (confirmations, cooling-off periods for irreversible acts) protects agency when stakes are high. DP2 distinguishes **protective friction** from **hostile friction** designed to prevent exit or understanding.

Protective friction should invite reflection on consequences and reaffirmation of intent. Its design should be assessed as part of the experience of agency: participants need to understand the prompt and retain the ability to exercise their own judgment.

8.3 Progressive disclosure without burial

Advanced controls may be layered, but never removed from accountability: search, assistive onboarding, and machine-readable policy summaries are

part of agency infrastructure.

8.4 Focus and attention controls

Interfaces should offer focus-enabling modes that reduce interruption and cognitive burden. These modes should make control understandable and accessible, with defaults oriented to participant attention and intent rather than engagement maximization.

8.5 Participant-arranged interface tools

Participants should be able to choose how multiple overlays coexist. An overlay is a tool that appears on top of a web page, such as an annotation panel, a discussion thread, or an AI assistant. These choices include which tools are pinned, expanded, reordered, or available on demand.

A shared tray can expose these choices without imposing a fixed screen position. A tool's own claim to priority does not override the participant's arrangement. For example, a participant could keep a fact-checking overlay pinned open while a shopping assistant stays collapsed until called (hypothetical). DP7 specifies the shared-interface pattern.

Participants should also be able to author, share, and remix navigational flows – guided sequences of pages and steps for tasks, learning, and collaborative work. A Navigator User Interface is a shared panel where participants choose these flows and adapt AI-guided exploration to their own purposes. For example, a teacher could publish a flow through sources on a local election, and students could adapt it for their own research (hypothetical).

9. Agency System Layer: Continuity, Delegation Integrity, and Enforceable Consent

Beyond interface controls and defaults, DP2 requires a coherent agency system layer that persists across environments, interactions, and time. This layer ensures that participant intent, consent, and control remain enforceable under scale, automation, and interoperability.

Agency is not simply the presence of controls. It is the ability to reliably change outcomes across systems without loss of intent, visibility, or

recourse.

9.1 Agency continuity across systems

Participant choices must persist across tools, zones, and integrations.

This requires:

- Preservation of consent, preferences, and delegation states across environments
- Explicit signaling when agency guarantees degrade or reset
- Protection against silent override of participant intent by downstream systems

A failure mode is **agency fragmentation**, where participant control is lost when moving across systems.

9.2 Delegation integrity and scope enforcement

Delegation must remain bounded, legible, and enforceable.

All delegated authority must be:

- Explicitly granted
- Scoped by capability, domain, and time
- Continuously visible to the participant
- Revocable without friction

Systems must prevent delegated agents from expanding scope beyond granted authority.

A failure mode is **delegation drift**, where agents act beyond intended scope without detection.

9.3 Consent durability and revocability

Consent must persist long enough to be meaningful, but remain revocable at all times.

This requires:

- Clear mapping between consent and system behavior
- Immediate or bounded-time revocation pathways

- Prevention of "zombie consent" where permissions persist beyond participant awareness

A failure mode is **consent decay**, where participants lose track of what they have authorized.

9.4 Anti-coercion and default integrity

Defaults must not be used to extract consent or steer behavior against participant interests.

Systems must:

- Prevent dark patterns and coercive flows
- Require explicit confirmation for high-impact or irreversible actions
- Surface when defaults materially affect outcomes

A failure mode is **coercive configuration**, where participants are nudged into decisions that undermine agency.

9.5 Cross-system agency semantics

Agency signals do not carry identical meaning across all systems.

Systems must:

- Signal when consent or delegation semantics change across contexts
- Prevent misinterpretation of permissions
- Allow participants to re-evaluate choices when entering new environments

A failure mode is **semantic drift**, where participant intent is misapplied across systems.

9.6 Agency memory and auditability

Participants must be able to reconstruct what they authorized, when, and why.

This includes:

- Logs of delegation, consent, and revocation events
- Visibility into system actions taken on behalf of the participant

- Tools for auditing past decisions and their consequences

A failure mode is **agency opacity**, where participants cannot understand or audit system behavior.

9.7 Agent inventory

Participants must be able to see, in one place, every agent they have delegated authority to and every agent they have engaged with that holds their consent, data, or standing permissions, across all the environments where they participate.

For each agent, the inventory shows:

- What it is and who is accountable for it (DP1, Section 8.3)
- What it was authorized to do: scope, domains, spend and rate limits, and data access
- When the authority was granted, when it expires, and when it last acted
- Where it operates: the environments, zones, and services that recognize its authority
- A control to pause, narrow, or revoke it, and a record of whether that revocation reached every place the agent operates

The inventory belongs to the participant, not to any single provider. It is assembled from the delegation and consent records that each environment must expose (Sections 9.1 and 10.2), and it is portable when the participant leaves (Section 11.1). An environment that cannot report an agent's authority to the participant's inventory must say so, and must not present that agent as fully under the participant's control.

Agents a participant has not authorized but that act upon them in a shared context (moderation agents, recommenders, counterparties' agents) are listed separately, so participants can tell agents working for them from agents working on them (DP13, Sections 2 and 9).

A failure mode is **delegation sprawl**, where a participant has granted authority to more agents, in more places, than they can recall, and so cannot meaningfully revoke any of it.

This agency system layer ensures that participant control is not an illusion created by interface design, but a durable property that persists under real-

world conditions.

10. Data, Automation, and Delegation: Agency Substrates

10.1 Purpose-limited processing

Collection and use are tied to **stated purposes with granular switches**, not monolithic "privacy" toggles (deep coupling to DP4).

Emotional and affective inference should be an explicit domain of consent. Participants should be able to understand and revoke the authorization, while the baseline protections for dignity and sensitive data continue to apply.

10.2 Agent delegation graph

For any automated or AI-mediated actor operating with participant intent, the system exposes:

- **Scope** (read/write domains, rate, spend limits where relevant)
- **TTL and renewal**
- **Attribution** to a responsible entity (see DP1, Section 11.3)
- A **kill switch** reachable from the primary interface layer

10.3 Human-in-the-loop gradients

Not every action needs a click, but **material actions** (payments, legal commitments, public attributions, irreversible posts) require explicit human confirmation unless a community zone defines a higher-automation norm with informed opt-in.

Systems MUST remain safe under automated delegation at scale. This includes resisting coordinated agent behavior, preventing silent escalation of authority, and ensuring that human override remains effective even under high-volume automated activity.

A failure mode is **automation overrun**, where agent activity exceeds human capacity to observe, intervene, or revoke, effectively nullifying participant agency.

11. Portability, Exit, and Interoperability as Agency Guarantees

Agency must survive movement. If a participant's control disappears at boundaries, the system is coercive by design.

11.1 Practical exit

Exit must be feasible in **human time** (hours or days for standard data classes). Stalling tactics, hidden dependencies, or degrading exports constitute agency violations.

Systems MUST:

- Provide complete, machine-readable exports for user-held data and artifacts
- Disclose exclusions (e.g., third-party licensed data) with clear rationale
- Preserve identity continuity signals where technically honest (DP1), or explicitly signal loss

Failure modes:

- **Exit obstruction**: artificial friction prevents leaving
- **Degraded export**: data is technically exported but unusable

11.2 Forking and continuity

Where communities fork norms or stacks (see DP1, Section 16.4), participants retain identity continuity and portable artifacts where technically honest, avoiding punishment for disagreement.

Systems SHOULD support:

- Portable credentials and attestations with provenance
- Migration guides and compatibility layers for common formats

Failure mode: **fork penalty**, where dissent results in loss of history or access.

11.3 Interoperability honesty

Interoperability claims MUST be truthful. If a system advertises portability or integration, it MUST specify:

- What is preserved (data, credentials, delegation states)
- What degrades (semantics, guarantees, rate limits)
- What is not transferable and why

Failure mode: **interop deception**, where portability is claimed but core agency properties are lost in transit.

12. Collective Agency and Community Tools

Individuals act within communities. DP2 requires that collective mechanisms enhance, rather than erase, individual agency.

Community agency includes the epistemologies and protocols embodied in its tools: which categories, evidence practices, and ways of organizing knowledge a community can author and revise. Shared interface compatibility should not require every community to adopt the same framework for knowing.

Communities MUST be able to:

- Define and publish mandates for stewardship roles (moderators, curators, treasurers)
- Set and revise community-level switches (e.g., human proof for governance votes) via governed processes (DP3, DP12)
- Audit decisions and reverse them through defined procedures

Systems MUST ensure:

- **Mandate transparency**: who can act, within what scope, for how long
- **Revocation pathways**: members can challenge or replace stewards
- **Proportional constraints**: community rules do not silently nullify individual controls

Failure modes:

- **Capture**: small groups entrench power and override member agency
- **Collective overreach**: community settings suppress legitimate individual choices without recourse

13. Governance, Accountability, and Agency Surfaces

DP2 is unusual among the Desirable Properties in that agency surfaces are not a supporting requirement but the property itself. A backend that faithfully enforces scoped delegation while exposing no way to see or change that scope has satisfied nothing. The condition DP2 imposes is therefore that the levers exist, that they are reachable without specialized expertise, and that pulling them demonstrably changes system behavior.

Participants must be able to:

- see, in one place, every active delegation: which agent, what scope, granted when, expiring when, acting on whose responsibility
- revoke any delegation and observe the effect within a bounded, stated time
- see what has been done on their behalf, with enough context to contest it
- see which defaults currently apply to them and which of those defaults materially affect data use, reach, or cost
- distinguish a protective confirmation from an obstruction, because the system states which stakes triggered the friction
- initiate export and exit, see what the export will and will not contain before requesting it, and receive it within human time
- understand, at a boundary, which of their choices will persist into the new context and which will not

Communities must be able to:

- publish stewardship mandates with scope and duration, and revise them through governed process rather than accretion
- set zone-level agency floors – for example, requiring confirmation for irreversible acts – that individual settings may strengthen but not silently weaken
- audit whether community-level switches are nullifying individual controls, and reverse them where they are
- preserve memory of why an agency rule was adopted and what it was responding to (DP3)
- retain the ability to fork or exit collectively, carrying member artifacts and continuity where technically honest

Example: A participant opens a single delegation view, sees that a shopping agent holds a spend limit expiring in nine days and a summarization agent with indefinite read access, revokes the second, and receives confirmation that the change took effect along with a log of what that agent had previously done.

Without these surfaces, the failure mode is **agency by assertion**, where a system's claims about participant control cannot be verified, adjusted, or contested by the participants those claims concern.

14. Incentives and Power Analysis

Agency is expensive to provide and profitable to withhold. Most agency failures are not the product of hostile design decisions but of ordinary optimization: the default that converts better ships, the export that no one is measured on degrades, the delegation scope that reduces friction expands.

The recurring pressures are:

- **Defaults as revenue.** Default selection is among the highest-leverage economic decisions a system makes. This is why DP2 places the burden of proof on whoever proposes a default that increases extraction, surveillance, or irreversible commitment, rather than treating defaults as neutral engineering choices.
- **Friction asymmetry as retention.** Where growth is measured and churn is penalized, signup friction is optimized away and cancellation friction is not. Parity between entry and exit is an incentive correction, not a courtesy.
- **Export as an unowned cost.** Portability generates no revenue and creates competitive risk, so exports degrade by neglect rather than intent. DP2 therefore requires that export usability be demonstrated in an independent system, not merely offered.
- **Automation as scope accumulation.** Broader agent authority produces better outcomes on most metrics, so delegation scope drifts outward absent enforcement. Time-bounding and continuous visibility exist to make drift visible rather than trusting restraint.
- **Legibility as competitive exposure.** Explaining how a system acts on a participant's behalf reveals mechanisms operators would prefer to keep

private. DP2 resolves this by requiring participant-legible explanation and audit paths rather than full internal disclosure.

- **Attention markets as concentration.** Where reach is the scarce good, fair rules still produce unequal outcomes. DP2 does not promise equal reach; it insists that access to the levers governing one's own visibility be equal within a zone.

Example: A system honors revocation faithfully but places the delegation view four levels deep, unlinked from any surface where agents actually act. No rule was broken; the lever was priced out of reach.

Why this matters: Power in agency systems accrues to whoever controls placement, default, and pace. DP2 treats those three as governed surfaces (DP3, DP12) precisely because they determine whether the formal guarantees are usable.

15. Community Signals Informing DP2

Recurring themes from public discourse (non-exhaustive) include both desires and tensions:

- Fatigue with consent theater and unreadable policies
- Demand for "show me what you're doing with my data right now" views
- Preference for AI copilots that **ask before acting** on the user's behalf
- Interest in portable reputation without panopticon scoring (tension with DP1)
- Desire for simplicity alongside real control (tension with complexity)

These signals reveal a core contradiction: participants want **power without overload**. DP2 addresses this through progressive disclosure, safe defaults, and auditability, rather than removing control.

16. Foresight and Failure Design

DP2 assumes that agency will erode rather than break. Systems rarely remove controls; they let controls fall out of correspondence with behavior as pipelines are added, agents gain capability, and integrations multiply. The

characteristic DP2 failure is a system whose settings page is accurate on the day it shipped and increasingly fictional thereafter.

Predictable degradation paths include:

- **Control drift**, where new processing pathways are added without being bound to existing participant switches, so the same toggle governs a shrinking share of actual behavior
- **Scope creep through capability updates**, where an agent's authority effectively expands because its tools expanded, without any new grant
- **Consent accumulation**, where authorizations pile up faster than any review cadence, until the delegation surface is too large to audit
- **Export decay**, where schema changes gradually strip meaning from an export that no one tests against an independent importer
- **Automation outpacing oversight**, where the interrupt still works but the actions it interrupts are already consequential
- **Collective substitution**, where community-level settings progressively absorb decisions that were previously individual, with no moment at which the change was contested

These paths compound. Scope creep raises the volume of automated action, which raises the audit burden, which reduces the probability that drift is noticed at all.

DP2 therefore requires safeguards designed in advance:

- **Delegation expiry by default**, so that inaction shrinks authority rather than preserving it
- **Consent review cadence**, surfacing accumulated grants at intervals rather than only on request
- **Boundary signaling**, so that degradation of agency guarantees is announced at the crossing rather than discovered afterward
- **Reachable interrupts** whose latency is stated and tested under load, not assumed
- **Export conformance testing** against at least one independent system, treated as a recurring obligation rather than a launch milestone
- **Postmortems for agency incidents** – coerced consent, automation overrun, failed revocation – linked to the default or scope change that

produced them

Failure is expected. Silent drift is not. A DP2-aligned system is one where the gap between what the controls claim and what the system does is observable while it is still small.

17. Relationship to Other Desirable Properties

- **DP1** supplies accountable actors; **DP2** supplies the levers those actors hold. Without DP1, agency collapses into anonymity games; without DP2, accountability becomes surveillance.
- **DP3** scales governance; DP2 ensures scale does not erase participatory steering.
- **DP4–DP6** ground agency in data, namespace, and commerce realities.
- **DP7–DP10**, **DP21** carry agency into experience, education, and modality.
- **DP11–DP13** bound AI so delegation does not swallow human steering.
- **DP14–DP17** make power auditable and sustainable.
- **DP18–DP20** turn agency into shared evolution of the stack.

Several of these couplings are load-bearing rather than thematic:

- **DP4** and DP2 share the consent stack. DP4 governs what may be collected, inferred, and retained; DP2 governs whether the participant can actually change those terms and observe the change. Purpose-limited processing is a DP4 mechanism exercised through a DP2 surface.
- **DP5** determines whether plural presence is possible at all. Per-zone visibility controls depend on identifiers that can remain distinct without being correlated by shared infrastructure.
- **DP6** makes economic surfaces agency-bearing: spend limits, subscription reversal, and agent budget mandates are DP2 controls enforced in a DP6 context.
- **DP7** determines whether agency survives movement. Interoperability honesty is the DP2-facing obligation that DP7 portability claims must satisfy.
- **DP13** provides the containment that makes delegation safe at scale. A kill switch is a DP2 affordance; the guarantee that it halts execution is a DP13 property.

- **DP20** is the final backstop: if the stack itself cannot be forked or collectively owned, every agency guarantee remains a grant that can be withdrawn.

18. Non-Goals and Explicit Boundaries

DP2 defines the conditions for agency; it does not promise universal outcomes.

DP2 does not:

- Guarantee equal outcomes or neutralize attention economics by fiat
- Eliminate all paternalistic protections in safety-critical zones (these must be labeled, bounded, and contestable)
- Replace DP1 accountability with unchecked "freedom to harm"
- Mandate a single UX globally; pluralism across zones is expected

DP2 also does not:

- Require full transparency of all system internals where doing so would enable exploitation; instead, it requires **participant-legible explanations and audit paths**
- Allow delegation to obscure responsibility; all automated action remains attributable (DP1)

Failure mode: **overreach**, where DP2 is interpreted to justify unsafe or unaccountable behavior.

19. Minimum DP2 Alignment

Minimum alignment is not a UX checklist. It is the threshold at which participant agency is **real, enforceable, and resistant to coercion, drift, and automation capture**.

A system that does not meet these conditions may expose controls, but it does not provide agency.

At minimum, a system claiming DP2 alignment MUST satisfy the following **irreducible conditions**:

19.1 Outcome-level control (not preference simulation)

- Participants MUST be able to change meaningful outcomes, not only surface preferences
- Core levers (visibility, data use, automation authority, exit) MUST directly affect system behavior
- Systems MUST NOT simulate control through settings that do not alter execution

Failure mode: **agency theater**, where interfaces imply control without changing outcomes.

19.2 Delegation visibility and revocation

- All automated or AI-mediated actions MUST be attributable and visible to the participant
- Delegation MUST include scope, duration, and authority limits
- Participants MUST be able to revoke delegation in real time or bounded time
- Participants MUST be able to see every agent they have delegated to or engaged, with its scope and status, from a single participant-held inventory (Section 9.7)
- Systems MUST report agent authority to that inventory, or clearly signal that they do not

Failure mode: **delegation opacity**, where systems act without legible authority or revocation.

19.3 Consent binding and enforcement

- Consent MUST be explicitly tied to system behavior
- Systems MUST enforce consent boundaries consistently across components and integrations
- Revocation MUST propagate across systems or clearly signal where it does not

Failure mode: **consent bypass**, where downstream systems ignore or reinterpret user intent.

19.4 Anti-coercion defaults

- Defaults MUST NOT materially disadvantage participants without explicit opt-in
- High-impact actions MUST require clear confirmation
- Systems MUST NOT use dark patterns to obtain or retain consent

Failure mode: **coerced consent**, where participants are steered into decisions against their interest.

19.5 Practical exit and portability

- Participants MUST be able to export their data and exit systems in reasonable human time
- Exit MUST NOT result in silent loss of identity continuity without explicit signaling (DP1)
- Systems MUST NOT impose artificial friction to prevent exit

Failure mode: **exit obstruction**, where users are technically allowed but practically unable to leave.

19.6 Cross-system agency integrity

- Participant choices MUST persist across integrations where technically feasible
- Systems MUST signal when agency guarantees degrade across contexts
- Downstream systems MUST NOT silently override upstream participant intent

Failure mode: **agency fragmentation**, where control is lost across system boundaries.

19.7 Auditability of system behavior

- Participants MUST be able to inspect what actions were taken on their behalf
- Systems MUST provide logs or summaries of automated decisions and their effects
- Critical actions MUST be reconstructable for dispute or review

Failure mode: **agency opacity**, where participants cannot understand or challenge system behavior.

These conditions define the **minimum viable agency layer** of the Meta-Layer.

Partial implementations that omit outcome control, delegation integrity, consent enforcement, or exit MUST NOT be considered aligned with DP2.

20. Open Questions and Future Work

- Portability vs. abuse: preventing weaponized export while honoring exit (interfaces with DP1 memory models)
- Legibility budgets: how much system behavior can be made comprehensible without overload; role of machine summaries vs. audits (DP14–DP15)
- Collective overrides: when may a community limit individual agency for safety without capture?
- Cross-zone identity correlation: separation vs. pressure for unified reputation
- Economic agency: tipping, subscriptions, and paid reach as agency-bearing surfaces (touchpoint DP6)

Further questions concern how agency is expressed, timed, and measured at the edges of a system:

- How to express delegation scope in terms participants can evaluate, given that capability boundaries are natural to engineers and opaque to nearly everyone else
- What revocation latency is acceptable for which action classes, and how that latency should be stated rather than left implicit
- How to keep an agency audit trail useful without turning it into a surveillance record of the participant's own behavior (tension with DP4)
- How to detect control drift automatically, so that a new processing pathway unbound to any existing switch is flagged rather than silently shipped
- How agency guarantees should degrade when a delegated agent operates in a system that does not implement DP2 at all

- How to measure agency health at the ecosystem level – revocation success rates, export usability, default asymmetry – without reducing it to a compliance score

21. Path Toward ML-RFC

Advancement from ML-Draft to ML-RFC should demonstrate that agency is not only described but **operationally verified**.

Key steps:

- Community review with builder and civil-society lenses
- Reference implementations of delegation graphs, revocation, and exit flows
- Conformance tests for **minimum alignment** (see Section 19) including adversarial scenarios (automation overrun, consent bypass, interop deception)
- Clear separation of **normative invariants** vs. **design space**
- Publication of audit reports demonstrating real-world behavior under load

Graduation criteria SHOULD include:

- Evidence that participants can revoke delegation and observe effect within bounded time
- Evidence that exports are usable in at least one independent system
- Evidence that automated actions are attributable and auditable end-to-end

22. Closing Orientation

DP2 asserts that participants are not merely subjects of systems, but operators within them.

When agency is real, people can:

- change outcomes
- understand system behavior
- delegate safely

- exit without penalty

When agency is simulated, systems accumulate hidden power: defaults decide, automation acts, and participants absorb the consequences.

DP2 is the commitment that control is not inferred, but **demonstrable**.

It is the difference between having settings and having a steering wheel.

. . .

DP1 asks who may act with integrity. DP2 asks whether participants hold the wheel – or only the liability.

Continue the conversation on DP2 online →

DESIRABLE PROPERTY 3

DP3

Adaptive Governance Supporting an Exponentially Growing Community

Scalable systems, sustainable communities – as the Meta-Layer expands, its governance must evolve, staying decentralized, fair, and responsive to the community that builds it.



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp03

1. Purpose of This Draft

This draft articulates Desirable Property 3 (DP3) as the condition under which governance scales with participation and capability without collapsing into centralized fiat, procedural paralysis, or symbolic participation.

DP3 defines how the Meta-Layer maintains legitimate, timely, and contestable rule-setting as communities, tools, and AI capabilities expand. It sits between DP1 (accountability), DP2 (participant agency), and DP12 (community governance of AI), and acts as the bridge between human deliberation and executable policy.

Governance is the system's primary control surface under conditions of scale, automation, and interoperability. When governance fails at system boundaries or under rapid change, every other property becomes unstable or performative.

If DP3 is weak, predictable failures follow: capture by early insiders, rubber-stamp councils, unbounded operator discretion, reform paralysis as conditions change, and AI-mediated scale overwhelming human governance loops.

DP3 does not prescribe a single voting system, constitution, or DAO pattern. It defines the minimum conditions under which governance remains adaptive, legible, and legitimate under exponential growth.

Governance is the control layer of the Meta-Layer. Systems that appear governed locally but fail across scale, speed, or system boundaries will be captured or bypassed.

2. Problem Statement

In today's web, governance consistently lags behind scale.

Communities begin with informal norms, moderators, and shared expectations. As participation grows, those structures fracture. Decision-making becomes opaque, centralized, or too slow to respond to real-time harms.

This produces recurring failures:

- scale shock, where participation outpaces moderation, policy, and appeals capacity

- frozen rules that fail to adapt to new behaviors, technologies, or adversaries
- governance theater, where surveys or advisory groups exist without decision power
- emergency centralization, where temporary powers become permanent
- invisible delegation, where operators change rules without traceable ratification

These failures are structural. Growth without governance capacity turns open systems into extractive or chaotic systems.

DP3 reframes governance as adaptive infrastructure: systems that can change at the speed of reality without losing legitimacy.

3. Threats and Failure Modes

3.1 Capture by concentrated stakeholders

Early contributors, large stakeholders, or sponsors lock rules that preserve their advantage.

Example: A founding team retains veto over all governance proposals despite community votes.

Why this matters: Governance without credible contestation becomes ownership theater.

3.2 Procedural overload

Governance becomes too slow to respond to urgent issues.

Example: A coordinated harassment campaign spreads while proposals wait weeks for quorum.

Why this matters: Governance must distinguish between deliberation and response.

3.3 Governance lag behind AI capability

AI systems evolve faster than governance cycles.

Example: Automated agents exploit a loophole for days while policy review is scheduled monthly.

Why this matters: Static governance cannot contain dynamic systems.

3.4 Scale without representation

New participants lack meaningful voice while incumbents dominate.

Example: Global users cannot participate in English-only governance calls.

Why this matters: Adaptation requires inclusive participation surfaces.

3.5 Symbolic decentralization

Governance appears decentralized but is controlled off-chain or off-process.

Example: Votes occur, but operators retain unilateral execution authority.

Why this matters: Legitimacy depends on alignment between process and power.

3.6 Emergency power drift

Temporary powers become permanent through inertia.

Example: Crisis moderation tools remain active without review or sunset.

Why this matters: Speed must not erode accountability.

3.7 Governance fatigue and disengagement

Participants disengage due to complexity or lack of impact.

Example: Only a small group consistently participates in governance decisions.

Why this matters: Low participation increases capture risk.

3.8 Cross-system governance divergence

The same rule is enforced differently, or not at all, depending on which tool, integration, or environment a participant is using.

Example: A zone prohibits automated amplification, but an integrated client applies no such constraint, so the rule holds only where participants

happen to be looking.

Why this matters: Governance that stops at an integration boundary invites arbitrage. Participants migrate to the least-governed surface, and the rule becomes advisory in practice while remaining binding on paper.

4. Core Principle

Adaptive governance in the Meta-Layer means authority evolves alongside scale, capability, and system interconnection while remaining visible, contestable, and bounded.

Governance must function not only within a single community or tool, but across the systems, environments, and contexts in which rules are applied, enforced, and experienced.

Rules must be able to change without becoming arbitrary. Emergency actions must be possible without becoming permanent. Participation must be distributed without becoming incoherent.

Example: A governance system defines standard decision cycles, emergency pathways with automatic expiration, and public policy diffs tied to incidents.

What this feels like: The system learns in public, without rewriting rules silently.

Without this: Growth becomes disenfranchisement disguised as efficiency.

5. Primary Mechanisms and Structural Conditions

5.1 Governance Layer: Execution, Memory, and Control

Adaptive governance requires more than the ability to change rules. It requires the ability to express, enforce, observe, and evolve those rules as part of a continuous operational system that persists across environments.

In many systems, governance fails not because rules are absent, but because they are not bound to behavior. Decisions exist as documents or discussions, while actual system behavior is shaped elsewhere by incentives, defaults, or hidden control layers.

DP3 therefore requires a shared governance layer composed of primitives that allow governance to operate as infrastructure rather than aspiration.

5.1.1 Policy objects

Governance rules must be represented as structured, versioned objects that can bind to runtime systems.

A policy object includes:

- scope: where the rule applies
- conditions: what triggers the rule
- constraints: what is allowed or prohibited
- enforcement bindings: how the rule executes

This allows governance to move from agreement to execution, and aligns directly with DP12's requirement for executable policy.

5.1.2 Governance receipts

Every governance action must produce a verifiable record.

A governance receipt includes:

- who proposed and approved a decision
- what changed
- when it changed
- what systems were affected
- what enforcement state was applied

These receipts allow participants and auditors to trace how governance decisions translate into system behavior, and connect governance to DP15 (security and provenance).

5.1.3 Governance diffing and versioning

Governance must evolve through visible change.

Participants must be able to see:

- what changed between rule versions
- why it changed
- what effects resulted

Silent rule replacement erodes legitimacy. Visible diffing preserves continuity and enables governance learning over time.

5.1.4 Zone governance profiles

Each community or interaction context operates under a defined governance profile.

A zone governance profile includes:

- active policy objects
- enforcement modes
- participation structures
- escalation pathways
- incentive constraints

This allows governance to adapt to context while remaining portable across systems, aligning with DP4 (data boundaries) and DP20 (community ownership).

5.1.5 Enforcement hooks

Governance must bind to systems that can enforce it.

These include:

- AI agent constraints (DP13)
- moderation systems
- access controls
- interaction limits

Without enforcement hooks, governance decisions remain advisory and are overridden by underlying system behavior.

These primitives do not replace governance processes. They make them operational. The mechanisms that follow operate through this layer, ensuring that decisions are not only made, but executed, observed, and revised within a coherent system.

5.2 Tiered decision systems

Not all decisions carry the same weight, risk, or urgency. Treating all governance actions as equivalent either slows the system to paralysis or opens

pathways for low-threshold capture of critical decisions.

DP3 requires that governance systems define clear decision tiers, each with appropriate processes, thresholds, and timelines. Routine decisions should be fast and low-friction. Significant decisions should involve deliberation and multi-stakeholder input. Existential decisions should require high thresholds and extended review.

A key failure mode is process flattening, where all decisions are routed through the same mechanism, allowing either trivial actions to clog governance or critical actions to slip through without sufficient scrutiny.

5.3 Delegation with revocability

As systems scale, direct participation in every decision becomes impossible. Delegation is therefore necessary, but without revocability it becomes a vector for capture.

DP3 requires that delegated authority be explicit in scope, time-bound where appropriate, and revocable by participants. Delegation must remain a tool of coordination, not a permanent transfer of power.

A failure mode is silent entrenchment, where delegated roles accumulate authority over time without clear pathways for removal or reassignment.

5.4 Emergency pathways with sunset

Governance systems must be able to respond rapidly to emergent threats, but speed introduces the risk of unbounded authority.

DP3 requires that emergency actions be clearly declared, automatically expire, and undergo post-hoc review and ratification. This creates a bounded exception rather than a precedent for permanent control.

A common failure mode is emergency normalization, where temporary powers persist due to inertia or lack of review, gradually shifting governance toward centralized control.

5.5 Governance memory

Governance decisions do not occur in isolation. They are part of a continuous process of learning, adaptation, and correction.

DP3 requires that all decisions be linked to their context, including prior versions, triggering events, dissenting views, and measurable outcomes. This creates continuity, supports accountability, and prevents repeated cycles of the same failures. Without memory, governance resets continuously and cannot improve.

A failure mode is historical amnesia, where prior decisions and their consequences are lost, forcing communities to relearn the same lessons under new conditions.

5.6 Capacity-aware governance

Governance is constrained not only by rules, but by the human and operational capacity required to execute them.

DP3 requires that governance systems explicitly provision for moderation, appeals, translation, and accessibility. Without this, governance becomes symbolic, with rules that cannot be enforced or contested in practice.

A failure mode is capacity illusion, where systems appear governed on paper but lack the resources to implement or uphold decisions.

5.7 Adversarial foresight

Governance systems must anticipate how they will be attacked or manipulated, rather than reacting only after failure occurs.

DP3 requires that communities model adversarial scenarios such as capture attempts, AI-driven manipulation, and scale shocks. This allows governance structures to incorporate safeguards before vulnerabilities are exploited.

A failure mode is reactive governance, where systems adapt only after harm occurs, often at higher cost and with reduced trust.

Foresight should examine who controls information channels and whose incentives shape the evidence reaching decision-makers. Conflicts of interest can compound across networks of institutions, even when individual conflicts are disclosed. Independent information and foresight channels help make those patterns contestable; centralized filtering of human speech is not by itself a remedy for conflicted governance.

5.8 Interoperable governance artifacts

Governance must not be confined to a single tool or platform. Policies, decisions, and governance structures must be able to move across systems without losing meaning.

DP3 requires that governance artifacts be exportable, comparable, and portable. This enables communities to fork, migrate, or integrate without resetting their governance systems.

A critical failure mode is governance lock-in, where rules exist only within a specific platform, making exit or replication impractical.

5.9 Continuous feedback loops

Governance cannot rely solely on periodic voting cycles. Systems must incorporate ongoing signals from participation, behavior, and outcomes.

DP3 requires continuous feedback mechanisms that inform governance in near real-time, allowing systems to adapt before issues become systemic failures.

A failure mode is episodic governance, where decisions are made in isolation from evolving conditions, leading to lag and misalignment.

5.10 Human-AI coordination boundaries

AI can significantly augment governance by processing information at scale, but it also introduces risks of manipulation, opacity, and overreach.

DP3 requires clear boundaries for AI participation, including roles in summarization, simulation, and pattern detection, while reserving material decisions for human ratification.

Governance scale depends on agent speed, reach, and interacting capabilities as well as participant count. Advanced systems may support analysis, monitoring, and infrastructure under defined mandates, while the boundary between assistance and legitimate governing authority remains explicit.

A failure mode is automation creep, where AI systems begin to effectively determine outcomes without explicit authorization or oversight.

5.11 Representation mechanisms

As participation scales globally, governance must ensure that diverse perspectives are meaningfully included.

DP3 requires mechanisms that support geographic diversity, language accessibility, and asynchronous participation. Representation must reflect the actual composition of the community, not just those able to engage in specific formats.

Freedom of expression depends in part on open information infrastructure through which people can contribute without a few actors monopolizing access or visibility. Protection of voice should operate alongside visible community rules and contestable moderation, rather than depend on a promise of an entirely ungoverned space.

A failure mode is structural exclusion, where governance participation is limited by language, time zones, or access constraints, concentrating power among a narrow subset of participants.

6. Scalable Governance Patterns

The mechanisms above describe what adaptive governance requires. This section describes recurring structural patterns through which those requirements are met in practice, and the scale conditions under which each pattern begins to fail. DP3 does not mandate any single pattern; it requires that whichever pattern a community adopts remain visible, contestable, and bounded as the community grows.

Scale does not degrade governance gradually. It degrades governance at thresholds, where a structure that worked at one order of magnitude stops working at the next. The patterns below are organized around those thresholds.

6.1 Subsidiarity and nested zones

Decisions are made at the smallest scope competent to make them, with escalation only where effects exceed that scope. Zone governance profiles are the mechanism; subsidiarity is the principle governing their composition.

This pattern absorbs growth by adding zones rather than adding participants to a single deliberative body. A community of ten thousand does not deliberate as ten thousand; it deliberates as many bounded contexts with defined escalation pathways between them.

Scale condition: subsidiarity fails when escalation thresholds are undefined, because every local dispute becomes a candidate for global attention. It also fails when zones proliferate faster than governance memory can link them, producing incoherence rather than autonomy.

6.2 Rotation, term limits, and sortition

Stewardship roles carry defined terms, with rotation or random selection used to prevent the accumulation of informal authority. Communities have consistently requested this (see *Community Signals Informing DP3*), and it directly counters silent entrenchment.

Rotation trades continuity for contestability. Sortition trades expertise for representativeness. Both are defensible; what DP3 requires is that the trade be explicit and that the resulting mandate be published with scope and duration.

Scale condition: rotation fails where the pool of willing, capable stewards is smaller than the rotation schedule requires, producing either burnout or nominal rotation among the same few actors. Capacity-aware governance is the precondition for rotation to be real.

6.3 Deliberative sampling

Rather than seeking mass participation on every question, a representative sample is convened, resourced, and given real decision weight on a specific matter. This addresses governance fatigue directly: it asks less of most participants while asking substantially more of a few, with support.

Scale condition: sampling fails without translation, asynchronous participation, and compensation, because the sample will otherwise reflect who can afford to participate rather than who is affected. It also fails where the sample's output is advisory in practice, which reproduces governance theater at higher cost.

6.4 Progressive thresholds

Decision thresholds scale with the reversibility and blast radius of the decision rather than with its category. A change that can be undone in an hour requires less than one that cannot be undone at all.

This pattern operationalizes tiered decision systems by making the tier a function of consequence rather than of nominal type, which resists the common failure where a consequential change is routed through a routine pathway because of how it was labeled.

Scale condition: progressive thresholds fail where reversibility is asserted rather than tested. A change believed reversible but not actually reversible receives the wrong threshold.

6.5 Circuit breakers and staged rollout

Rules and system changes are deployed to a bounded population or zone before general application, with automatic halt conditions if defined harm indicators trigger. This converts governance decisions into observable experiments with bounded downside.

Scale condition: staged rollout fails where the staging population is unrepresentative, or where halt conditions are defined only for harms already anticipated. Adversarial foresight is what makes the halt conditions meaningful.

6.6 Forkability as a governance constraint

The credible ability to fork, carrying governance artifacts, memory, and member continuity, disciplines incumbents without requiring any fork to occur. Interoperable governance artifacts are the mechanism; forkability is the effect.

Scale condition: forkability fails when governance artifacts are portable in principle but the community's identity, data, or economic relationships are not. Exit that costs everything is not exit, and its disciplining effect disappears.

6.7 Automation as instrumentation, not authority

AI and automated systems are used to observe governance at scale (detecting drift, surfacing dissent, summarizing deliberation, flagging capacity shortfalls) while material decisions remain human-ratified.

The productive framing is that automation extends the *sensing* capacity of governance without extending its *deciding* capacity. This is what makes governance able to notice, at scale, the conditions it must respond to.

Scale condition: this pattern fails through automation creep, where an advisory signal becomes the operative decision because no human has capacity to review it. The boundary must be enforced by structure, not by intention.

6.8 Vicariance and protected incubation

Vicariance is temporary, partial separation that gives fragile social, technical, or cultural designs room to develop before confronting dominant actors and wider competition. Communities should be able to incubate alternatives and choose gradual integration, preserving diverse paths of emergence.

AI-mediated recommendation, translation, or outreach should respect the community's chosen dissemination boundaries instead of silently dissolving its incubation conditions. Insulation should remain reviewable and compatible with exit and accountability; it is not a justification for permanent enclosure. DP8 provides the zone setting and DP12 governs its enforceable boundaries.

6.9 Composing patterns

These patterns are not alternatives. A community operating at scale will typically combine nested zones with rotation inside each zone, progressive thresholds across zones, staged rollout for consequential changes, and automated instrumentation throughout. What DP3 requires of any composition is that the resulting system produce receipts, maintain memory, expose diffs, bound emergencies, and remain forkable.

A failure mode specific to composition is **pattern accretion**, where structures are added in response to successive crises without ever being retired,

until the governance system is too complex for participants to navigate or contest. Governance complexity is itself a capture surface. Periodic simplification is a governance obligation, not housekeeping.

7. Governance, Accountability, and Agency Surfaces

Governance surfaces determine whether participants can meaningfully understand, influence, and contest the rules that shape their environment. Without these surfaces, governance becomes opaque, unchallengeable, and ultimately extractive.

In many systems, governance exists in theory but not in practice: decisions are made elsewhere, authority is unclear, and appeals are ineffective or absent. DP3 requires that governance be experienced as a visible and navigable system, not an abstract promise.

Without these surfaces, governance loses legitimacy. Participants may continue to engage, but without trust, recourse, or real influence, participation becomes performative rather than constitutive.

Participants must be able to:

- see who holds authority at any moment
- understand how decisions are made
- appeal decisions within defined timelines
- track changes in rules over time

Communities must be able to:

- update governance structures without restarting from zero
- audit whether decisions were executed as approved
- fork governance when legitimacy breaks

8. Incentives and Power Analysis

Adaptive governance fails when incentives quietly outrun rules.

In many systems, decision procedures are visible while the forces shaping outcomes remain hidden. Optimization targets, funding dependencies, and

growth pressures steer behavior in ways governance cannot easily detect or correct.

DP3 requires that incentive structures be treated as first-class governance objects.

This includes making visible:

- what metrics systems optimize for (engagement, retention, revenue, safety)
- how those metrics influence policy enforcement and prioritization
- where economic or reputational rewards create pressure to bypass rules

Example: A platform publishes moderation policies that prioritize safety, but internal ranking systems reward engagement spikes. Harmful content persists not because rules are absent, but because incentives contradict them.

Why this matters: Governance that does not act on incentives will be bypassed by them.

DP3 therefore expects governance systems to:

- expose optimization targets where they materially affect outcomes
- allow communities to constrain or reshape those targets within zones
- link policy objects to incentive conditions where appropriate (e.g., disallow reward for rule-violating behaviors)

When incentives and governance are aligned, systems become self-reinforcing. When they diverge, governance becomes symbolic.

Decentralization should be evaluated by its effects on participants and communities. Incentive review should ask whether a proposed economic or token design rewards zero-sum behavior, concentrates control, or undermines the shared purpose, rather than treating distributed architecture alone as evidence of benefit.

9. Community Signals Informing DP3

Across communities and platforms, recurring signals point to a shared breakdown between scale and governance:

- frustration with decisions made by unseen operators without traceable process
- demand for term limits, rotation, and clearer accountability for stewards
- desire for asynchronous, multilingual participation that reflects global communities
- concern that AI and automation are overwhelming human moderation and deliberation capacity
- fatigue with feedback channels that do not result in observable change

These signals are not isolated complaints. They indicate structural gaps in how governance adapts, records, and responds at scale.

DP3 treats these signals as design inputs, not after-the-fact feedback.

10. Foresight and Failure Design

Adaptive governance must assume adversarial pressure, rapid scale changes, and technological acceleration. Failures do not typically occur as singular events, but as gradual degradation across multiple dimensions: participation, enforcement, legitimacy, and coherence.

Common failure paths include:

- coordinated capture attempts by concentrated stakeholders
- automated or AI-driven manipulation of governance processes
- overload of moderation and review capacity during rapid growth
- exploitation of gaps between policy definition and enforcement

These failures often compound. For example, as participation grows, capacity gaps emerge. These gaps increase reliance on delegation or automation, which may introduce new vectors for manipulation. Over time, trust erodes, participation declines, and governance becomes easier to capture.

Cross-system environments introduce additional risks:

- governance rules diverging across tools and integrations
- inconsistent enforcement depending on where interactions occur
- loss of governance memory or context during migration

DP3 therefore requires designing safeguards in advance:

- circuit breakers for pausing high-risk processes
- escalation pathways for urgent interventions
- public postmortems linking incidents to policy changes

Governance must be able to detect not only discrete failures, but slow-moving drift toward illegitimacy.

Failure is expected. Silent failure is not.

11. Relationship to Other Desirable Properties

DP3 operates as the structural backbone for governance across the Meta-Layer.

- DP1 ensures actions within governance are attributable and contestable
- DP2 enables participants to exercise agency through delegation and participation
- DP4 constrains how data can be used within governance decisions
- DP6 and DP9 shape the economic and incentive context in which governance operates
- DP7 enables portability of governance artifacts across tools
- DP11–DP13 define how AI behavior is disclosed, governed, and contained
- DP12 provides the execution layer that binds governance to runtime systems
- DP15 ensures governance actions are provable and auditable
- DP17 ensures governance has sustained resources
- DP20 defines who ultimately owns and can fork governance systems

Two couplings deserve emphasis, because DP3 fails first at their boundaries.

The first is **DP17**. Capacity-aware governance is not a procedural nicety; it is an economic dependency. Moderation, appeals, translation, and review are staffed activities, and a governance system that scales its rules faster than its funded capacity produces the capacity illusion described above. DP3's requirement to provision for enforcement is therefore a claim on DP17's sustainability mechanisms.

The second is **DP23**. Representation mechanisms depend on linguistic access. A governance system that deliberates in one language while binding participants across many has already excluded most of those it governs, regardless of how open its formal process is. Asynchronous, multilingual participation is a DP3 requirement met through DP23 infrastructure.

DP3 does not stand alone. It coordinates these properties into a functioning system.

12. Non-Goals and Explicit Boundaries

DP3 does not:

- guarantee optimal or unanimous outcomes in all decisions
- eliminate the need for expertise, stewardship, or delegated authority
- replace legal governance systems or jurisdictional requirements
- mandate a specific voting, DAO, or constitutional model

DP3 also does not:

- require that every participant participate in every decision; delegation and sampling are legitimate responses to scale, provided they remain revocable and visible
- treat decentralization as an end in itself; a widely distributed process that produces unaccountable outcomes fails DP3 as surely as centralized fiat does
- prohibit emergency authority; it requires that such authority be declared, bounded, and automatically expiring
- prescribe how communities weight votes, reputation, or stake, only that the weighting be visible and contestable
- guarantee that governance decisions will be correct, only that they will be traceable, reviewable, and reversible

Failure mode: **procedural fetishism**, where DP3 is invoked to justify processes that generate legitimacy signals without producing enforceable, revisable decisions.

DP3 defines conditions for adaptive governance. It does not prescribe a single implementation.

13. Minimum DP3 Alignment

A DP3-aligned system should, at minimum:

- define decision tiers with corresponding processes and thresholds
- bind governance decisions to enforceable mechanisms (via policy objects and enforcement hooks)
- produce governance receipts for material changes
- maintain visible versioning and diff history for rules and policies
- include bounded emergency pathways with automatic sunset and review
- provide appeal and contestation pathways with defined timelines
- plan for governance capacity (moderation, review, translation) proportional to scale

It should further:

- preserve governance memory linking decisions to triggering events, dissent, and observed outcomes
- publish stewardship mandates with explicit scope and duration, and support revocation of delegated authority
- export governance artifacts in a form another system can interpret, so that forking is practically available
- signal explicitly where governance guarantees do not hold across an integration or system boundary
- keep AI participation in governance advisory, with material decisions human-ratified and the ratification recorded

Partial compliance that omits execution, memory, or auditability should not be treated as alignment.

14. Open Questions and Future Work

Key open questions for adaptive governance include:

- how to achieve Sybil-resistant participation without excluding legitimate users (DP1)
- how to scale governance participation without overwhelming contributors

- how to integrate AI assistance in governance while preventing manipulation or capture
- how to balance global governance coherence with local autonomy and cultural context
- how to measure governance health beyond participation counts (e.g., decision latency, reversal rates, appeal outcomes)
- how to support forking and recomposition of governance systems without loss of continuity

Further questions concern the maintenance of a governance system over time, where accretion, underfunding, and illegibility accumulate faster than any single decision:

- how to retire governance structures that have outlived their purpose, given that accretion is easier than simplification and complexity favors incumbents
- how to fund capacity counter-cyclically, so that appeals and moderation scale with incident volume rather than with revenue
- how to detect governance drift automatically without turning detection into a new unaccountable authority
- what evidentiary standards should govern cross-zone escalation, so that propagation is proportionate rather than reputational
- how to make policy diffs legible to non-specialists, since a visible change that no one can interpret provides transparency without contestability

These are not reasons to delay implementation. They are areas for iterative experimentation within visible governance systems.

15. Path Toward ML-RFC

Advancing DP3 toward ML-RFC requires:

- standardizing formats for governance receipts, policy diffs, and audit trails
- developing reference implementations of tiered governance and emergency pathways

- testing governance loops in live communities with varying scale and risk profiles
- aligning governance artifacts with identity, data, and interoperability standards

Progress should be demonstrated through working systems, not only conceptual agreement.

16. Closing Orientation

DP3 is the claim that governance can scale without losing legitimacy.

It rejects the tradeoff between speed and accountability, and between participation and coherence.

When adaptive governance is real, communities do not outgrow their ability to govern themselves.

When it is absent, growth concentrates power, erodes trust, and replaces coordination with control.

Continue the conversation on DP3 online →

DESIRABLE PROPERTY 4

DP4

Data Sovereignty and Privacy

Your data, your terms: meaningful control over use, sharing, and protection, with visible limits and accountable follow-through.



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp04

1. Purpose of This Draft

This draft articulates Desirable Property 4 (DP4) as the condition under which participants and communities can meaningfully govern data about themselves and their activity in the Meta-Layer.

DP4 does not treat privacy as a settings menu, a compliance ritual, or a legal disclaimer. It defines the conditions under which claims of ownership, consent, confidentiality, deletion, and portability remain meaningful in practice.

The core claim is that sovereignty over data depends on more than access controls. It depends on whether collection, inference, retention, sharing, and reuse are bounded by visible purposes, governed by revocable permissions, and constrained by structures that communities can understand and audit.

If DP4 is weak, predictable failures follow: consent theater, surveillance-by-default, inference without accountability, lock-in through broken portability, deletion promises that stop at the first vendor boundary, and community rules that cannot survive contact with underlying data pipelines.

DP4 therefore functions as a precondition for multiple later properties. Agency cannot be exercised over invisible data flows. Governance cannot constrain systems that communities cannot inspect. Ethical AI cannot be meaningful where the data it sees, stores, or trains on is structurally uncontrolled.

DP4 does not resolve all legal, jurisdictional, or sector-specific privacy questions. It defines the minimum conditions under which sovereignty and privacy remain real at the interface where data is created, combined, interpreted, and acted upon.

2. Problem Statement

In today's web, privacy is often presented as disclosure without control.

Participants are shown banners, terms updates, and granular-looking toggles, yet the underlying system still optimizes for maximal collection, indefinite retention, behavioral inference, and partner expansion. In many

cases, the formal interface of consent exists while the operational reality of choice does not.

This produces recurring failures:

- participants cannot tell what is being collected, for what purpose, for how long, or by whom
- data collected for one function expands into advertising, analytics, resale, or model training
- sensitive inferences are derived from behavior without clear notice, recourse, or deletion pathways
- data exports preserve liability but not usability, making exit technically legal but practically costly
- account deletion rarely maps cleanly to embeddings, downstream processors, backups, or partner copies
- communities cannot enforce stricter data norms inside their spaces because underlying systems remain vendor-shaped

These failures are not edge cases. They are structural consequences of architectures designed to treat data accumulation as default value creation.

DP4 addresses this by defining data sovereignty as an operational condition. Privacy becomes meaningful only when participants and communities can see the active terms of data use, limit those terms in practice, revoke permissions without fiction, and move or leave without losing the structure of their digital lives.

3. Threats and Failure Modes

3.1 Consent theater

Interfaces bundle unrelated processing into a single act of acceptance.

Example: A participant accepts a terms update to continue using a service and, in doing so, silently authorizes secondary uses of behavioral data for recommendation tuning, advertising, and model training.

Why this matters: The system records consent, but the participant did not experience a meaningful choice. DP4 treats this as a sovereignty failure, not a paperwork issue.

Consent theater occurs when a system records agreement without providing participants with a meaningful opportunity to understand or control the uses of their information. A participant may authorize one function while unknowingly permitting additional uses that are unrelated to the original purpose.

Example: A community application requires participants to accept updated terms before continuing to use the service. The agreement permits ordinary participation but also authorizes behavioral data to be used for advertising, recommendation systems, and AI model training. Although the participant accepts the terms to retain access, the system treats that acceptance as authorization for several distinct processing activities.

Why this matters: Formal agreement does not establish meaningful consent when unrelated purposes are combined into a single decision. Participants may be unable to distinguish the information required to provide a service from information collected for secondary uses. The resulting consent record creates an appearance of authorization without demonstrating that the participant had a meaningful choice.

DP4 addresses this failure through purpose binding and the consent stack. Distinct uses of participant information must remain separately identifiable and governable. Participants should be able to approve, decline, or revoke permissions for secondary uses without unnecessarily losing access to the underlying service.

Without these distinctions, consent becomes a condition of participation rather than an expression of participant control. Privacy remains formally disclosed but operationally constrained by the system's terms.

3.2 Purpose creep and secondary use

Data collected for one function expands into new products, ranking systems, partner programs, or model behaviors without a fresh social contract.

Example: Location data collected for safety or delivery is later used for engagement scoring, ad targeting, or brokered partner analytics.

Why this matters: The participant's mental model of risk becomes false. Trust erodes even where no obvious breach has occurred.

Purpose creep occurs when information collected for a specific function is later used for additional purposes without giving participants a meaningful opportunity to understand or authorize those changes. An initial permission may be treated as continuing authorization even when the scope of processing expands beyond the purpose originally disclosed.

Example: A community delivery application collects a participant's location to coordinate deliveries and support safe service operation. The same information is later used to measure engagement, target advertisements, or provide behavioral insights to third-party processors. Although the participant authorized location access for delivery, that permission did not necessarily extend to these secondary uses.

Why this matters: Participants cannot meaningfully govern their information when the purposes governing its use change without their knowledge. A permission granted under one set of expectations may become the basis for activities the participant did not anticipate or approve. This creates a gap between the participant's understanding of how their information is used and the system's actual processing practices.

DP4 addresses this failure through purpose binding. Collection and processing activities must remain connected to their declared purposes, and material changes must require visible reauthorization, reclassification, or zone-level review before the new use begins. Participants should be able to understand what has changed and exercise meaningful control over permissions for additional uses.

Without these safeguards, permission for a limited function can gradually become authorization for unrelated processing. Data sovereignty depends on ensuring that the conditions governing information remain visible, bounded, and enforceable as its uses evolve.

3.3 Illusory portability

Export exists formally but fails functionally.

Example: A participant downloads an archive that contains files and timestamps but omits social graph edges, permission history, role context, provenance, or schemas needed to restore meaningful continuity elsewhere.

Why this matters: Exit is made to look possible while dependency is preserved. DP4 requires portability that preserves usable structure, not only raw payloads.

Illusory portability occurs when participants can formally export their information but cannot meaningfully use it to continue their activities elsewhere. A system may provide access to stored records while omitting the relationships, permissions, roles, and contextual structures that give those records practical value.

Example: A participant leaves a community platform and downloads an archive containing files, messages, and timestamps. The export excludes relationship edges, role metadata, permission history, provenance markers, and the schemas needed to reconstruct interactions in another system. Although the participant receives a copy of their information, the structure required to restore meaningful participation is lost.

Why this matters: Data portability does not provide meaningful control when leaving a platform requires participants to abandon the relationships, context, and continuity established through their participation. An export that preserves content but not its usable structure may satisfy a formal requirement while maintaining practical dependence on the original system.

DP4 addresses this failure through meaningful portability. Exports must preserve the information and supporting structures necessary for practical migration, including relevant relationships, permissions, provenance, and contextual metadata. Where information or functionality cannot be transferred, systems must disclose those limitations and identify any resulting degradation.

Without these conditions, the ability to leave remains largely procedural. Data sovereignty requires that participants and communities can move between systems without unnecessarily losing the structure and continuity that make their information useful.

3.4 Inference without accountability

Systems derive high-stakes conclusions from behavioral traces without clearly governing how those inferences are created, used, challenged, or removed.

Example: A wellness application infers stress or depression risk from typing cadence and browsing patterns, then shares a derived score with an advertising or insurance intermediary.

Why this matters: The participant never explicitly submitted the sensitive category, yet is still acted upon as if they had.

Inference without accountability occurs when systems derive information about participants from their behavior without providing meaningful visibility into how those conclusions are produced, used, or shared. Such information may reveal sensitive characteristics or influence decisions even when participants never directly disclosed the underlying attributes.

Example: A wellness application analyzes typing cadence, search activity, and other behavioral patterns to estimate a participant's stress level. The resulting inference is shared with a third-party processor for purposes beyond the application's original function. Although the participant did not explicitly provide information about their emotional state, the system creates and distributes a derived assessment that may influence how the participant is classified or treated.

Why this matters: Information generated through inference can be as consequential as information directly supplied by a participant. When these conclusions remain invisible or cannot be challenged, participants may be subject to decisions based on interpretations they did not authorize, cannot verify, or may have reason to dispute. Inaccurate or misleading inferences can persist across systems, creating consequences that extend beyond the original interaction.

DP4 addresses this failure through sensitive inference governance, purpose binding, and auditability of use. Significant inferences must remain connected to their sources, declared purposes, and applicable permissions. Participants must have meaningful opportunities to understand consequential inferences, challenge inaccurate conclusions, and limit their use or downstream disclosure where applicable.

Without these protections, systems can create new forms of personal information that operate outside the participant's practical control. Data sovereignty requires that derived information remain governable rather

than becoming exempt from accountability because it was generated through automated analysis.

3.5 Retention without sunset

Data persists because retention is cheap, deletion is operationally inconvenient, and analytics cultures prefer indefinite memory.

Example: A participant deletes an account, but vector embeddings, partner datasets, abuse-model features, and backup systems continue to retain traces with no coherent deletion pathway.

Why this matters: Sovereignty requires time bounds. Without them, institutions remember indefinitely while participants bear the burden of asymmetrical memory.

Retention without sunset occurs when personal information remains stored or accessible after the purpose for which it was collected has ended. Account deletion or withdrawal of permission may remove information from a participant-facing service while leaving copies, derived artifacts, and associated records in downstream systems without a defined retention limit or verifiable deletion pathway.

Example: A participant deletes an account from an online service with the expectation that the associated information will be removed. Although the account is no longer accessible, copies of its contents remain in backup systems, partner databases, and analytics pipelines. Derived artifacts, including behavioral profiles and embeddings, may also persist without clear expiration conditions or a mechanism for confirming their removal.

Why this matters: Data sovereignty requires meaningful limits on institutional retention. When information persists indefinitely, participants may remain exposed to future uses, disclosures, or inferences long after their original participation has ended. The ability to delete an account does not provide meaningful control when the underlying information and its derivatives continue to exist beyond the participant's visibility or influence.

DP4 addresses this failure through retention clocks and propagation discipline. Information must remain subject to visible, purpose-bound or time-bound retention conditions. Deletion, correction, and withdrawal requests must propagate to known downstream processors, with auditable records

indicating which actions have been completed, which remain pending, and where technical limitations prevent full removal.

These protections do not require guarantees of perfect erasure where information has already been copied, transformed, or incorporated into systems that cannot fully reverse its use. They require transparent accounting of retention limits, propagation outcomes, and unresolved obligations.

Without these conditions, deletion becomes a change in account status rather than a meaningful reduction in retained information. Data sovereignty depends on ensuring that information does not remain indefinitely available simply because a system has the technical capacity to preserve it.

3.6 Cross-context correlation

Identifiers, device graphs, and fingerprinting techniques merge activity across settings that participants experienced as distinct.

Example: Pseudonymous participation in a civic forum is quietly linked to shopping behavior, social browsing, or location history through shared infrastructure.

Why this matters: Plural identity becomes decorative. Communities cannot sustain contextual integrity if correlation silently defeats boundaries.

Cross-context correlation occurs when information from separate areas of a participant's digital life is linked without their knowledge or meaningful authorization. Shared identifiers, device graphs, fingerprinting, and integrated processing systems may allow activities intended to remain distinct to be associated with the same participant.

Example: A participant contributes to a community forum using a pseudonymous identity that does not publicly reveal their personal information. The infrastructure supporting the forum also processes shopping activity, browsing behavior, or location history. Through shared identifiers or behavioral signals, these activities become associated with the participant's forum contributions, allowing information from otherwise separate contexts to be combined into a broader profile.

Why this matters: Contextual separation allows participants to engage in different communities without exposing unrelated aspects of their lives. When systems silently connect these activities, pseudonymity may no longer provide the expected protection, and information disclosed within one setting may influence how participants are interpreted or treated in another.

DP4 addresses this failure through persona boundaries, purpose binding, and consent propagation. Access permissions must remain scoped to the relevant persona or context, and information derived within one context must retain those boundaries as it moves between systems. Linking separate personas must require an explicit participant decision rather than occur as an unintended consequence of shared infrastructure or integrated services.

These protections do not require participants to maintain completely disconnected digital identities. They require that connections between contexts remain visible, deliberate, and subject to meaningful participant control.

Without these conditions, separate identities may exist only at the interface while the underlying systems continue to assemble a unified profile. Data sovereignty depends on preserving contextual boundaries not only in what participants choose to disclose, but also in what systems are permitted to connect, infer, and reuse.

3.7 False anonymity and weak de-identification

Organizations describe datasets as anonymized even where re-identification remains plausible or contractually enabled downstream.

Example: A mobility dataset stripped of names still exposes sparse routines in a small town, allowing individuals to be reconstructed through outside knowledge.

Why this matters: DP4 requires honesty about residual risk. "De-identified" cannot be treated as a magic word that dissolves responsibility.

False anonymity occurs when information is presented as anonymous or de-identified despite retaining characteristics that may allow individuals to be identified. Removing names and other direct identifiers does not necess-

arily prevent re-identification when behavioral patterns, contextual details, or external datasets can be used to reconstruct identity.

Example: A transportation dataset from a small town removes participant names before being shared for analysis. Although direct identifiers are absent, recurring travel routes, departure times, and frequently visited locations may reveal recognizable patterns. When combined with publicly available information or local knowledge, these patterns may allow individual participants to be identified.

Why this matters: Participants and communities may make decisions about information sharing based on the assumption that removing direct identifiers eliminates the possibility of identification. When residual risks are not disclosed, information may be shared, analyzed, or reused under privacy expectations that the underlying data cannot reliably support.

DP4 addresses this failure through data minimization, purpose binding, and transparent assessment of re-identification risk. Systems should evaluate whether remaining attributes, contextual patterns, or combinations with external information could reasonably expose participant identities before data is transferred or reused. The limits of de-identification should remain visible to participants and communities, particularly where datasets involve small populations or sensitive activities.

Zone-scoped privacy profiles can establish additional restrictions on the disclosure, processing, and reuse of information where re-identification could create elevated risks. These protections may include limiting the granularity of shared information, restricting access to sensitive datasets, or requiring privacy-preserving methods appropriate to the context.

Without these safeguards, de-identification may create an appearance of protection while leaving participants exposed to identification through indirect means. Data sovereignty requires honest accounting of residual risk rather than treating the removal of names as sufficient evidence of anonymity.

3.8 Partner sprawl without propagation

Deletion, revocation, and correction stop at the first layer of control.

Example: A participant deletes messages in one tool, but analytics vendors, cloud backups, and SDK partners continue to retain copies without visibility or participant recourse.

Why this matters: Sovereignty that fails at the first subcontractor boundary is not sovereignty.

Partner sprawl without propagation occurs when information is distributed across multiple systems or organizations without reliable mechanisms for communicating and applying subsequent changes to that information. A participant may delete, correct, or revoke access to information in the originating system while copies, derived artifacts, or associated permissions remain unchanged elsewhere.

Example: A participant deletes messages from a community application. The messages disappear from the participant-facing interface, but copies remain in cloud backups, analytics services, and partner systems that previously received the information. Because the originating application does not maintain effective downstream propagation records, these systems may continue to retain or process information that the participant believes has been removed.

Why this matters: Participant control becomes fragmented when actions taken in one system do not extend to other systems that received the same information. Deletion, correction, and revocation may appear successful while downstream copies remain available for continued use. Without visibility into where information has traveled and whether changes have been honored, participants cannot reliably understand the consequences of their decisions.

DP4 addresses this failure through propagation discipline, purpose binding, and auditable downstream obligations. Systems must maintain sufficient records of information sharing to communicate relevant deletion, correction, and permission changes to known downstream processors. Propagation outcomes must remain visible and accountable, distinguishing completed actions from pending requests, failures, and limitations that prevent full compliance.

These mechanisms do not guarantee that every copy can be located or removed, particularly where information has been transferred outside enf-

orceable relationships or incorporated into systems that cannot fully reverse prior processing. They require systems to preserve traceability, communicate applicable obligations, and disclose unresolved propagation failures rather than presenting local changes as universal completion.

Without these conditions, participants may exercise control over an interface while losing practical influence over the information distributed through its wider network. Data sovereignty requires that downstream processing remain accountable to the permissions, corrections, and retention limits governing the original information.

3.9 Youth and vulnerable-context overexposure

Defaults optimized for adult engagement expose minors and vulnerable users to data-intensive patterns they are less equipped to assess or contest.

Example: A youth-oriented social tool enables location sharing, behavioral profiling, or AI-mediated emotional inference by default.

Why this matters: DP4 requires higher baselines where stakes are higher. Uniform defaults can produce unequal harm.

3.10 Persona collapse

Separate contexts a participant deliberately kept apart are merged, either through shared infrastructure, an integration that resolves both to one account, or a vault whose contents are exposed as a single undifferentiated store.

Example: A participant maintains distinct personas for professional and health-support contexts. A newly added recommendation service reads across both because it was granted account-level rather than persona-level access.

Why this matters: Contextual integrity is the practical form most privacy takes. Collapse is usually irreversible: once two contexts have been linked, the inference cannot be withdrawn, and no subsequent revocation restores the separation.

Persona collapse occurs when information associated with distinct participant personas is combined or made accessible across contexts without meaningful authorization. Although participants may maintain separate

identities, roles, or participation settings, shared infrastructure and broadly scoped permissions can undermine the boundaries intended to keep those contexts distinct.

Example: A participant maintains one persona for professional networking and another for participation in a health support community. A recommendation service is introduced with broad access to information associated with both personas. The service begins combining professional activity with sensitive community interactions to generate recommendations or participant profiles, despite the participant having maintained these contexts separately.

Why this matters: Participants disclose different information in different contexts according to the relationships, expectations, and purposes associated with each setting. When those boundaries are bypassed, information shared within one context may become available for unrelated uses in another. This can expose sensitive associations, undermine trust, and create consequences that cannot be fully reversed once information has been combined or disclosed.

DP4 addresses this failure through persona-scoped permissions, purpose binding, and explicit authorization for cross-context correlation. Access granted within one persona must not automatically extend to another, including when services operate through shared infrastructure or integrated applications. Where participants choose to connect personas, the scope and consequences of that connection must remain understandable and subject to meaningful control.

These protections do not require every persona to function as a completely isolated technical identity. They require systems to preserve the contextual boundaries participants establish and prevent those boundaries from being silently overridden by broad access privileges, service integrations, or downstream processing.

Without these conditions, separate personas may exist only as interface distinctions while the underlying system treats their information as a unified profile. Data sovereignty requires that participants retain meaningful control over when, why, and how information from different parts of their digital lives may be connected.

Failed data sovereignty can also encourage retreat into silos, fabricated accounts, or manufactured activity while extraction continues. That response can damage the social signals on which others rely; privacy protection should not depend on fabricating a digital presence. Pseudonymity itself is not evidence of deception.

4. Core Principle

Data must retain meaning, consent, and accountability as it moves across systems. If data loses its binding to purpose, provenance, or permissions under transformation, sovereignty collapses into simulation.

Data sovereignty and privacy in the Meta-Layer require that personal and community data be collected, inferred, stored, shared, and reused only under visible, bounded, and governable conditions.

Those conditions must include:

- clear purpose binding
- minimization of collection, access, and retention
- meaningful consent and withdrawal
- portability with practical utility
- deletion or attenuation pathways that propagate as far as technically possible
- auditability of significant access, inference, and transfer events
- community capacity to impose stricter norms within governed zones

These conditions are not independent features. Together, they form the operational foundation of data sovereignty. Each mechanism protects a distinct aspect of participant and community control, while depending on complementary protections to remain effective. Consent without meaningful portability can still leave participants locked into a platform. Deletion without downstream propagation can leave copies and derived information beyond the reach of the original request. Transparency without auditability and accountability can reveal stated practices without providing meaningful evidence that those practices were followed.

The Meta-Layer therefore treats privacy as a coordinated system of protections rather than a collection of isolated controls. Data sovereignty depends

not only on the presence of individual mechanisms, but also on their ability to operate together across information collection, use, transfer, transformation, and eventual deletion or retention.

In today's web, these conditions rarely hold together. A system may disclose collection without limiting reuse, provide deletion without propagation, or offer export without restoration value. DP4 treats such partial compliance as insufficient.

The Meta-Layer reframes privacy as operational control at the point of interaction.

Example: A participant opens a data lens and sees active purposes, relevant processors, current retention clocks, sensitive inferences attached to their account, and downstream systems that have accessed their data. They can revoke training permission, export their activity in an interoperable format, contest a high-risk inference, and receive a propagation receipt for deletion requests.

What this feels like: Privacy stops being a maze of legal text and becomes a set of understandable levers tied to real system behavior.

Without this: Privacy becomes trust in opacity, and opacity fails precisely where accountability matters most.

Operational control means that participants and communities have meaningful, ongoing authority over how their information is collected, accessed, processed, shared, retained, and removed. This requires more than confidentiality or formal consent. It requires understandable information about data practices, actionable permissions, and mechanisms for verifying whether participant decisions have been honored.

Example: A participant accesses a community platform and can inspect the purposes for which their information is collected, the permissions currently in effect, the categories of recipients that have accessed it, and the retention conditions governing its use. The participant can also review consequential inferences associated with their information and identify available mechanisms for challenging them. If the participant withdraws permission for AI training, the system records and applies that change to future processing within its control. If the participant requests deletion, the system

provides visibility into downstream notifications, completed actions, pending obligations, and any limitations that prevent full removal.

Why this matters: Privacy cannot depend solely on information remaining hidden or on participants trusting that organizations will follow published policies. Meaningful data sovereignty requires participants to understand the practical consequences of information use and to exercise control over those consequences as information moves between systems. Permissions that cannot be inspected, changed, or meaningfully enforced provide only limited operational authority.

DP4 supports operational control through purpose binding, consent management, retention clocks, meaningful portability, sensitive inference governance, and auditable propagation of participant decisions. These mechanisms allow permissions and obligations to remain associated with information throughout its lifecycle, rather than existing only at the point of collection.

Participant-facing controls may bring these capabilities together through a coherent interface that displays active permissions, declared purposes, relevant access and processing history, retention conditions, and available actions. Such an interface should distinguish requested actions from confirmed outcomes and disclose limitations where complete enforcement or verification is not technically possible.

Without operational control, privacy protections may exist in policy or system architecture without becoming meaningful capabilities for participants. Data sovereignty requires that participants and communities can understand the rules governing their information, act within those rules, and assess whether the system has fulfilled its obligations.

5. Primary Mechanisms and Structural Conditions

5.1 Purpose binding

Every collection and processing pathway must declare its purpose in terms legible to both participants and communities. Material changes in purpose require visible reauthorization, reclassification, or zone-level review.

Example: A discussion zone permits summarization for moderation assistance but prohibits model training on participant content unless a separate, revocable grant is given.

Without purpose binding, consent collapses into blanket exposure.

Purpose binding requires information to remain associated with the specific, declared purposes for which its collection, processing, and use have been authorized. Permission granted for one purpose must not be treated as general authorization for unrelated or materially different uses. Purpose restrictions must remain applicable as information moves between systems, processors, and participating organizations.

Example: A community discussion platform uses AI-assisted tools to help moderators identify potentially harmful content. Participants understand that their contributions may be processed for this moderation purpose. The platform subsequently proposes using those same conversations to train a general-purpose AI model. Because model training represents a materially different purpose, the existing moderation authorization does not automatically extend to the proposed use. Participants and the community are informed of the new purpose, and the applicable authorization must be established before the additional processing begins.

Why this matters: Participants and communities make decisions about information sharing within particular contexts and expectations. When systems expand those purposes without meaningful authorization, information supplied for a limited function may become available for uses that participants neither anticipated nor accepted. This undermines the relationship between informed participation, permission, and accountability.

DP4 addresses this failure by binding information and derived artifacts to declared purposes and applicable permissions. Systems must distinguish authorized uses from proposed new uses and provide mechanisms for reviewing and approving material changes. Where a new purpose requires additional participant or community authorization, processing for that purpose must remain restricted until the relevant conditions have been satisfied.

Purpose binding must also extend to downstream processors. Information transferred to another system does not become unrestricted simply bec-

ause it has changed location, format, or processing environment. Recipients remain subject to the applicable purpose limitations, and any proposed expansion of use must follow the relevant authorization process.

Without these conditions, consent becomes an open-ended permission rather than a meaningful expression of participant and community intent. Data sovereignty requires that the purposes governing information remain explicit, enforceable, and subject to renewed authorization when materially changed.

5.2 Data minimization by design

Systems must begin from the least collection, retention, and sharing compatible with the function being offered, and expand only through visible, justified choices.

Example: A messaging tool does not upload contacts by default. Contact sync is presented as a distinct choice with plain-language scope and a reversible off switch.

This is not anti-functionality. It is a refusal to make maximal collection the silent baseline.

Data minimization by design requires systems to limit the collection, processing, retention, and disclosure of information to what is necessary for specific, declared purposes. The technical ability to collect information does not establish a legitimate need to do so. Systems should operate with the least information and access privileges required for their intended functions, rather than treating broad collection as the default.

Example: A community messaging application allows participants to communicate without requiring access to their device contacts. Contact discovery is available as an optional feature. When a participant chooses to use it, the application explains what information is needed, how it will be processed, and whether it will be retained or shared. The participant can decline the feature without losing access to core messaging functions and can withdraw the associated permission when contact discovery is no longer desired.

Why this matters: Unnecessary collection creates additional opportunities for exposure, misuse, unauthorized inference, and downstream disclosure.

Once information has entered a system, it may be copied, transformed, or distributed in ways that are difficult to reverse. Reducing collection at the outset limits these risks and reduces the burden placed on participants to manage information that was not necessary to provide the service.

DP4 addresses this failure by making minimization a design constraint rather than an optional privacy setting. Systems should distinguish information required for a declared function from information requested for additional features or purposes. Optional collection must remain subject to appropriate authorization, purpose limitations, and retention conditions. Where feasible, systems should prefer processing methods that avoid unnecessary disclosure or persistent storage.

Minimization must continue throughout the information lifecycle. Data that was necessary for an initial function does not automatically remain necessary indefinitely, and information collected for one feature must not become generally available to unrelated functions. Retention, access, and downstream sharing should remain limited to what the authorized purpose requires.

Without these conditions, participants may be asked to surrender more information than a service needs simply to participate. Data sovereignty requires systems to reduce unnecessary information collection before relying on consent, access controls, or deletion mechanisms to manage the resulting risks.

5.3 Consent stack

Permission must be layered, granular, and revocable, with separate scopes for distinct categories of data use.

This draft uses **consent stack** as a mechanism-level abstraction: a structured set of permissions that distinguish service provision, analytics, sharing, automation, and training from one another.

Example: A participant permits AI-assisted summarization of their workspace but declines training use and third-party analytics. Revoking training permission does not disable the summarization feature they actually wanted.

The consent stack makes partial participation possible without forcing blanket surrender.

The consent stack provides a structured mechanism for expressing, managing, and enforcing permissions across distinct uses of participant information. Authorization for one purpose must not be treated as blanket permission for unrelated collection, processing, sharing, or reuse. Permissions must remain sufficiently granular to reflect meaningful differences between purposes, features, recipients, and participation contexts.

Example: A participant uses a community platform that provides AI-assisted discussion summaries. The participant authorizes the processing necessary to generate summaries but does not authorize the use of their conversations for general-purpose AI model training or disclosure to external analytics providers. The platform maintains these permissions separately, allowing the participant to use the summarization feature without automatically enabling unrelated processing.

Why this matters: Participants may value particular services while objecting to secondary uses of the information those services require. When permissions are bundled into a single acceptance decision, participants may be pressured to authorize processing beyond what is necessary for their intended participation. This weakens the relationship between informed consent, purpose limitations, and meaningful control.

DP4 addresses this failure through a consent stack that distinguishes applicable participant permissions, community-level governance conditions, and system-level constraints. Permissions must remain connected to their declared purposes and relevant contexts, with mechanisms for inspecting, granting, declining, and withdrawing authorization where those choices are applicable. A permission granted at one level must not silently override restrictions established at another.

Consent changes must also have operational consequences. When a participant withdraws authorization for a particular use, systems must prevent further processing under that permission within the applicable scope and communicate relevant changes to known downstream processors. Previously completed processing, retained copies, and derived artifacts rem-

ain subject to the applicable retention, revocation, and propagation requirements, including disclosure of technical limitations.

The consent stack does not require every function to operate independently of the information technically necessary to provide it. It requires systems to distinguish necessary processing from optional or unrelated uses and to avoid conditioning access to one function on authorization for another without a justified dependency.

Without these conditions, consent becomes a broad contractual acceptance rather than a meaningful expression of participant and community intent. Data sovereignty requires permissions that are understandable, purpose-specific, context-aware, and capable of being exercised throughout the information lifecycle.

5.4 Meaningful portability

Portability must preserve enough structure to support continuity, not just compliance.

This includes, where technically honest and appropriate:

- stable identifiers
- schemas
- role metadata
- interaction history
- provenance markers
- permissions history
- relationship edges needed for mainstream migration use cases

Example: A participant exports a discussion archive that can be imported into another tool with thread structure, moderation history, authorship context, and trust signals intact.

Without this, "take your data with you" becomes formal rights without real exit.

Meaningful portability requires systems to support the practical migration of participant and community information between compatible environments. Portability must preserve not only the underlying content but also the relationships, permissions, roles, provenance, and contextual struct-

ures necessary to continue activities elsewhere. Providing downloadable files without the information required to interpret and reuse them does not, by itself, establish meaningful portability.

Example: A community decides to migrate from one discussion platform to another. The originating platform provides an export containing discussion content, thread structure, authorship information, moderation history, role metadata, and relevant relationship and permission records. The receiving platform can interpret the supported structures and reconstruct the community's discussions with sufficient continuity for participants to resume their activities without rebuilding the entire environment.

Why this matters: Participants and communities invest time in establishing relationships, organizing information, developing shared practices, and building trust within digital systems. When those structures cannot be transferred, changing platforms may require abandoning significant portions of that accumulated value. Formal access to exported information offers limited practical freedom when migration results in the loss of essential context or functionality.

DP4 addresses this failure by requiring portability to preserve usable structure rather than raw payloads alone. Where technically honest and appropriate, exports should include stable identifiers, schemas, role metadata, interaction history, provenance markers, permissions history, and relationship edges necessary for practical migration. These elements allow receiving systems to interpret the information and reconstruct supported aspects of the originating environment.

Meaningful portability must also respect applicable permissions, privacy boundaries, and the rights of other participants. Exporting a participant's information does not automatically authorize the disclosure of another participant's private information or guarantee that trust relationships, moderation decisions, or access privileges will carry identical authority in a different system. Receiving environments may require validation, adaptation, or renewed authorization before particular structures can be restored.

Systems must disclose material omissions, unsupported structures, and expected degradation where full continuity cannot be achieved. Portability does not require perfect replication of every feature or relationship, but it

does require an honest and usable pathway for migration rather than an export that preserves content while making practical continuation impossible.

Without these conditions, participants and communities may remain dependent on an originating platform despite possessing formal rights to retrieve their information. Data sovereignty requires that the ability to leave a system be supported by the practical ability to carry forward the information and context necessary for meaningful participation elsewhere.

5.5 Retention clocks and propagation discipline

Retention must be bounded by event-driven or purpose-bound clocks, not indefinite convenience. Deletion, correction, and revocation requests must propagate to known downstream systems with auditable outcomes.

This draft uses **retention clocks** as a mechanism-level abstraction: visible timers tied to categories of data and stated purposes.

Example: A participant can see that support logs expire in 30 days, abuse-review evidence in 180 days, and AI training exclusion tags apply immediately going forward. When deletion is requested, the system generates a receipt chain showing which processors complied, which are pending, and which limits remain technically unresolved.

DP4 does not require dishonest promises of perfect erasure. It requires propagation discipline and truthful accounting.

Retention clocks and propagation discipline establish time-bound limits on information use and accountable mechanisms for communicating deletion, correction, and permission changes across systems. Information must remain subject to defined retention conditions, while participant decisions affecting that information must be communicated to known downstream processors and tracked through auditable outcomes.

Example: A community platform provides participants with visibility into the retention conditions governing different categories of information. Support requests may have a retention period of 30 days, while moderation records may be retained for 180 days under a separately declared purpose. A participant who withdraws permission for AI model training can inspect the resulting change in authorization. When the participant requests delet-

ion, the platform records the request and reports which downstream processors have completed the applicable action, which remain pending, and where technical or other limitations prevent full removal.

Why this matters: Participants and communities cannot exercise meaningful control over information when retention periods are indefinite, undisclosed, or disconnected from the purposes for which information was collected. Similarly, deletion and correction requests provide limited protection when they affect only the originating system while copies or derived artifacts remain unchanged elsewhere. Without visibility into retention obligations and propagation outcomes, participants must rely on institutional assurances rather than verifiable evidence of follow-through.

DP4 addresses these failures through explicit retention clocks associated with information categories, declared purposes, and applicable governance conditions. Retention limits must remain understandable and enforceable, with information subject to deletion, expiration, or other authorized disposition when the relevant conditions are met. Continued retention must remain justified by an applicable purpose or obligation rather than by the technical ability to preserve information indefinitely.

Propagation discipline requires systems to maintain sufficient records of downstream information sharing to communicate applicable deletion, correction, and revocation requests to known recipients. These requests must produce auditable outcomes that distinguish completed actions from pending obligations, unsuccessful attempts, and disclosed limitations. A locally completed request must not be represented as universally fulfilled when downstream processing remains unresolved.

Retention and propagation mechanisms must also account for derived information, backups, and processing environments where immediate removal or reversal may not be technically possible. Withdrawal of permission can restrict future authorized use without necessarily reversing previously completed processing. Systems must disclose relevant limitations and preserve accountability for outstanding obligations rather than promising forms of erasure or reversal they cannot reliably perform.

Without these conditions, retention policies become statements of intent and participant requests become interface actions without dependable ope-

rational consequences. Data sovereignty requires that information have accountable retention boundaries and that changes to its permitted use remain traceable as it moves through participating systems.

5.6 Sensitive inference governance

Derived data can be more consequential than submitted data. High-risk inferences therefore require stronger conditions than ordinary processing.

This includes inferences relating to health, finances, minors, politics, biometric patterns, relational vulnerability, and similar domains of elevated risk.

Example: A system that predicts self-harm risk from behavioral cues must disclose that such inference exists, limit its downstream use, provide human escalation where appropriate, and prohibit repurposing for advertising or engagement optimization.

Inference must be governable as first-class data, not treated as exempt because it was machine-generated.

Sensitive inference governance establishes additional protections for information derived from participant data when that information may reveal sensitive characteristics, create significant risks, or influence consequential decisions. Derived information must not be treated as unrestricted simply because it was generated through analysis rather than directly supplied by a participant. The sensitivity and potential consequences of an inference may exceed those of the individual observations from which it was produced.

Example: A participant uses a community wellness application to record daily activities and routines. Through behavioral analysis, the system generates an assessment suggesting that the participant may be experiencing depression or financial stress. Although the participant did not directly disclose either condition, the resulting inference may reveal sensitive information or influence how the participant is classified, treated, or offered services. The system must therefore govern the inference according to its potential consequences, rather than assuming that permissions associated with the original activity records authorize every subsequent use.

Why this matters: Inferences can transform ordinary behavioral information into sensitive assessments that participants may not anticipate, understand, or consider accurate. When these conclusions remain invisible or are shared without appropriate restrictions, participants may experience consequences based on information they did not knowingly provide and cannot meaningfully challenge. The risks may be amplified when uncertain or incorrect inferences are treated as established facts.

DP4 addresses this failure through heightened governance of sensitive and consequential inferences. Systems must maintain appropriate visibility into significant inferred attributes, their declared purposes, and the permissions governing their use. Where applicable, participants must have meaningful opportunities to understand consequential inferences, challenge inaccurate conclusions, and seek correction or restriction of their use.

Inference governance must also preserve lineage and applicable permission boundaries. Derived information should remain associated with the sources, purposes, and authorization conditions that govern its creation and use. A sensitive inference must not become available for unrelated processing or downstream disclosure merely because it has been transformed into a new data artifact.

These protections must account for uncertainty and technical limitations. An inference is not necessarily a verified fact about a participant, and systems should not present probabilistic assessments as definitive conclusions. Where full explanation, correction, or removal is not technically possible, relevant limitations must be disclosed and appropriate restrictions maintained.

Without these conditions, systems can create consequential personal information outside the participant's practical awareness or control. Data sovereignty requires that sensitive inferences remain subject to meaningful governance throughout their creation, use, sharing, and retention.

5.7 Zone-scoped privacy profiles

Communities must be able to define stricter privacy norms within their zones while remaining interoperable with broader infrastructure.

This draft uses **privacy profile** as a mechanism-level abstraction: a machine-readable expression of the data rules that apply inside a zone.

A privacy profile may specify, for example:

- no third-party behavioral advertising
- no model training on participant contributions
- local-only processing for sensitive content
- elevated rules for youth participation
- stronger consent requirements for biometric or emotional inference

Example: A health-support community publishes a privacy profile that restricts cloud-based inference, blocks third-party SDKs, and requires explicit opt-in before any content can enter training pipelines.

Without zone-scoped privacy profiles, communities may have values but not operational control.

Zone-scoped privacy profiles allow communities to establish and enforce privacy requirements appropriate to the contexts they govern. These profiles define applicable constraints on information collection, processing, sharing, retention, and secondary use within a community zone. Participation in a larger digital ecosystem must not require communities to abandon privacy protections that are necessary for their members or purposes.

Example: An online health support community establishes a privacy profile prohibiting the use of member conversations for general-purpose AI model training, restricting third-party tracking, and requiring sensitive information to be processed only through approved services. These conditions apply to information processing within the community zone, including processing performed by integrated applications and authorized downstream services. A service that cannot satisfy the applicable restrictions must not be treated as automatically authorized to process the community's information.

Why this matters: Communities operate in different contexts and face different privacy risks. A health support community may require stronger protections for sensitive discussions, while a youth organization or neighborhood association may establish different restrictions appropriate to its

participants and activities. Uniform platform defaults may not adequately reflect these differences. Without community-level governance, participants may be exposed to information practices that conflict with the expectations and purposes of the spaces in which they participate.

DP4 addresses this need through zone-scoped privacy profiles that express community-level constraints in a form that participating systems can interpret and enforce. Profiles may establish restrictions on permitted processing purposes, external disclosures, tracking, retention, and the use of privacy-preserving methods. Applicable restrictions must remain effective across services operating within the zone and must not be silently weakened by platform defaults or third-party integrations.

Zone-level governance must operate alongside participant permissions and broader system constraints. Community authorization does not automatically override individual privacy protections, and individual permission does not automatically authorize processing prohibited by the community's applicable profile. Where requirements conflict, systems must identify the incompatibility and avoid processing that cannot satisfy the relevant conditions.

Privacy profiles should also support interoperability without assuming that every external service can satisfy every community requirement. Systems must make relevant limitations visible and restrict incompatible processing rather than presenting unsupported protections as enforceable guarantees.

Without these conditions, community privacy policies may remain statements of preference while the underlying infrastructure continues to apply uniform or weaker rules. Data sovereignty requires communities to exercise meaningful governance over information practices within the spaces they establish and maintain.

5.8 Auditability and provenance of use

Significant data access, transfer, and inference events must be inspectable in participant-legible and community-legible forms.

This does not require exposing every security detail publicly. It requires enough visibility to support contestation, trust, and oversight.

Example: A participant can see that an automated moderation agent accessed a document under a specific policy version, for a named purpose, with a recorded outcome and timestamp.

Privacy claims that cannot be audited remain aspirational.

Auditability and provenance of use require systems to maintain accountable records of significant information access, processing, transfer, and automated decisions. These records must provide sufficient context to establish what information was involved, which system or authorized actor performed the action, when it occurred, the declared purpose under which it was performed, and the resulting outcome where applicable.

Example: An AI-assisted moderation service reviews a participant's community post to assess whether it may violate an applicable community rule. The system records that the post was accessed for moderation, identifies the processing service and relevant purpose, records the time of review, and preserves the resulting moderation outcome. The participant and authorized community reviewers can inspect appropriate information about the action, allowing them to understand what occurred and determine whether the review followed the community's governance conditions.

Why this matters: Participants and communities cannot meaningfully assess compliance when consequential information use occurs without accessible evidence. Undocumented access, transfers, and automated decisions make it difficult to identify errors, investigate disputes, or determine whether processing remained within its authorized purpose. Transparency without reliable records may describe intended behavior without demonstrating what actually occurred.

DP4 addresses this failure through auditable records that connect significant processing events to their purposes, permissions, responsible systems, and relevant information lineage. These records support the examination of whether information was accessed or used within applicable constraints, including participant permissions, community privacy profiles, and downstream processing obligations.

Auditability must also respect privacy and security boundaries. Access to records should be appropriate to the reviewing participant or authorized governance role, and audit mechanisms should avoid unnecessarily exposi-

ng other participants' information, sensitive operational details, or additional personal data. Records should provide meaningful evidence without becoming a new source of unrestricted surveillance or indefinite retention.

Provenance of use must remain distinguishable from provenance of information. Knowing where information originated does not, by itself, establish how it was subsequently accessed, transformed, or disclosed. Systems must preserve sufficient evidence of significant uses and relevant transformations to support accountability across the information lifecycle.

Without these conditions, participants and communities may be asked to trust that systems have complied with their obligations without a meaningful way to examine that claim. Data sovereignty requires that consequential information use remain sufficiently traceable, understandable, and reviewable to support oversight, contestability, and correction where applicable.

5.9 Training and model-use boundaries

Where participant or community content could enter model training, fine-tuning, embedding pipelines, or retrieval systems, those pathways must be separately governed.

Example: A public discussion zone allows search indexing but defaults to no training use. Participants can grant corpus-level permission for research or model improvement on a renewable basis, and declined content carries an exclusion marker through the training pipeline.

This is a direct dependency between DP4 and later AI properties. Ethical AI claims are weak if model access to human data is structurally obscure.

Training and model-use boundaries require systems to distinguish information processing necessary for community participation and authorized services from processing intended to train, fine-tune, evaluate, or otherwise improve AI models. Permission to publish, search, summarize, or moderate information must not be treated as blanket authorization for its use in model development. AI-related processing must remain subject to declared purposes, applicable participant permissions, and community governance constraints.

Example: A participant contributes posts to a community discussion forum. Other authorized members can read and search those contributions

as part of the forum's normal operation. The platform does not automatically treat this availability as permission to use the posts for general-purpose AI model training. Where the community permits optional contributions to AI research or model improvement, the participant can authorize the applicable use separately. Declining that authorization does not, by itself, prevent the participant from continuing to use the forum's ordinary discussion features.

Why this matters: Community contributions are often created for communication, collaboration, mutual support, or shared knowledge development. Using those contributions to develop AI models may introduce materially different purposes, recipients, retention conditions, and downstream consequences. Participants and communities may reasonably authorize access to information within a particular setting while declining its incorporation into model-development processes.

DP4 addresses this distinction through purpose binding, granular authorization, and explicit governance of training and model-use permissions. Systems must distinguish the purposes for which information is available from the purposes for which it may be used in AI development. Training-related permissions must remain subject to applicable participant and community constraints, and those restrictions must be preserved when information is transferred to authorized processors.

Changes to training permissions must have operational consequences. Withdrawal of authorization must restrict subsequent processing within the applicable scope and trigger relevant downstream obligations. Systems must distinguish future training use from previously completed processing and disclose limitations where trained models or derived artifacts cannot be fully reversed, corrected, or removed.

These boundaries must also account for different forms of AI processing. Using a model to provide an authorized feature does not necessarily constitute permission to train or improve that model using the participant's information. Systems should make material distinctions between inference, temporary processing, evaluation, fine-tuning, and persistent incorporation into training datasets or model artifacts understandable wherever they affect participant or community authorization.

Without these conditions, information shared for community participation may become an unrestricted resource for AI development. Data sovereignty requires that training and model-improvement uses remain separately governable rather than being treated as automatic consequences of making information accessible within a digital system.

5.10 Jurisdictional and transfer honesty

Cross-border transfers and legal regime changes must be visible as part of the participant's risk surface.

Example: A participant is shown that a given processor operates under a different legal regime, that redress pathways are limited, and that a community zone therefore blocks that transfer category by default.

Global systems do not excuse vague disclosure. They heighten the need for explicitness.

Jurisdictional and transfer honesty requires systems to provide meaningful visibility into the jurisdictions in which participant information is stored, processed, accessed, or transferred. Where information crosses legal or organizational boundaries, systems must disclose material differences in applicable protections, access conditions, and accountability mechanisms rather than treating the transfer as an inconsequential technical operation.

Example: A community platform stores participant information within a locally governed environment but proposes using an external processing service operating in another country. Before transferring information, the platform identifies the relevant processing jurisdiction, explains material differences in applicable legal protections, and determines whether the proposed transfer complies with participant permissions and the community's privacy profile. Where the community has established restrictions on particular cross-border transfers, those restrictions remain applicable to the external service.

Why this matters: The jurisdiction in which information is processed can affect the legal obligations of service providers, the circumstances under which information may be accessed or disclosed, and the remedies available when disputes arise. Participants and communities may authorize information use within one legal or governance context without intending to

accept materially different conditions elsewhere. When transfers occur without adequate visibility, those expectations may be undermined without a corresponding opportunity for informed governance.

DP4 addresses this failure through transparent transfer conditions, purpose binding, and zone-scoped privacy requirements. Systems must identify material jurisdictional dependencies and preserve applicable restrictions when information is transferred to external processors. Proposed transfers must remain subject to the permissions, community rules, and legal obligations governing the information, rather than being justified solely by technical convenience or service integration.

Jurisdictional transparency must also distinguish physical data location from legal exposure. Information stored within one country may remain accessible to organizations or authorities operating under another jurisdiction, depending on the relevant legal and organizational relationships. Systems should therefore avoid presenting local storage or regional hosting as an unconditional guarantee of jurisdictional isolation.

Where a transfer cannot satisfy applicable restrictions, systems must disclose the incompatibility and prevent unauthorized processing. Where legal protections, enforcement conditions, or transfer consequences cannot be stated with certainty, systems must communicate those limitations rather than offering misleading assurances.

Without these conditions, participants and communities may lose meaningful oversight of the legal and institutional environments governing their information. Data sovereignty requires that cross-jurisdictional processing remain visible, appropriately authorized, and accountable to the protections under which information was originally entrusted to a system.

5.11 Data System Layer: Lineage, Transformation Integrity, and Consent Propagation

Beyond individual mechanisms, DP4 requires a coherent data system layer that preserves **lineage, semantics, and permissions** across pipelines, services, and time. This layer ensures that data remains trustworthy under transformation, scale, and adversarial use.

The data system layer requires information to retain sufficient provenance, contextual meaning, purpose limitations, and applicable permissions as it moves between systems or undergoes transformation. Copying, summarizing, aggregating, or deriving new information must not automatically remove the governance conditions associated with its sources. Information must remain understandable and accountable throughout its lifecycle, including when processed by automated systems.

Example: A participant shares information with a community service for a specific, authorized purpose. The service generates a summary, transfers selected information to an approved processing tool, and subsequently incorporates part of the resulting material into an automated workflow. Throughout these operations, the systems preserve sufficient records to identify the information's origin, the purposes governing its use, significant transformations, relevant recipients, and the permissions that continue to apply. The creation of a summary or derived artifact does not automatically authorize processing beyond the scope of the original permissions.

Why this matters: Participant and community control depends on the ability to understand not only what information exists, but also where it originated, how it has changed, and which restrictions govern its use. When information becomes detached from its context or permissions, downstream systems may treat it as unrestricted, even though its collection and processing were originally subject to specific conditions. This can undermine purpose binding, consent, auditability, and accountability across the wider ecosystem.

DP4 addresses this failure through information lineage, permission inheritance, and preservation of governance metadata across processing boundaries. Systems must maintain sufficient associations between source information, derived artifacts, declared purposes, applicable permissions, and significant transformations to support appropriate downstream enforcement and review.

These associations must remain meaningful when information is copied, summarized, combined, or transformed. Where derived information incorporates multiple sources, applicable restrictions must be evaluated together rather than discarded because the resulting artifact has a different

form. A transformation must not be used to bypass restrictions that would otherwise govern the information's use.

Preserving governance conditions does not require every processing system to retain an identical copy of all source records or metadata. Systems may use references, structured metadata, or other interoperable mechanisms, provided that the relevant provenance and obligations remain available and enforceable within the applicable processing context. Where lineage or permissions cannot be reliably preserved, the limitation must be disclosed and incompatible processing restricted.

Without these conditions, information may retain its content while losing the context and rules that make its use accountable. Data sovereignty requires that information remain connected to its origins, purposes, permissions, and significant transformations throughout the systems that process it.

5.11.1 Lineage continuity

- Data MUST carry provenance linking it to source, purpose, and processing context
- Derived artifacts (features, embeddings, summaries) MUST reference upstream lineage

Failure mode: **lineage loss**, enabling untraceable reuse and accountability gaps.

5.11.2 Transformation integrity

- Transformations MUST be attributable to an actor (human/AI) and a declared purpose
- Material transformations SHOULD be reversible or auditable where feasible

Failure mode: **data laundering**, where meaning or risk is altered without trace.

5.11.3 Consent propagation

- Permissions MUST travel with data across internal and external systems
- Downstream processors MUST honor upstream constraints or declare degradation explicitly

Failure mode: **consent bypass chains**, where integrations ignore or reinterpret permissions.

5.11.4 Anti-replay and non-duplication

- Identity- or consent-bound artifacts MUST NOT be reused to gain additional value without attribution
- Systems SHOULD detect duplicate extraction across pipelines

Failure mode: **replay extraction**, where the same data yields multiple unaccounted benefits.

5.11.5 Inference binding and governance

- Inferences MUST be treated as first-class data with lineage, purpose, and revocation pathways
- High-risk inferences require stricter constraints and auditability

Failure mode: **inference drift**, where derived signals are reused outside their declared context.

5.11.6 Cross-system semantics

- Systems MUST signal when data meaning or guarantees change across contexts
- Mappings between schemas MUST preserve or explicitly degrade semantics

Failure mode: **semantic drift**, where data is misinterpreted after transfer.

This layer does not require centralization. It requires **coherence under movement**.

5.12 Private community coordination

Communities should be able to coordinate through encrypted group communication and participant- or community-controlled hosting, reducing dependence on a single platform for their data and continuity. For high-stakes disclosure, including whistleblowing, tools should support deliberate control of identity exposure, storage, and disclosure timing.

Deletion or anonymization requests should be available within the systems' actual control, with explicit limits once information has been shared or

copied. Encrypted transport and decentralized storage are useful mechanisms, not guarantees that every adversary or downstream recipient loses access. DP15 addresses the threat model and DP18 closes the reporting feedback loop.

5.13 Reference implementation example: PDP-Connect (PDPP)

This subsection is non-normative. It names a concrete standards-track effort that demonstrates how DP4's consent, purpose, minimization, and durable-record mechanisms can be realized in practice. Adoption of PDPP for any given Meta-Layer system is an Overweb ADR decision, not a DP4 requirement.

PDPP (Personal Data Portability Protocol) is an open specification for moving personal data with the consent of the person it belongs to. It is developed by PDP-Connect, a lab at Linux Foundation Decentralized Trust. Source: pdpp.dev.

Mechanism mapping:

- OAuth 2.0 plus RFC 9396 (Rich Authorization Requests) in PDPP is one realization of the consent stack and scope binding described in DP4, Section 5.3.
- Source, Accessor, and Operator roles with declared purpose and expiration correspond to DP4's purpose binding under Section 5.1.
- Agreed terms visible to both parties satisfy the durable record requirement in DP4, Section 5.8.
- Range-limited selection requests (specific fields, specific date windows) correspond to DP4's minimization by design under Section 5.2.

Conformance posture:

PDPP exposes two verification rungs. A Conformant mark is produced by an automated check. A Verified mark requires human review by a technical committee, with identity anchored in a recognized trust registry. Source: pdpp.dev.

Cross-references: DP1, Section 9; DP7, Section 10; DP14, Section 9.

6. Data Vaults and Personas

The mechanisms above define the conditions data must satisfy. This section defines the two structures through which participants actually hold and partition it: the **vault**, which is where data rests under participant control, and the **persona**, which is the boundary across which data is permitted to be seen.

Together, participant-controlled vaults and personas make data sovereignty practical. Vaults provide participants with meaningful control over their information, while personas preserve the contextual boundaries between different parts of their lives. Just as people share different information with healthcare providers, employers, neighborhood associations, and support groups, digital systems should respect these distinctions rather than merging information from separate contexts into a single profile.

Both structures are necessary because they address different dimensions of data sovereignty. A vault without personas may provide a well-governed information store while still allowing information from distinct contexts to be combined into a unified, legible profile. Personas without a participant-controlled vault may provide contextual separation at the interface level while leaving the underlying information accessible to integrated processors capable of correlating those contexts.

Working together, vaults and personas enable participants to govern their information while controlling how it is disclosed and represented across different contexts. This combination establishes a practical foundation for data sovereignty by preserving participant control, supporting community governance, and maintaining contextual integrity as information moves throughout the Meta-Layer.

6.1 The personal data vault

A vault is a participant-controlled store in which data, credentials, and derived artifacts reside with their purpose bindings, retention clocks, and permission history attached. It is the operational answer to the question of where sovereignty physically lives.

A vault SHOULD provide:

- **Custody with meaningful control.** Vaults may be self-hosted, community-hosted, or provider-hosted. What DP4 requires is not a specific topology but that custody arrangements be disclosed and that provider-hosted vaults remain exportable and revocable, per the mechanisms in Sections 5.3 and 5.4.
- **Permission-bearing storage.** Data in a vault carries its consent state, not merely its content. A record retrieved from a vault arrives with the purposes for which it may be used, so that downstream processors receive constraints rather than payloads.
- **Retention clocks at rest.** Expiry is a property of the stored object. Data that has passed its clock is attenuated or removed by the vault itself rather than depending on each consuming system to remember.
- **Query-time minimization.** Where a request can be satisfied by an attestation rather than the underlying data (that a participant is over an age threshold, holds a credential, or belongs to a zone), the vault SHOULD answer with the attestation. Disclosure is the fallback, not the default.
- **Access logging as participant-facing history.** Every significant read, transfer, or inference against vault contents is recorded in terms the participant can review, supporting the mechanism in Section 5.8 at the point where access actually occurs.
- **Sensitive partitions.** Categories of elevated risk (health, financial, biometric, youth-context, relational) SHOULD be separable within the vault, so that a grant covering ordinary activity does not silently extend to them.

The characteristic failure mode is **vault theater**, where a store is described as participant-controlled while the operator retains parallel copies, unlogged access, or the practical ability to deny export. A vault that cannot be audited or exited is a storage location, not a sovereignty mechanism.

6.2 Personas as context boundaries

A persona is a bounded presentation of a participant within a context: what is visible, what is linkable, and what may be inferred. Personas are how cross-context correlation (Section 3.6) is resisted structurally rather than by policy promise.

DP4 requires that:

- **Personas be first-class in permission grants.** Access is granted to a persona, not to a participant. A processor authorized within one persona has no standing to read, correlate, or infer across others.
- **Correlation be an explicit act.** Linking personas is a participant decision with visible consequences, not a side effect of shared identifiers, device graphs, or an integration that resolves both to a common account.
- **Persona boundaries survive movement.** When data crosses a system boundary, its persona scope travels with it or the degradation is signaled, consistent with the requirement in Section 5.11.6.
- **Inference respect persona scope.** A derived signal produced within one persona inherits that persona's boundary. Inference is a common route by which separation is defeated, because the derived artifact is often treated as new data rather than as a descendant of its source.
- **Personas be enumerable and reviewable.** Participants must be able to see which personas exist, what each exposes, which processors hold grants against each, and what has been inferred within each.

The characteristic failure mode is **persona collapse**, described above under threats. It deserves particular design caution because it is effectively irreversible: revocation stops future access but cannot unlink an inference already drawn.

Personas should support anonymous, pseudonymous, or authenticated interactions where the relevant zone permits them. Authentication should not automatically disclose a public identity or link other personas. Connected-device identities also need contextual boundaries, since a device can otherwise become the bridge that silently reconnects separated activity.

6.3 Zone interaction

Personas and zone-scoped privacy profiles compose. A persona determines what a participant exposes; a zone profile determines what the environment permits to be collected, inferred, or transferred from whatever is exposed. A participant entering a health-support zone under a dedicated pers-

ona is protected twice: by the boundary they set and by the rules the community enforces.

Where the two conflict, the stricter constraint applies. A zone may impose limits beyond what a persona's grants allow, but a zone profile must not be usable to widen access beyond the persona's scope. Communities set floors, not ceilings, on participant protection.

6.4 Relationship to identity and naming

Vaults and personas depend on DP1 and DP5. DP1 supplies the accountability binding that makes plural presentation safe rather than evasive: personas are distinct without being unattributable, because action-bound accountability persists beneath them. DP5 supplies identifiers that can remain uncorrelated at the naming layer, without which persona separation is defeated before any data mechanism engages.

This is the practical shape of sovereignty in DP4: data held where the participant can see and move it, partitioned along boundaries the participant chose, carrying permissions that constrain whoever receives it, under rules the community can strengthen.

7. Governance, Accountability, and Agency Surfaces

DP4 is not satisfied by backend architecture alone. Participants and communities need interfaces through which data conditions become governable.

Participants must be able to:

- see active purposes, processors, retention clocks, and major inference categories
- revoke permissions without unfairly losing unrelated core functionality where technical separation is possible
- export their data in forms that preserve practical continuity
- correct or contest significant false inferences
- understand what deletion means in each category of storage and reuse

Communities must be able to:

- publish stricter privacy profiles for their zones
- reject tools that cannot meet those profiles
- audit aggregate compliance without turning privacy governance into a new surveillance regime
- preserve governance memory around why a privacy rule exists and when it changed
- distinguish individual confidentiality from community-level observability of system behavior

Example: A civic deliberation zone prohibits third-party trackers and emotional classification systems. Any overlay or agent entering the zone must declare compatibility with the zone's privacy profile or operate in a visibly constrained mode.

Without these surfaces, privacy remains vendor-defined even when communities appear to have rules.

8. Incentives and Power Analysis

Commercial systems tend to treat data surplus as strategic advantage.

Retention expands because future uses may be profitable. Inference expands because prediction creates leverage. Consent becomes cosmetic where friction threatens growth. SDK ecosystems and downstream processors thrive precisely when participants cannot trace the full chain of use.

DP4 does not assume these incentives disappear. It requires that their effects become visible and contestable.

Example: A platform discloses that feed ranking depends partly on behavioral surplus gathered across sessions. A community operating inside the Meta-Layer disables that ranking signal within its zone because it conflicts with the zone's purpose.

This matters because many privacy harms are not caused by one malicious actor. They emerge from ordinary growth logic operating without adequate brakes.

DP4 therefore treats incentive visibility as part of sovereignty. Participants should be able to know when they are not merely receiving a service, but

being rendered into a data asset.

9. Community Signals Informing DP4

Across contexts, similar signals recur:

- fatigue with unreadable policies and false choice architectures
- demand for portability that supports real migration
- frustration with invisible partner ecosystems and silent SDK extraction
- concern about sensitive inferences that participants never explicitly supplied
- desire to ask, in operational terms, "what does the system think it knows about me?"
- expectation that youth and vulnerable contexts should receive safer defaults, not merely more warnings

These signals are not abstract. They arise when people sense that the surface language of privacy no longer matches the structure underneath.

DP4 responds to that gap by making data conditions inspectable, debatable, and governable.

10. Foresight and Failure Design

DP4 assumes that data sovereignty degrades by accumulation rather than by breach. The characteristic failure is not an incident but a slow divergence between what a system's privacy surface describes and what its pipelines actually do, produced by ordinary engineering: a new processor added, a schema migrated, a model retrained, an integration shipped.

Predictable degradation paths include:

- **Purpose accretion**, where each individual expansion is defensible and the aggregate no longer resembles what was consented to
- **Lineage decay across migrations**, where provenance survives within a pipeline but is dropped at the boundary between one system's schema and the next
- **Inference laundering**, where a derived signal is treated as new primary data rather than as a descendant carrying its source's constraints,

resetting its permission state

- **Propagation rot**, where a deletion pathway that once reached every processor silently stops reaching a partner added after the pathway was built
- **Retention drift**, where clocks are defined but exceptions (legal hold, abuse review, backup, model memory) accumulate until the effective retention is indefinite
- **Profile erosion**, where a zone's privacy profile remains published while newly integrated tools are exempted for compatibility

These paths compound in a specific and dangerous way. Lineage decay makes inference laundering undetectable; undetected inference expands the effective purpose surface; an expanded purpose surface makes propagation incomplete by definition, because no one knows the full set of places the data went.

Two properties make DP4 failures harder to remediate than most:

Irreversibility. An inference drawn cannot be un-drawn, and a correlation established cannot be unestablished. Revocation stops future use but does not restore the prior state. This asymmetry means DP4 must weight prevention far more heavily than response.

Latency of harm. The consequences of over-collection may appear years later, under a different operator, a different legal regime, or a different threat model. Data retained today is evaluated against tomorrow's adversary, including capabilities that do not yet exist. Quantum-vulnerable encryption of long-retained data is the clearest current instance of this pattern.

DP4 therefore requires safeguards designed in advance:

- **Purpose diffing**, so that expansion of declared purpose is a visible change with a record, comparable to governance diffs under DP3
- **Lineage conformance testing** at every boundary crossing, treating provenance loss as a defect rather than an acceptable cost of integration
- **Propagation audits** that verify deletion reach against the current processor set, not the set that existed when the pathway was designed
- **Retention exception review**, with exceptions themselves carrying clocks

- **Inference registries**, so that the set of high-risk inferences a system produces is enumerable rather than emergent
- **Forward-secrecy posture** for long-retained data, treating future decryption capability as an expected condition rather than a remote scenario
- **Postmortems for privacy incidents** that link the harm to the specific mechanism gap (the unbound pipeline, the dropped lineage, the unreached processor) and to the change that introduced it

Failure is expected. Undetectable failure is not. A DP4-aligned system is one where the divergence between stated and actual data conditions is observable while consent can still be meaningfully revisited, and where the data most likely to cause future harm was never collected in the first place.

11. Relationship to Other Desirable Properties

DP4 is foundational and interdependent.

- **DP1** anchors accountability for who accessed, inferred, retained, or transferred what
- **DP2** ensures participants have meaningful agency over data flows and delegation scopes
- **DP5** supports identifiers and namespaces that allow identity and reputation to remain portable without forcing correlation
- **DP6** depends on commerce patterns that do not require surveillance as the default price of participation
- **DP7–DP10** shape whether sovereignty is actually usable through interface quality, comprehension, and incentives
- **DP11** depends on bounded visibility into what AI systems may see, remember, infer, and influence
- **DP12** provides the community processes through which stricter privacy norms can be defined and revised
- **DP13** enforces containment over automated access, training pathways, and runtime data exposure
- **DP14–DP15** reinforce provenance, auditability, and transparency of data use and policy change

- later governance and ownership properties depend on collective data practices that do not silently strip participants of control

A failure in DP4 propagates upward. If data conditions are opaque, later governance becomes symbolic, ethical AI becomes ungrounded, and participant agency becomes procedural rather than real.

12. Non-Goals and Explicit Boundaries

DP4 defines a minimum condition. It does not solve every problem associated with data, secrecy, or identity.

DP4 does not:

- guarantee perfect anonymity in all settings
- prohibit all inference or all data sharing categorically
- replace law, regulation, or sector-specific obligations in health, finance, education, or public safety
- mandate one universal privacy culture across all communities
- promise technically impossible forms of deletion while downstream copies or model memorization remain unresolved
- remove the need for accountability-linked identity in contexts where stronger attribution is legitimately required

These boundaries matter because absolutist privacy claims often collapse under real-world complexity.

For example, some communities may require stronger identity assurance to support trust and accountability. DP4 does not forbid that. It requires that data burdens attached to such assurance remain bounded, visible, and contestable.

Likewise, some inference may be necessary for accessibility, fraud prevention, or urgent safety intervention. DP4 does not deny that. It requires those pathways to be governed explicitly rather than smuggled in under vague necessity claims.

DP4 further does not:

- mandate a specific vault topology; self-hosted, community-hosted, and provider-hosted custody are all permissible where custody is disclosed and exit is real
- require that every participant maintain multiple personas, only that persona separation be available and enforceable for those who need it
- treat data localization as equivalent to sovereignty; jurisdiction is one factor in a risk surface, not a substitute for purpose binding and propagation discipline
- guarantee that privacy interfaces will be simple; it requires that they be legible, which sometimes means more surface rather than less

Failure mode: **privacy absolutism**, where DP4 is invoked to block accountability, safety intervention, or community oversight that the property explicitly contemplates as governable rather than prohibited.

13. Minimum DP4 Alignment

Minimum alignment is not a policy checklist. It is the threshold at which data sovereignty is **enforceable, portable, and resistant to laundering, drift, and silent reuse**.

A system that does not meet these conditions may disclose practices, but it does not provide sovereignty.

At minimum, a system claiming DP4 alignment MUST satisfy the following **irreducible conditions**:

13.1 Purpose binding and enforcement

- All collection and processing MUST declare purpose and enforce it in execution
- Material purpose changes MUST require reauthorization or zone review

Failure mode: **purpose creep**.

13.2 Consent propagation

- Permissions MUST travel with data across pipelines and partners
- Downstream systems MUST honor or explicitly degrade constraints

Failure mode: **consent bypass chains.**

13.3 Lineage and provenance

- Data and derivatives MUST carry reconstructable lineage
- Significant transformations MUST be attributable

Failure mode: **lineage loss / data laundering.**

13.4 Meaningful portability

- Exports MUST preserve structure needed for practical migration (schemas, relationships, permissions)
- Systems MUST disclose omissions and degradation

Failure mode: **illusory portability.**

13.5 Retention and propagation discipline

- Retention MUST be time- or purpose-bound with visible clocks
- Deletion/revocation MUST propagate with auditable receipts

Failure mode: **retention without sunset / partner sprawl.**

13.6 Inference governance

- High-risk inferences MUST be disclosed, bounded, and contestable
- Inferences MUST support correction or attenuation where applicable

Failure mode: **inference without accountability.**

13.7 Auditability of use

- Participants MUST be able to inspect significant access, transfer, and inference events
- Systems MUST provide logs or summaries sufficient for contestation

Failure mode: **opaque processing.**

13.8 Interoperability honesty

- Systems MUST state what is preserved, degraded, or non-transferable across boundaries

Failure mode: **interop deception.**

13.9 Context separation

- Permission grants MUST be scoped to a persona or context rather than to a participant in the aggregate
- Correlation across contexts MUST require an explicit participant act, not arise as a side effect of shared infrastructure
- Derived signals MUST inherit the context boundary of their source

Failure mode: **persona collapse.**

These conditions define the **minimum viable data sovereignty layer** of the Meta-Layer.

Partial implementations that omit purpose enforcement, consent propagation, lineage, or propagation discipline MUST NOT be considered aligned with DP4.

14. Open Questions and Future Work

DP4 surfaces unresolved design challenges that require further work:

- how to standardize interoperable privacy profiles across tools and zones
- how to represent retention, revocation, and transfer states in ways ordinary participants can understand
- how to govern model memorization and partial deletion in training-intensive systems
- how to balance privacy-enhancing technologies with the need for accountability and community oversight
- how to manage joint controllership and downstream processor responsibility in federated ecosystems
- how to handle collective consent for shared datasets or community archives
- how to distinguish emergency access from routine surveillance creep
- how to make privacy interfaces usable for low-literacy, multilingual, and neurodiverse participants
- how to audit third-party SDK ecosystems without reproducing surveillance in the name of governance

- how to preserve interoperability while allowing communities to adopt materially stricter norms

The vault and persona structures raise a further set of unresolved questions:

- how vaults should interoperate, so that a participant can move custody without re-establishing every grant, lineage record, and retention clock from scratch
- how to make persona separation usable rather than merely available, given that maintaining boundaries is cognitively expensive and mistakes are irreversible
- how to express long-horizon retention risk to participants, including risks from decryption capabilities that do not yet exist
- how to enumerate the inferences a system produces when many are emergent properties of models rather than declared features
- how to verify propagation claims independently, since deletion receipts are currently self-reported by the parties with the least incentive to report failure

These are not reasons to delay better defaults. They mark the frontier where DP4 must mature through practice, governance, and implementation evidence.

15. Path Toward ML-RFC

Progression from draft to RFC-grade maturity would require:

- stable invariants around purpose binding, minimization, meaningful portability, and propagation honesty
- a clearer grammar for privacy profiles, retention clocks, consent stacks, and audit events
- implementation evidence from zones with different stakes and governance norms
- demonstrable ways for communities to adopt stricter data rules without breaking interoperability
- better treatment of partial deletion, inference correction, and training exclusions

- alignment with privacy law and privacy-enhancing technologies without waiting for legal perfection before improving defaults

Graduation criteria SHOULD additionally include:

- reference schemas for vault records, persona scopes, consent stacks, retention clocks, and propagation receipts
- conformance tests for lineage preservation across at least one system boundary
- evidence that an export is usable in an independent system without loss of permission state
- evidence that persona separation resists correlation under realistic integration conditions

A decentralized privacy-routing mesh is one possible implementation approach for reducing the exposure of interaction, contribution, and identity-resolution traffic. Multiparty relays and onion routing are candidate mechanisms for limiting the visibility of communication patterns and reducing reliance on individual intermediaries. These approaches should remain optional and be evaluated against their actual privacy benefits, operator dependencies, accessibility, reliability, and performance before broader adoption. DP7 addresses compatibility with participant-chosen privacy infrastructure, while DP17 addresses potential models for supporting relay operations and funding.

Local, multi-sector civic sandboxes, including the proposed UMi Insecurity Initiatives and Sandbox environments, provide potential settings for evaluating whether privacy mechanisms are understandable, equitable, and effective under real-world conditions. Such trials should examine not only technical performance but also participant comprehension, accessibility, governance, operational dependencies, and unintended consequences. A pilot's institutional setting or participation by trusted organizations is evidence to examine, not proof that the mechanisms being tested provide meaningful privacy protections.

The goal is not to freeze one final model of privacy. It is to establish durable conditions under which sovereignty claims can be tested, challenged, and improved.

16. Closing Orientation

DP4 is where the Meta-Layer rejects the old bargain of convenience in exchange for invisibility.

Sovereignty is not achieved when a participant is merely informed that data extraction may occur. It is achieved when the participant and the communities they inhabit can see the operative terms of data use, shape those terms where appropriate, withdraw from them in meaningful ways, and leave without losing the structure of their digital life.

When DP4 is strong, trust in governance, AI, commerce, and collaboration becomes plausible.

When DP4 is weak, every higher-order property is forced to fight against a substrate that quietly converts participation into extraction.

Continue the conversation on DP4 online →

DESIRABLE PROPERTY 5

DP5

Decentralized Namespace

Claim your space in the Meta-Layer – meta-domains and personal identifiers give you sovereign, portable identity, owned by you, not a platform.



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp05

1. Purpose of This Draft

This draft articulates Desirable Property 5 (DP5) as the condition under which people, communities, agents, artifacts, and spaces can be named, addressed, discovered, traded, and governed across the Meta-Layer without dependency on a single platform or registry.

DP5 introduces meta-domains and personal identifiers as sovereign, portable identity and addressability primitives. These identifiers allow participants to claim space in the Meta-Layer, link that space to existing web domains or decentralized identifiers, and use names as anchors for identity, ownership, trust, commerce, and governance.

The core claim is simple:

> *Meta-domains and personal identifiers give participants sovereign, portable identity – owned by them, not rented from a platform.*

The Meta-Layer introduces a decentralized namespace system where identity is not merely a login and addressability is not merely a URL. Names become portable anchors for people, ideas, artifacts, communities, overlays, smart tags, and virtual spaces across the open web.

DP5 guides implementation, governance design, and future ML-RFC development for decentralized naming, meta-domain registration, namespace rights, conflict resolution, and interoperable naming semantics.

2. Problem Statement

Why namespaces matter.

The contemporary web depends on naming systems, but most participant-facing names are not truly participant-owned. Handles, usernames, pages, tags, groups, channels, and platform identities are typically rented from centralized services. They can be revoked, shadowed, duplicated, impersonated, renamed, captured, or made non-portable by the platforms that host them.

This creates recurring failures:

- participants build identity and reputation around names they do not control
- communities lose continuity when platforms change rules or shut down access
- artifacts cannot be reliably addressed across tools
- names become vulnerable to spoofing, squatting, seizure, and censorship
- cross-system identity and ownership degrade because identifiers do not preserve meaning across contexts

DP5 reframes naming as civic infrastructure. A decentralized namespace is not only a convenience layer. It is the addressability substrate for identity, agency, data, commerce, interoperability, and governance.

Without DP5, the Meta-Layer cannot reliably answer basic questions:

- What is this person, persona, community, artifact, tag, overlay, or space called?
- Who controls that name?
- What does the name resolve to?
- What rights, policies, and histories are bound to it?
- How does the name remain interpretable across systems?

3. Threats and Failure Modes

3.1 Platform-rented identity

Participants build identity around handles or pages that can be revoked, hidden, renamed, or monetized by platform operators.

Failure mode: identity continuity depends on platform permission.

3.2 Namespace capture

Dominant registries or intermediaries control which names are valid, visible, or resolvable.

Failure mode: decentralized naming becomes centralized gatekeeping.

3.3 Spoofing and impersonation

Attackers create visually, semantically, or structurally similar names to mislead participants.

Failure mode: names become attack surfaces for scams and trust abuse.

3.4 Squatting and speculative enclosure

Valuable names are claimed not for use, but to extract rents from future participants or communities.

Failure mode: addressability becomes enclosure before public value can form.

3.5 Semantic drift

A name carries one meaning in one system and a different meaning elsewhere without signaling.

Failure mode: identity, trust, or ownership claims are misinterpreted across contexts.

3.6 Registry fragmentation

Multiple naming systems emerge without interoperability or conflict-resolution pathways.

Failure mode: participants cannot know which namespace claims are authoritative, compatible, or contested.

3.7 Artifact ambiguity

Objects, tags, posts, paths, and digital artifacts cannot be reliably referenced across systems.

Failure mode: knowledge, provenance, and ownership degrade because identifiers are not stable.

3.8 Non-human namespace ambiguity

AI agents, organizations, bots, and autonomous systems operate without clear namespace rights or management structures.

Failure mode: non-human actors become hard to distinguish, govern, or hold accountable.

These eight patterns are elaborated in operational detail under *Namespace Attack Taxonomy*, which adds resolver capture, ownership laundering, registry amnesia, and namespace flooding as concrete attack surfaces with required responses. The distinction is one of altitude rather than kind: this section names the structural failures a namespace must be designed against; the taxonomy names how those failures are actually attempted.

4. Core Principle

Names in the Meta-Layer must be portable, resolvable, governable, and resistant to capture. A namespace that cannot preserve identity, meaning, and control across systems becomes another platform dependency.

DP5 treats names as more than labels. Names are anchors for participation, reference, ownership, navigation, reputation, and coordination.

A DP5-aligned namespace must therefore support:

- participant-owned identifiers
- community-owned identifiers
- artifact and object identifiers
- namespace portability across tools
- conflict resolution and reservation logic
- verifiable ownership and control
- interpretable resolution across systems
- resistance to squatting, spoofing, and seizure

5. Primary Mechanisms and Structural Conditions

DP5 is enacted through mechanism families that together turn a name from a label into an accountable, resolvable, governed object. Each family addresses a distinct question that a namespace must be able to answer, and a namespace that answers some but not others will exhibit the failure modes named above.

- **Namespace objects** define *what can be named*: meta-domains for spaces, personal identifiers for participants and personas, artifact identifiers for objects, name chains for structured composition, and well-known URI anchors for publisher-controlled trust data.
- **The namespace system layer** defines *what a name is*, across five coupled layers: syntax, resolution, control, meaning, and history. Most existing naming systems implement one or two of these. DP5 requires all five, because omitting any one produces a specific, predictable failure – ambiguity, invisible override, ghost control, semantic drift, or history erasure.
- **The resolution protocol** defines *how a name is interpreted*: a governed, inspectable process that normalizes, validates, queries multiple resolvers, returns explicit state, attaches provenance, applies local policy, and renders an interface signal.
- **The state machine** defines *what a name currently is*: available, claimed, provisional, verified, active, disputed, quarantined, transferred, retired, or forked, with receipts on transitions that affect ownership, resolution, or trust.
- **The attack taxonomy** defines *what a name must withstand*, pairing each known attack with a required structural response rather than a moderation promise.
- **Registry architecture and registration rules** define *how names come into being*: ingest, candidate classification, canonicality, snapshots with checksums, pluggable indexers, and deterministic validation of labels.
- **Tradeability with bounded trust** defines *how a name may change hands* without laundering its history, through transfer receipts, provenance preservation, and visible dispute state.

The structural condition uniting these is that **a name must never assert more trust than it can demonstrate**. Every mechanism in this draft exists to close a gap between what a name appears to mean and what can actually be verified about it. Where verification is unavailable, the mechanism's obligation is to signal uncertainty rather than to default to trust – which is why explicit state, resolver provenance, and visible degradation recur throughout.

6. Primary Namespace Objects

6.1 Meta-domains

Meta-domains address virtual spaces within the Metaweb, similar to how traditional domains address web spaces.

A meta-domain may refer to:

- a participant-controlled overlay space
- a community zone
- a smart-tag namespace
- an application surface
- a virtual or conceptual space
- a bridge between a traditional domain and Meta-Layer objects

Example forms include:

- `boeing.com.web4`
- `apple.com.web4`
- `example.com.meta`
- `<label>.example.com.meta`

Meta-domains can link seamlessly to the broader web while functioning within the Metaweb overlay framework.

6.2 Personal identifiers

Personal identifiers address participants and personas, similar to email addresses, handles, or DIDs, but portable across Meta-Layer contexts.

Example:

- `shiftshapr.web4`
- `@jaime`
- `@jaime/artifact99`

Personal identifiers may be connected to decentralized identifiers (DIDs), credentials, proof-of-humanity mechanisms, or zone-specific identity contexts.

6.3 Digital artifact identifiers

Digital artifacts may serve as identifiers, assets, or NFTs that can be bought, sold, transferred, authenticated, or referenced.

Artifacts may include:

- posts
- paths
- smart tags
- annotations
- media objects
- overlays
- lists
- navigation components
- assertions
- credentials

6.4 Name chains

Name chains provide structured, semantic identifiers such as:

- `@user/object`
- `@community/tag`
- `@publisher/claim`

Name chains function as URI-like trust anchors, combining identity verification, object authentication, and conflict resolution pathways.

6.5 Decentralized URIs and well-known paths

DP5 also recognizes publisher-controlled decentralized URI patterns, such as:

- `/.well-known/trust.txt`

These allow trusted data to be anchored under existing publisher-controlled domains without requiring centralized registries.

7. Namespace System Layer: Resolution, Ownership, and Trust Semantics

This section defines DP5 as a **runtime-resolvable, multi-layer namespace system** rather than a static registry.

A DP5-compliant namespace operates across coupled layers:

7.1 Naming Layer (Syntax)

Defines canonical forms, label rules, and human-readable structure.

Examples:

- `<label>.example.com.meta`
- `@user`
- `@user/object`

Requirements:

- deterministic parsing
- canonical normalization

Failure mode: syntactic ambiguity.

7.2 Resolution Layer (Mapping)

Maps names → entities (people, agents, artifacts, spaces).

Resolution MUST include:

- multi-source resolvers (not single authority)
- explicit resolver provenance
- deterministic fallback rules
- verifiable resolution outputs

Resolution states MUST be visible:

- verified
- provisional
- disputed
- forked
- quarantined

Failure mode: invisible resolution override.

7.3 Control Layer (Authority)

Defines who can act on a name.

Control may derive from:

- cryptographic keys (wallets)
- DIDs
- registry attestations
- community governance

Control MUST be:

- transferable
- revocable
- auditable

Failure mode: ghost control (no accountable owner).

7.4 Meaning Layer (Semantics)

Defines what the name *means* in context.

Includes:

- identity type (human, org, agent, artifact)
- trust signals
- governance policies
- reputation bindings

Meaning MUST:

- travel with the name
- degrade visibly across contexts

Failure mode: semantic drift.

7.5 Temporal Layer (History)

Names are not static. They evolve.

Systems MUST preserve:

- ownership history
- resolution changes
- dispute timelines
- transfers

Failure mode: history erasure.

These five layers together define a **full-stack namespace**. Most systems today only implement 1–2 layers. DP5 requires all five.

8. Name Resolution Protocol (Reference Model)

DP5 requires a shared resolution model so names can be interpreted consistently across tools, overlays, registries, and communities.

A reference resolution flow SHOULD include:

1. **Normalize** the name into canonical form.
2. **Validate** syntax, reserved status, and structural constraints.
3. **Query** one or more resolvers or registries.
4. **Return state**: verified, provisional, disputed, forked, quarantined, retired, or unresolved.
5. **Attach provenance**: resolver source, timestamp, signatures, registry snapshot, and confidence level.
6. **Apply local policy**: zone rules, trust profiles, community reservations, and agent constraints.
7. **Render interface signal**: show the participant what the name means, what is uncertain, and what actions are safe.

Resolution must therefore be understood as a governed process, not a hidden lookup.

A minimal response object SHOULD include:

```
{
  "name": "@jaime/artifact99",
  "canonical": "@jaime/artifact99",
  "entity_type": "artifact",
  "controller": "did:example:123",
  "state": "verified",
  "resolver": "registry.example.meta",
  "resolved_at": "2026-04-26T00:00:00Z",
  "provenance": {
    "snapshot": "graph-snapshot-v12.json",
    "checksum": "sha256:...",
    "signatures": ["..."]
  },
  "policy": {
    "transferable": true,
    "dispute_status": "none",
    "zone_constraints": []
  }
}
```

Failure mode: **opaque resolution**, where a name appears trustworthy but participants cannot inspect how that trust was produced.

9. Namespace State Machine

DP5-aligned systems SHOULD expose name lifecycle states explicitly.

A name may move through the following states:

```
available → claimed → provisional → verified → active
                      ↘ disputed → resolved
                      ↘ quarantined → restored / retired
active → transferred → active
active → expired / abandoned → available or archived
active → forked → parallel-governed state
```

Core states:

- **available**: no active claim exists
- **claimed**: a participant or community has initiated registration
- **provisional**: claim exists but is pending verification or policy review
- **verified**: claim has met required proof conditions
- **active**: name is resolvable and usable

- **disputed**: competing claims or safety concerns exist
- **quarantined**: name is constrained due to suspected abuse, spoofing, or policy conflict
- **transferred**: control has changed with receipt
- **retired**: name is no longer active but history is preserved
- **forked**: multiple governance contexts recognize different meanings or controllers

State transitions MUST produce receipts where they affect ownership, resolution, trust, or participant expectations.

Failure mode: **state invisibility**, where participants cannot tell whether a name is safe, contested, provisional, or compromised.

10. Namespace Attack Taxonomy

DP5 treats names as attack surfaces. Naming systems concentrate trust, discovery, ownership, and memory, making them attractive targets for adversarial actors.

Common attacks include:

10.1 Spoofing

Attackers register visually or semantically similar names to impersonate trusted entities.

Examples:

- `appIe.meta` using confusing characters
- `paypaI.web4`
- near-match community names

Required response: similarity detection, warnings, and dispute pathways.

10.2 Squatting

Actors claim names for speculative rent extraction rather than use.

Required response: reservation rules, renewal logic, staking, decay, or community challenge mechanisms.

10.3 Resolver capture

A resolver becomes a de facto authority by controlling defaults.

Required response: multi-source resolution, resolver provenance, and fallback transparency.

10.4 Ownership laundering

A name is transferred repeatedly to obscure harmful history, evade sanctions, or reset trust.

Required response: transfer receipts and visible lineage.

10.5 Semantic hijacking

A name is used in a new context to imply trust or meaning it did not originally carry.

Required response: semantic profiles and degradation signaling.

10.6 Agent impersonation

Automated or AI actors adopt names that imply human identity, authority, or community status.

Required response: mandatory entity classification and controller binding.

10.7 Registry amnesia

A registry loses or suppresses prior state, disputes, or ownership transitions.

Required response: append-only logs, snapshots, checksums, and independent archival.

10.8 Namespace flooding

Attackers create many names to overwhelm discovery, governance, or trust review.

Required response: rate limits, economic friction, proof thresholds, and anti-spam containment.

11. Reference Patterns and Compatibility

DP5 does not replace existing naming systems. It defines how Meta-Layer naming can interoperate with them.

11.1 DNS-style names

DNS provides global resolution and familiar domain semantics, but is vulnerable to centralized registrar control and does not natively express trust state, provenance, or community governance.

DP5 can use DNS-linked anchors while adding overlay-level trust semantics.

11.2 DID-style identifiers

DIDs support decentralized identity and controller binding, but may be difficult for ordinary participants to read or remember.

DP5 can bind human-readable names to DID-backed controllers.

11.3 ENS / SNS-style naming

Blockchain naming systems support ownership and transfer, but often emphasize asset ownership more than contextual governance, dispute visibility, or semantic profiles.

DP5 can learn from these systems while requiring visible state, provenance, and governance.

11.4 Ordinals / BRC333-style artifacts

Ordinal inscriptions and BRC333-shaped artifacts can anchor durable metadata and namespace records.

DP5 can use inscription ordering, registry snapshots, and graph facts to support canonicality and provenance.

11.5 Well-known URI anchors

Publisher-controlled paths such as `/.well-known/trust.txt` allow existing domain holders to publish trust-relevant metadata without centralized registry dependency.

DP5 can compose these with meta-domain records and overlay resolution.

Publisher-controlled declarations can also state relationships among publishers and associations. They can carry extension fields, for example AI-training permissions or social identifiers (trust.txt specification). A declaration is the publisher's own claim. Verification therefore separates what a publisher declares from independent evidence that a claimed relationship is valid. For example, a hypothetical news site may list membership in a press association, but the listing counts as verified only if the association confirms it. A declared permission also needs a clear scope. Publishing a field does not by itself establish that the permission is enforced or widely adopted.

The goal is not one namespace to rule them all. The goal is interoperable naming with explicit trust semantics.

12. Meta-Domain Registry Architecture

A Meta-Domain Registry may operate as a public ingest and registry service for meta-domains, SNS-shaped JSON, BRC333-related ordinals, and related naming artifacts.

A registry can include:

- block-tip polling
- indexer search through services such as UniSat or pluggable alternatives
- candidate classification
- structural graph facts
- quarantined trust edges
- versioned graph snapshot releases with checksums

Such a registry SHOULD distinguish between:

- structural facts that can be merged freely
- trust assertions that require quarantine or policy review
- local drafts that remain outside public canonical state

12.1 Candidate classes

Candidate logs may include:

- `.meta` tokens

- BRC333-shaped bodies
- SNS-alias JSON candidates
- tag registry records
- anchor records
- name-chain references

12.2 Registry outputs

A registry SHOULD provide:

- canonical records
- candidate logs
- graph snapshots
- checksums
- dispute or quarantine markers
- provenance for resolver decisions

12.3 Canonicality and ordering

Where ordinal inscriptions are used, canonicality SHOULD be determined by documented ordering rules, such as lowest qualifying inscription number where adopted, rather than arbitrary string ordering.

12.4 Pluggable indexers

Registries SHOULD support pluggable indexers and resolution sources so that namespace infrastructure does not depend on a single data provider.

Failure mode: **indexer dependency**, where resolver integrity depends on one commercial or centralized API.

13. Registration Rules and Validation

DP5 supports concrete registration rules for product-level namespaces. The ASCII-only label rules below are an example profile for one product namespace; they are not a DP5 requirement.

13.1 Canonical form example

A v1 product canonical form may be:

```
<label>.example.com.meta
```

Where:

- `<label>` is the registrant-chosen segment
- `example.com.meta` is a configurable suffix or canonical parent

13.2 Label rules

A valid label SHOULD satisfy:

- lowercase ASCII letters, digits, and hyphen only (`a-z`, `0-9`, `-`)
- no underscores or Unicode in v1 unless explicitly expanded later
- must not start or end with `-`
- length between 1 and 63 characters
- must not be an integer-only label
- must not be reserved

13.3 Structural validation regex

Label-only validation:

```
^(?!-)([a-z0-9](?:[a-z0-9-]{0,61}[a-z0-9])?)$
```

Full-domain validation example:

```
^(?!-)([a-z0-9](?:[a-z0-9-]{0,61}[a-z0-9])?)\.example\.com\.meta$
```

Regex alone is insufficient. Reserved-name and integer checks must be separate.

13.4 Recommended validation order

1. Parse `label` as the substring before the configured suffix.
2. Reject if label fails length or charset validation.
3. Reject if label matches `/^\d+$/`.
4. Reject if `label.toLowerCase()` is in the reserved set.
5. Reject if the normalized domain is already in use.
6. Accept and register.

13.5 In-use rule

"In use" means that a record already exists with the same normalized domain string under the applicable active status rule.

Storage and comparison SHOULD normalize to lowercase.

14. Interoperability and Tradeability

Tradeable meta-assets allow participants to exchange digital spaces, objects, and names, fostering virtual commerce and community formation.

However, tradeability must remain bounded by trust, provenance, and governance.

DP5 requires:

- transfer receipts
- provenance preservation
- dispute visibility
- constraints on reserved or protected names
- compatibility with commerce and incentive systems (DP6, DP9)
- alignment with identity and accountability requirements (DP1)

A name may be tradeable, but the trust attached to the name cannot be treated as a blank commodity. Reputation, history, and community meaning must remain visible across transfer.

15. Governance, Accountability, and Agency Surfaces

This section defines **interface-level namespace governance**, aligning with the Meta-Layer's overlay architecture.

Namespaces are not governed only in registries. They are governed at the **point of interaction** via overlays, filters, and community rules.

15.1 Participant-facing surfaces

Participants MUST be able to:

- claim names under transparent rules

- see resolution paths (who resolved this name?)
- inspect ownership and controller state
- view trust signals and classification (human, agent, org)
- transfer names with receipts
- initiate disputes
- view dispute status in real time

Failure mode: invisible governance.

15.2 Overlay-mediated governance

Meta-Layer overlays SHOULD expose:

- name provenance tooltips
- impersonation warnings
- namespace conflicts
- transfer history
- community annotations

This turns naming into a **live civic surface**, not a backend database.

Failure mode: governance hidden behind APIs.

15.3 Community governance powers

Communities MUST be able to:

- reserve names
- define namespace policies
- classify entities
- enforce local naming norms
- quarantine suspicious identities
- fork namespace rules when needed

Failure mode: centralized naming authority.

15.4 Dispute visibility

All conflicts MUST be visible as states, not silent overrides:

- competing claims

- impersonation reports
- trademark conflicts
- governance disagreements

Failure mode: silent winner-take-all resolution.

15.5 Interface as enforcement boundary

DP5 aligns with the principle that governance must live at the interface layer, where users experience identity and trust.

Browser overlays act as **civic membranes** where naming rules become visible, contestable, and enforceable in real time.

Failure mode: governance only enforced off-screen.

16. Incentives and Power Analysis

Namespaces concentrate value at three points, and each attracts a distinct form of capture. Understanding where the money and leverage sit explains why DP5's structural requirements – multi-source resolution, visible state, transfer receipts, registry snapshots – are economic interventions rather than technical preferences.

Registration is where scarcity is manufactured. Names are a positional good: `@jaime` has value precisely because no one else can hold it. Any registry therefore issues a monopoly with every name, and holds the power to define what may be issued, reserved, priced, or refused. Squatting is the market's rational response to unpriced scarcity, and anti-squatting mechanisms – staking, renewal, decay, community challenge – are attempts to price holding without pricing out legitimate participants. The tension is unresolved and named as such under *Open Questions and Future Work*: mechanisms that deter hoarding also burden the communities and individuals least able to pay.

Resolution is where the deepest power sits, and it is the least visible. A resolver that becomes the default for most clients does not need formal authority over the namespace; it exercises effective authority by determining what most participants see. This is the same dynamic through which DNS registrars and app-store search became control points despite holding no

formal ownership. DP5's requirement for multi-source resolution and exposed resolver provenance exists because *default position is the asset*, and because capture at this layer is invisible to the participant by construction. A single commercial indexer dependency reproduces platform risk under decentralized branding.

Transfer is where accountability leaks. A tradeable name accumulates trust and can then be sold, which means reputation built by one party can be exercised by another. Ownership laundering is not an exotic attack but the predictable arbitrage of this gap: buy the history, discard the obligation. Transfer receipts and visible lineage exist to make trust non-fungible even where the name is fungible.

Several further pressures recur:

- **Speculative capital outruns civic use.** Names with obvious community value are claimed before the community exists to claim them. Reservation rules are a redistribution of the option value from speculators to prospective communities.
- **Registry economics favor amnesia.** Storing disputes, quarantines, and full ownership lineage costs money and creates liability; discarding it is cheaper and reduces exposure. Append-only logs with independent archival exist because the incentive runs toward forgetting.
- **Verification becomes a rent.** Where one attestation path is the only credible route to a verified state, verification becomes a tollgate on legitimacy. Plural control derivation – keys, DIDs, registry attestations, community governance – is the counterweight.
- **Agent identity is economically attractive to obscure.** An automated actor that reads as human or as an established organization captures trust it did not earn. Mandatory entity classification is an incentive constraint on whoever deploys agents at scale.
- **Fragmentation serves incumbents.** Where namespaces do not interoperate, each becomes a captive market. Interoperability reduces the value of holding a namespace and is therefore under-supplied by the actors best positioned to provide it.

Example: A registry publishes open registration rules and charges nominal fees, while its own resolver ranks names it has commercial relationships

with above equally valid alternatives. Nothing in the naming rules changed; the namespace was captured at the resolution layer.

Why this matters: Naming power is exercised through defaults, ordering, and memory far more often than through explicit denial. DP5 therefore requires that resolution be inspectable, state be explicit, and history be preserved – not because operators are presumed hostile, but because these are the surfaces where capture is otherwise undetectable.

17. Community Signals Informing DP5

Community submissions and aligned work point to several recurring signals:

- desire for publisher-controlled trust anchors, such as decentralized URIs under existing domains
- need for namespace rights and management for non-human entities and AI systems
- interest in hierarchical, semantically meaningful names independent of physical nodes
- support for trust-schema-driven semantics and context-aware permissions
- demand for name chains as resolvable trust anchors for people, objects, and assertions
- need for decentralized identifiers and tags for posts, paths, and navigation objects
- concern that community identifiers may be seized, censored, or captured by platforms

These signals indicate that DP5 is not only about naming people. It is about naming the structure of the Metaweb itself.

18. Foresight and Failure Design

Namespaces fail slowly and then all at once. For long periods a naming system appears healthy – names resolve, transfers settle, disputes are rare – while the conditions for its capture accumulate beneath the surface. By the time failure is visible, the namespace is usually load-bearing for identity,

commerce, and provenance across many systems, which makes correction expensive and migration nearly impossible.

Predictable degradation paths include:

- **Resolver consolidation.** Multi-source resolution is implemented, but clients converge on one resolver for latency, cost, or convenience. The plural architecture remains intact while the effective architecture becomes singular.
- **State flattening.** Provisional, disputed, and quarantined states are supported by the protocol but collapsed to a single indicator in interfaces, because nuance is hard to render. Participants lose the ability to distinguish a verified name from a contested one.
- **Snapshot gaps.** Registry snapshots are produced regularly until an outage, migration, or cost review interrupts the cadence. Nothing appears broken, but a window of history becomes unreconstructable – and windows are exactly where laundering hides.
- **Reserved-set erosion.** Reservations protecting communities, institutions, or safety-critical names are relaxed case by case under commercial pressure, until the reserved set no longer reflects the risks it was built for.
- **Similarity-detection decay.** Spoofing defenses are calibrated against the confusables and scripts known at build time, while attackers move to newly supported characters, scripts, or rendering contexts.
- **Semantic profile abandonment.** Meaning-layer data is populated at registration and then left to rot, so trust signals attached to a name increasingly describe a state that no longer exists.
- **Quarantine backlog.** Suspicious names are quarantined faster than review capacity can adjudicate them, and the backlog is eventually cleared in bulk to restore throughput.

These paths compound in a characteristic sequence. Resolver consolidation makes state flattening consequential, because a single resolver's rendering becomes the participant's only view. Snapshot gaps make ownership laundering undetectable. Undetectable laundering makes semantic profiles unreliable, which makes state signals meaningless, which returns participants

to judging names by appearance – the precise condition DP5 exists to escape.

Two properties make namespace failures unusually severe:

Irreversibility of trust transfer. Once a name with accumulated trust has been used to deceive, the harm is distributed across everyone who relied on it, and cannot be recalled by revoking the name. Prevention dominates remedy.

Lock-in by dependency. Names become embedded in links, credentials, contracts, archives, and other systems' data. A namespace that must be abandoned takes that dependent structure with it. This is why forkability, snapshot integrity, and cross-namespace interoperability are safety features rather than conveniences.

DP5 therefore requires safeguards designed in advance:

- **Resolver diversity monitoring**, treating concentration of resolution traffic as a measurable risk indicator rather than a market outcome
- **State-rendering conformance**, so that a compliant interface cannot silently collapse distinct name states into undifferentiated trust
- **Snapshot continuity auditing**, with checksums verified against independent archives rather than self-reported
- **Adversarial similarity testing** on a recurring schedule as scripts, fonts, and rendering surfaces change
- **Quarantine capacity provisioning** proportional to registration volume, since review capacity is the binding constraint on abuse response (DP3, DP17)
- **Fork drills**, in which a community exercises the ability to fork namespace rules and carry its records, so that the exit option is known to work before it is needed
- **Public postmortems** for namespace incidents – a successful spoof, a laundered transfer, a resolution override – linked to the specific layer that failed

Failure is expected. Failure that becomes visible only after the namespace is load-bearing is not. A DP5-aligned namespace is one where the concent-

ration of resolution power, the erosion of state visibility, and the loss of registry memory are all observable while they are still reversible.

19. Relationship to Other Desirable Properties

DP5 is foundational and interdependent.

- **DP1** depends on identifiers that bind identity and accountability without forcing platform control
- **DP2** depends on participant control over names, handles, personas, and namespace portability
- **DP4** depends on identifiers that preserve data provenance without forcing cross-context correlation
- **DP6** depends on tradeable meta-assets, domains, and artifacts that preserve ownership and settlement integrity
- **DP7** depends on canonical forms and resolution semantics that work across systems
- **DP9** depends on attribution and contribution identifiers that cannot be trivially spoofed or replayed
- **DP12–DP13** depend on agent and tool identifiers that can be constrained, audited, and governed
- **DP14–DP15** depend on provenance and transparency for names, artifacts, and registry changes
- **DP18–DP20** depend on community and ownership identifiers that persist across governance and migration

Two couplings deserve emphasis, because DP5 is where they become concrete rather than aspirational.

DP3 governs the namespace itself. Reserved sets, dispute procedures, quarantine thresholds, fork legitimacy, and canonicality rules are all governance artifacts, subject to DP3's requirements for tiered decisions, receipts, visible diffs, bounded emergency authority, and memory. A registry that changes its reserved set or ordering rules without a traceable, contestable process has made a governance decision outside governance.

DP23 determines who the namespace is for. Unicode support, multilingual naming, and script integrity are the difference between a namespace that

serves a global population and one that serves the subset writing in Latin script. This sits in direct tension with spoofing resistance, since script diversity expands the confusable space – a tension DP5 names openly rather than resolving by exclusion.

A failure in DP5 propagates upward. If names cannot be trusted, ownership cannot be trusted, identity fragments, and interoperability becomes deception.

20. Non-Goals and Explicit Boundaries

DP5 does not:

- require one global namespace for all contexts
- replace DNS, DIDs, ENS, SNS, or publisher-controlled URIs
- guarantee that all valuable names will be available
- eliminate disputes, squatting, or fraud completely
- treat tradeability as superior to stewardship
- require real-name identity
- collapse human, AI, organizational, and artifact namespaces into one undifferentiated model

DP5 further does not:

- prescribe a pricing, staking, or renewal model; it requires that whichever model is adopted be visible and governed
- guarantee that a name means the same thing in every context; it requires that divergence be signaled rather than hidden
- treat blockchain anchoring as a requirement, only as one available mechanism for durable provenance
- adjudicate trademark or legal ownership claims; it makes such conflicts visible as states and leaves adjudication to competent authorities and community process
- promise that a name, once held, is held permanently; expiry, abandonment, retirement, and forking are legitimate states

Failure mode: **namespace maximalism**, where DP5 is invoked to justify a single canonical registry, on the grounds that coherence requires centrali-

zation. DP5 requires interoperable naming with explicit trust semantics, which is the opposite claim.

DP5 defines conditions under which naming remains interoperable, governable, and accountable across systems.

21. Minimum DP5 Alignment

This section states **testable compliance criteria** rather than descriptive aspirations.

A system claiming DP5 alignment MUST pass the following checks:

21.1 Deterministic naming

- Canonical forms are defined and machine-verifiable
- Validation produces identical results across implementations

Test: same input → same validity result everywhere

21.2 Multi-source resolution

- Names resolve via more than one possible source
- Resolver provenance is exposed

Test: user can inspect where resolution came from

21.3 Explicit state signaling

Every name MUST expose state:

- verified
- provisional
- disputed
- quarantined
- forked

Test: UI or API returns state metadata

21.4 Ownership auditability

- Current controller is identifiable
- Transfer history is preserved

Test: ownership lineage query returns full chain

21.5 Portability

- Names function across at least two independent systems

Test: same name resolves meaningfully in multiple contexts

21.6 Anti-spoofing safeguards

- System detects or flags similarity-based impersonation

Test: registering visually similar name triggers warning or constraint

21.7 Registry memory

- Historical snapshots exist with integrity proofs

Test: past state can be reconstructed with checksum validation

21.8 Dispute mechanism

- Users can initiate and observe disputes

Test: dispute creates visible state transition

21.9 Non-human classification

- Agents and automated systems are distinguishable from humans

Test: entity type is explicitly encoded and exposed

These criteria transform DP5 from a principle into a **verifiable standard**.

These conditions define the minimum viable namespace layer of the Meta-Layer.

22. Open Questions and Future Work

DP5 identifies key frontier problems:

22.1 Cross-namespace interoperability

How do independent naming systems interoperate without collapsing into monopoly or fragmentation?

22.2 Semantic portability

How can meaning travel with names across cultural, linguistic, and technical contexts?

22.3 Anti-squatting mechanisms

What mechanisms balance open access with protection against speculative enclosure?

22.4 AI-native identity

How should agent identities evolve as they gain autonomy, persistence, and economic activity?

22.5 Namespace governance forks

What legitimacy models determine when a community can fork naming rules?

22.6 Human-readable vs machine-secure naming

How do we balance usability with cryptographic robustness?

22.7 Unicode and global inclusion

How do we support multilingual naming without increasing spoofing risk?

22.8 Economic design

What pricing, staking, or decay mechanisms prevent hoarding while preserving ownership?

22.9 Resolution concentration

What measures of resolver diversity are meaningful, and what interventions are available when resolution traffic concentrates without any formal change in authority?

22.10 Namespace succession

What happens to names, artifacts, and communities when a registry, resolver, or community steward ceases to operate, and how is continuity established without a central custodian of last resort?

These questions define the path toward ML-RFC maturation.

23. Path Toward ML-RFC

Progression toward ML-RFC maturity should include:

- standardized canonical forms for meta-domains, personal identifiers, and artifact identifiers
- shared validation rules and reserved-name policies
- resolver and registry schemas
- dispute-state semantics
- transfer receipt formats
- registry snapshot standards with checksums
- conformance tests for resolution integrity, ownership binding, and namespace portability
- implementation evidence from registry prototypes and overlay applications

Stable portions may be promoted first, especially canonical form rules, validation logic, and registry state semantics.

24. Closing Orientation

DP5 is the claim that participants, communities, agents, and artifacts deserve names they can carry across the Meta-Layer.

Without sovereign names, identity is rented, ownership is fragile, and discovery remains platform-shaped.

With DP5, people can claim space, communities can preserve continuity, artifacts can be addressed, and the Metaweb can become navigable without surrendering naming power to a single platform.

DP5 turns names into civic infrastructure.

To claim your space in the Meta-Layer is not merely to register a label. It is to anchor presence, meaning, and accountability in a shared world.

Continue the conversation on DP5 online →

DESIRABLE PROPERTY 6

DP6

Commerce

Fair value, everywhere – the Meta-Layer supports seamless, flexible transactions, empowering communities and individuals to earn, exchange, and sustain themselves on their own terms.



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp06

1. Purpose of This Draft

This draft articulates Desirable Property 6 (DP6) as the condition under which value exchange in the Meta-Layer is fair, legible, and aligned with human and community flourishing, not reducible to extraction, dark patterns, or attention rents dressed as markets.

Commerce is not an add-on to the Meta-Layer. It is how incentives (DP9), ownership (DP20), and sustainability (DP17) become real.

DP6 therefore defines the minimum conditions under which exchange remains trustworthy at the interface where money moves and value is recognized.

Commerce is the highest-value attack surface in the Meta-Layer. Systems that appear fair locally but fail across system boundaries, delegation, or routing layers will be systematically exploited.

DP6 connects to:

- DP4 (data minimization in payments and ads)
- DP2 (participant agency and consent)
- DP9 (incentive alignment)
- DP12 (executable policy at checkout)
- DP13 (containment of automation and abuse)
- DP15 (receipts and provenance)
- DP17 (financial sustainability)

If DP6 is weak, predictable failures follow: surveillance advertising as hidden tax, fee obfuscation, lock-in through proprietary rails, platform capture of creator revenue, and AI-mediated manipulation at conversion.

DP6 does not prescribe a specific payment rail, currency, or chain. It defines legitimacy conditions for commerce in the Meta-Layer.

2. Problem Statement

In today's web, commerce is fused with attention manipulation and opacity.

Participants encounter:

- pricing that cannot be verified before commitment

- fee stacks that are only visible after settlement
- payment flows bundled with surveillance and data extraction
- AI systems that optimize for conversion rather than user welfare

This produces recurring failures:

- surprise charges and coercive conversion flows
- creator and worker value hidden behind opaque fee structures
- failed micropayment systems due to friction and rent extraction
- communities unable to capture value from activity they host

These failures are structural. When transparency and portability are optional, extraction dominates.

DP6 reframes commerce as governed exchange: visible rules, bounded automation, and enforceable fairness at the point of transaction.

3. Threats and Failure Modes

3.1 Fee fog and stacked take rates

Participants cannot see effective take rates across layers.

Example: A creator sees a platform fee but not downstream processing, ranking, or conversion costs.

Why this matters: Fairness requires full distribution visibility.

3.2 Dark patterns at conversion

Design exploits cognitive load to increase conversion.

Example: A free trial converts with hidden renewal and hard-to-find cancellation.

Why this matters: Coercive commerce violates DP2 agency.

3.3 Attention rents disguised as markets

"Free" is subsidized by hidden data extraction.

Example: Ad SDKs exfiltrate unrelated data to fund access.

Why this matters: Externalities shift cost to users without consent (DP4).

3.4 Lock-in through wallets and closed loops

Value cannot exit proprietary rails.

Example: Credits and reputation cannot be ported without loss.

Why this matters: Undermines DP7 interoperability and DP4 export.

3.5 AI-mediated financial harm

Agents steer decisions without accountability.

Example: Assistants recommend higher-commission products as "best."

Why this matters: Requires DP11 disclosure and DP13 bounds.

3.6 Community extraction without reciprocity

Economic activity uses community trust without funding it.

Example: Marketplaces leverage forums without contributing to moderation or safety.

Why this matters: Violates DP20 ownership and DP17 sustainability.

3.7 Cross-border confusion

Fees, taxes, and currencies are unclear.

Example: Display currency differs from settlement with hidden spreads.

Why this matters: Legibility must include jurisdictional honesty.

3.8 Predatory targeting

Dynamic pricing or credit targets vulnerable users.

Example: BNPL prompts cluster around financially stressed users.

Why this matters: Intersects DP4 inference limits and DP11 ethics.

3.9 Platform self-preferencing in AI commerce

Default assistants route to house inventory.

Example: "Buy now" prioritizes sibling brands under "personalization."

Why this matters: Requires disclosure and contestability.

3.10 Cross-system commerce distortion

Commerce terms, attribution, or protections change or become exploitable when transactions, identities, or value move across systems.

Example: A checkout with full fee disclosure exports to a partner flow where additional fees are added post-commitment, or a refund policy is not honored after handoff.

Why this matters: Commerce that fails at system boundaries enables arbitrage and hidden extraction.

3.11 Community economic capture

A community's economic surface is controlled by a subset of its participants, or by the operator hosting it, such that surcharges, treasury allocation, or vendor admission serve narrow rather than collective interests.

Example: A community treasury funded by transaction surcharges is directed by a small group of long-tenured members toward projects they benefit from, with no rotation, disclosure, or contestation pathway.

Why this matters: Giving communities economic power creates a new capture surface. The remedy is not to withhold the power but to bind it to governance (DP3) and ownership (DP20) conditions. A captured community economy is harder to exit than a captured platform, because the social relationships and the economic relationships are the same.

4. Core Principle

Commerce in the Meta-Layer is fair only when pricing, fees, risks, and responsibilities are legible at the point of exchange; defaults are non-exploitative; automation and AI assistance are accountable and bounded; and communities can shape and, where appropriate, capture economic activity that depends on shared trust and infrastructure.

Example: A checkout shows item price, full fee breakdown, data uses tied to the transaction, AI involvement, cancellation path, and any community surcharge with a link to governance.

What this feels like: Paying feels like choosing, not being herded.

Commerce guarantees must remain valid not only within a single interface, but across the systems, tools, and contexts where transactions are initiated, routed, and settled.

When commerce flows cross systems, pricing, fees, policies, and responsibilities must either:

- persist with integrity, or
- degrade in a way that is visible and contestable

Commerce that becomes opaque at handoff points is structurally unsafe.

5. Primary Mechanisms and Structural Conditions

5.1 Commerce Layer: Execution, Proof, and Settlement

Commerce in the Meta-Layer cannot rely on interface clarity alone. It must be anchored in a substrate that binds pricing, allocation, and settlement to verifiable, enforceable structures. Without this, even well-designed interfaces can be subverted downstream, where actual value movement occurs.

DP6 therefore defines a commerce layer composed of primitives that make exchange legible, auditable, and governable across the full lifecycle of a transaction, from intent to settlement to dispute.

5.1.1 Transaction objects

Transactions are not ephemeral UI events. They are structured, machine-readable objects that encode the full economic reality of an exchange. This includes itemized pricing, discounts, fee distribution, data use, jurisdictional context, and the governing policies at the moment of commitment.

By binding these elements together, transaction objects ensure that what a participant sees at checkout is not merely descriptive, but enforceable. This is what connects commerce to executable policy (DP12).

A failure mode here is selective omission or delayed disclosure, where certain fees or conditions are only materialized after commitment. Systems must therefore ensure that transaction objects are complete before commitment and immutable afterward except through governed updates.

5.1.2 Settlement and flow proofs

Settlement must be provable, not assumed. Participants and communities need to be able to trace how value moved: who received what, when, and under what conditions.

Settlement proofs create continuity between intent and outcome. Without them, systems can display one set of expectations at checkout and execute another at settlement.

Failure modes include delayed settlement visibility, partial disclosure of recipients, or selective omission of intermediary captures. Systems must treat settlement as a first-class observable state, not a backend detail.

5.1.3 Commerce policy binding

Zones and communities attach policy objects directly to transactions. In plain terms, a rule that is bound to a sale must be followed when the sale happens. These rules are not advisory; they execute at runtime and shape what is allowed, required, or disallowed in a transaction.

This includes restrictions on categories, required disclosures, surcharge logic, and AI usage constraints. The key shift is that commerce behavior is governed at the point of execution, not after harm occurs.

The adversarial pressure here is policy bypass through integration pathways. Transactions must carry their governing policy context with them across systems or explicitly signal where enforcement no longer holds.

5.1.4 AI disclosure and constraint hooks

AI participation in commerce must be visible, attributable, and bounded. This includes pricing adjustments, ranking decisions, bundling, and negotiation.

Participants must be able to distinguish between neutral presentation and optimized persuasion. Without this, AI becomes a hidden incentive layer that distorts economic decisions.

Participants should always know when a page, listing, or recommendation is wholly or partly generated by AI, before they rely on it to buy or sell.

A critical failure mode is undisclosed optimization, where AI systems steer users toward outcomes that maximize commission or retention while presenting themselves as neutral assistants.

Example: An AI assistant helps a person compare repair shops. It says that it is an AI. It says when a shop pays it a referral fee and gives the main reason each result was shown. It can draft a booking, but it cannot spend more than the person's set limit without asking first. The receipt lists the AI's role, the referral fee, the price, the cancellation terms, and who is responsible for doing the work.

5.1.5 Commerce memory

Commerce systems must retain a structured memory of transactions and policy evolution. This includes prior fee structures, changes in rules, and the outcomes of disputes.

This historical layer enables accountability over time and allows communities to detect drift toward extraction or manipulation.

Without commerce memory, each transaction exists in isolation, making systemic issues difficult to detect and correct.

These primitives transform commerce from a series of isolated interactions into a coherent, governable system that can withstand adversarial pressure and evolve over time.

5.2 Fee and take-rate transparency

Fee transparency is not simply about showing a number. It is about ensuring that participants can understand the full economic structure of a transaction before they commit.

In many systems, fees are fragmented across layers (platform fees, payment processing, ranking or promotion adjustments, community-level surcharges, or currency spreads), each of which may be individually legible but collectively opaque. This creates a situation where the true cost of participation cannot be known until after settlement.

DP6 requires that effective fees be computable and attributable before commitment, not reconstructed after the fact. Participants must be able to see

the full take rate across all layers, and how value is distributed among them, in a way that is coherent and comparable.

A core failure mode is fee fragmentation, where systems distribute costs in ways that obscure total extraction. Systems must therefore recombine all fee components into a clear, pre-commitment view of total cost.

5.3 Honest defaults and reversal paths

Defaults are one of the most powerful levers in commerce systems. When defaults are misaligned, even transparent systems can become coercive in practice.

DP6 requires that material commitments be opt-in, not opt-out, and that reversal paths such as cancellation or refund follow the same level of friction as signup. This creates symmetry between entry and exit, which is essential for real agency (DP2).

A common failure mode is asymmetrical friction, where signup is immediate but cancellation is buried, delayed, or requires additional steps. Systems must treat reversibility as a first-class design constraint, not a secondary feature.

Example: A person joins a learning service. Before they pay, one screen shows the monthly price, the payment fee, any community charge, the renewal date, the data the service needs, the data it would like but does not need, and a cancel link. They can cancel from the same account page they used to join. Their receipt keeps the terms that applied that day, so they have proof if there is a dispute.

5.4 Separation of payments and surveillance

Commerce systems often bundle payment with data extraction, turning transactions into opportunities for surveillance. This creates hidden costs that are not reflected in price.

DP6 requires that payment does not require unrelated data processing (DP4). The data required to complete a transaction must be limited to what is strictly necessary, and any additional data use must be explicitly disclosed and optional.

A key failure mode is covert bundling, where data collection is technically optional but practically unavoidable. Systems must ensure that participants can complete transactions without consenting to unrelated data flows.

Payment privacy should also be evaluated at the transport carrying a transaction. Optional decentralized routing can reduce exposure of transaction metadata, but its limits and dependencies must remain visible. DP4 treats a privacy mesh as an implementation option rather than a guarantee of anonymity.

5.5 Interoperable value rails

Value must be able to move without losing meaning, ownership, or accountability. When value is trapped within proprietary systems, participants are subject to platform-defined rules that cannot be contested or exited.

DP6 therefore prefers open protocols and requires export and audit capabilities for closed systems (DP7). Participants must be able to leave economic relationships without disproportionate friction, moving balances, entitlements, receipts, and transaction history without losing integrity.

A primary failure mode is economic lock-in, where value can technically be withdrawn but at significant loss or friction, often because records, identity, or value are fragmented. Systems that make exit costly create economic capture even when pricing appears fair; they must treat portability as a constraint on extraction, not an optional feature.

Interoperable rails also mean keeping the price a person sees separate from the tool they use to pay. A sale can show its price in local money, and the buyer still pays with any lawful method that works where they live. Before the buyer agrees, the system shows the exchange rate, when it was set, how long it holds, any conversion cost, and the final amount, and it never requires one coin, wallet, or chain.

5.6 Creator and worker fairness

Commerce systems depend on contributors whose work generates value, yet those contributors are often the least protected participants in the system.

DP6 requires that attribution and payouts be tamper-evident and that disputes be resolved in a timely and transparent manner. Contributors must be able to verify how their work is valued and compensated.

A failure mode here is delayed or opaque payout logic, where contributors cannot trace how their compensation was calculated or why it changed. Systems must ensure that payout logic is both visible and contestable.

5.7 Community economic surfaces

Communities create the conditions under which commerce is trusted, yet often lack the ability to shape the economic activity that depends on them.

DP6 enables zones to impose rules, surcharges, or bans with executable policy (DP12). This allows communities to align commerce with their values and to capture a portion of the value generated within their environments.

A key failure mode is extraction without reciprocity, where economic activity leverages community trust without contributing to its maintenance. Systems must ensure that communities can define and enforce economic participation terms.

Community economic surfaces should support mutual aid and private peer-to-peer value exchange, including for people excluded from or poorly served by conventional financial infrastructure. Privacy of communication supports that use case but does not by itself resolve settlement, fraud recourse, or access to the necessary services.

5.8 High-stakes commerce pathways

Not all transactions carry the same level of risk. High-stakes categories such as financial products, healthcare, or legal services require additional safeguards.

DP6 requires human confirmation or expert gating for sensitive categories. This ensures that automation and AI do not make consequential decisions without appropriate oversight.

A failure mode is over-automation, where systems optimize for efficiency at the cost of safety. Systems must introduce friction where necessary to prevent harm.

5.9 Sustainability linkage

Commerce systems do not exist in isolation. They depend on shared infrastructure, communities, and public goods that must be maintained over time.

DP6 requires that fees transparently fund commons maintenance (DP17). This creates a visible link between economic activity and the sustainability of the systems that support it.

A failure mode is invisible extraction, where value is removed from ecosystems without reinvestment. Systems must make sustainability contributions explicit and traceable.

Example: A community market adds a 1% charge to sales in its zone, and the bill shows it before payment. The charge has a cap and a stated purpose. Every three months the community shares a short report on how the money paid for moderation, translated safety guides, accessibility work, and dispute help. Members can read the report, question a choice, vote on the next budget or renewal of the charge, and keep their receipts if they leave.

5.10 Receipts and dispute evidence

Receipts are not merely confirmations of payment. They are the foundation of accountability in commerce systems.

DP6 requires machine-readable receipts that support fair resolution (DP15). Every transaction should produce an evidence bundle: signed receipts, the policy state at the time of purchase, and clear identification of responsible parties, with enough detail to reconstruct the transaction and its governing conditions. This shifts dispute resolution from subjective interpretation to evidence-based adjudication.

Failure modes include incomplete or unverifiable receipts, missing policy context, and ambiguity about responsibility across multi-party transactions, which make disputes difficult or impossible to resolve. Systems must treat receipts as evidence, not just records.

5.11 Accessibility of economic surfaces

Commerce must be accessible to all participants, regardless of ability, device, or connectivity constraints.

DP6 requires that checkout and transaction flows work across assistive technologies and low-bandwidth contexts. Accessibility is not only a usability concern, but a fairness constraint.

Economic accessibility also includes affordability. Participants should be able to see fees, surcharges, and minimum commitments for core participation, so a technically available capability does not become accessible only to affluent users. This does not require every optional transaction or service to be free.

A failure mode is exclusion by design, where systems assume high bandwidth, modern devices, or specific interaction patterns. Systems must ensure that economic participation is not gated by technical privilege.

5.12 Cross-system commerce integrity (DP7 alignment)

Commerce systems must preserve the relationship between price, fee, policy, and settlement across environments.

This includes:

- maintaining itemized pricing and fee visibility across handoffs
- preserving policy bindings (refunds, cancellations, disclosures) across integrations
- ensuring receipt and provenance continuity across tools (DP15)
- signaling when guarantees or protections change in a new environment

Commerce must not be portable in ways that enable hidden fees, policy resets, or accountability gaps.

5.13 Agent-to-agent commerce integrity

Commerce is increasingly mediated not just by humans, but by agents acting on behalf of participants. These agents may search, negotiate, bundle, and execute transactions across multiple systems without direct human interaction at each step.

DP6 requires that agent-to-agent commerce remains legible, bounded, and accountable at the same level as human-facing transactions.

This includes:

- ensuring agents carry explicit mandates, constraints, and budget limits from their principals
- preserving transaction visibility such that participants can inspect decisions made on their behalf
- binding agent actions to the same policy, fee, and disclosure requirements as direct transactions
- preventing agents from exploiting speed, scale, or opacity to bypass safeguards

A critical failure mode is delegated opacity, where agents transact in ways that are technically valid but practically uninspectable, allowing hidden fees, biased routing, or exploitative bundling.

Systems must ensure that delegation does not reduce accountability. Agent-mediated commerce must remain reconstructable, auditable, and interruptible by participants and governance systems.

6. Meta-Community Economies

The mechanisms above describe fairness conditions at the level of a transaction. This section describes what becomes possible when communities themselves hold economic surfaces: the ability to fund what they depend on, to set terms for economic activity that relies on their trust, and to retain value that would otherwise flow entirely outward.

A **meta-community economy** is the economic layer of a community zone. It exists because communities generate the conditions that make commerce trustworthy (moderation, reputation, curation, norms, dispute resolution, and the social relationships within which exchange feels safe), and because in current systems that contribution is uncompensated by default. Marketplaces leverage forums; recommendation systems harvest curation; platforms monetize trust they did not build. *Community extraction without reciprocity* is not a marginal harm; it is the normal operating condition.

6.1 What a community economy comprises

- **A treasury or common pool**, funded through zone surcharges, membership contributions, revenue shares on activity conducted in the zone, or grants, with allocation governed by the community rather than the host.
- **Zone economic policy**, expressed as policy objects that execute at transaction time: which commerce categories are permitted, which require disclosures, what surcharge applies, which vendors are admitted, and what AI participation is allowed at conversion.
- **Contribution accounting**, recording who produced the value the zone depends on (moderation, curation, translation, dispute resolution, infrastructure), so that distribution is grounded in a record rather than in social standing. This is where DP6 meets DP9. The value a zone depends on also includes creating and verifying connections among artifacts (DP9) and designing or implementing the application substrate. Contributors should be able to inspect how the record relates to what they receive, whether reciprocity, gift, redistribution, barter, direct non-monetary exchange, or fiat or token rewards; on-chain recording is one option, not a requirement.
- **Reciprocity terms**, defining what external commercial actors owe when they conduct business inside the zone or rely on its trust signals, and making that obligation enforceable rather than voluntary.
- **Inter-community settlement**, allowing value, credits, or obligations to move between zones without requiring a single currency or clearing authority, and without losing provenance in transit.

6.2 Composition across zones

Communities are not economically isolated. A participant may belong to several zones with different economic rules, and a transaction may touch more than one. DP6 requires that when zone economic policies compose, the result be determinate and visible rather than emergent.

Two rules follow. First, **surcharges and constraints must be enumerable at checkout**: a participant sees each zone's contribution as a distinct line, not a blended total, so that what they are funding is legible. Second, **the stricter**

constraint governs: where one zone prohibits a category another permits, the prohibition applies to transactions within its scope. A zone's economic policy may raise the floor of protection for its participants; it may not lower another zone's floor.

Example: A sale takes place in both a general market and an education zone. The education zone bans gambling offers and requires clear warnings on financial products, so those rules apply to the sale. The education zone also adds a small charge for moderation and access tools, and the bill shows it on its own line with a short reason. Protections stack toward the stronger rule; charges never blend into one total.

6.3 Sustainability rather than rent

The distinction between a community economy and a rent-seeking intermediary is not the presence of a fee. It is whether the fee is bound to the maintenance of something participants depend on, and whether that binding is verifiable.

DP6 therefore requires, for community economic surfaces:

- **Traceable use of funds.** A surcharge is linked to what it maintains (moderation capacity, appeals capacity, infrastructure, translation, accessibility work), with settlement proofs that participants can inspect. This is the same requirement as *Sustainability linkage*, applied at community scale.
- **Proportionality to contribution.** A community's claim on economic activity should bear a defensible relationship to what the community actually provides. A zone that hosts a transaction and enforces its rules has a stronger claim than one that merely happens to be adjacent.
- **Exit that preserves value.** Participants and vendors must be able to leave a zone economy with their receipts, history, and where supported their balances or entitlements. Without this, community economic power becomes lock-in with better rhetoric.
- **Governed allocation.** Treasury decisions are governance decisions, subject to DP3's requirements for tiered thresholds, receipts, visible diffs, and revocable delegation, and to DP20's ownership conditions.

6.4 The capture problem

Giving communities economic power creates a new capture surface, described above under *Community economic capture*. This is the central structural risk of this section and cannot be designed away, only bounded.

The bounds DP6 requires are:

- **Visible allocation.** Where treasury funds go, and who decided, is inspectable by every participant subject to the surcharge that funded them.
- **Rotation and mandate limits** on the roles controlling economic policy and allocation, since economic authority entrenches faster than moderation authority.
- **Contestable admission.** Vendor admission and exclusion decisions are appealable, because the power to exclude from a market is more consequential than the power to exclude from a conversation.
- **Forkability with economic continuity.** A community that disagrees with how its economy is being run must be able to fork, carrying contribution records and member continuity, or the disciplining effect of exit disappears (DP20).
- **Audit without surveillance.** Aggregate economic outcomes (concentration of payouts, vendor diversity, surcharge burden distribution) should be reviewable without exposing individual transaction histories (DP4).

6.5 Why this matters

Without community economies, the Meta-Layer reproduces the current arrangement in new architecture: communities supply trust, platforms and intermediaries capture its value, and the work of maintaining a habitable space is performed for free until the people performing it stop.

With them, a community can fund its own moderation, compensate its curators and translators, set terms for commerce that depends on its reputation, and sustain itself without a host extracting the difference. That is the point at which DP6 stops being a set of protections against bad commerce and becomes infrastructure for economic self-determination, and the point

at which DP17's sustainability and DP20's ownership become achievable rather than aspirational.

6.6 Optional infrastructure service markets

Where a privacy-routing service charges for use, it could offer session- or bandwidth-based pricing, including token-based microtransactions, with visible shares for operators and a community treasury. Participant-chosen third-party services could use opt-in revenue-sharing agreements without making such agreements a prerequisite for interoperability.

The choice between public funding, participant payment, and mixed models should preserve access and avoid surveillance-based monetization unless opted into by the participant. DP17 addresses whether relay revenue can cover continuing costs.

7. Governance, Accountability, and Agency Surfaces

Commerce is not neutral infrastructure. It encodes choices about power, risk, and value distribution, often in ways that are invisible to participants. In many systems, these choices are embedded in defaults, routing logic, or fee structures that cannot be contested or even observed.

DP6 requires that these choices become visible and governable at the point of transaction. This means participants are not only protected from harm, but able to understand and shape the economic conditions they are subject to.

Without these surfaces, commerce systems revert to extraction: participants transact, but cannot see how value flows, cannot challenge outcomes, and cannot meaningfully exit or redirect the system.

Participants must be able to:

- see true prices, fees, and net outcomes before commitment
- understand what data is used in the transaction and revoke unrelated scopes (DP4)
- identify AI involvement in pricing, ranking, or negotiation (DP11)
- access clear cancellation, refund, and dispute pathways
- export receipts and transaction history where supported (DP7, DP15)

Participants must also be able to understand how commerce terms change across systems.

This includes:

- whether pricing and fee structures persist across integrated flows
- whether policies such as refunds, disputes, and disclosures carry over
- where accountability shifts between parties during multi-system transactions

Participants must not be required to reverse-engineer commerce conditions across environments.

Communities must be able to:

- define allowed and disallowed commerce patterns within zones
- impose disclosures, caps, or surcharges via executable policy (DP12)
- audit aggregate outcomes for fairness, capture, and harm (without exposing individuals)
- evolve rules with memory of prior incidents and outcomes (DP12 governance loops)

Example: A community zone bans predatory lending ads and requires fee disclosures for all financial products. Enforcement occurs at runtime through policy binding, not moderator memory.

8. Incentives and Power Analysis

Commerce determines where value accumulates. Incentives determine how that value is pursued.

In many systems, pricing and ranking are optimized for conversion and revenue, while governance claims prioritize safety or fairness. This creates a structural mismatch where economic incentives override stated values.

DP6 requires incentive legibility within commerce systems:

- how ranking, promotion, or bundling affects price and visibility
- how commissions, fees, or partnerships influence recommendations
- how optimization targets (conversion, revenue, retention) shape outcomes

Example: A marketplace ranks products based on commission rather than relevance, while presenting results as "best match."

Why this matters: When incentives are hidden, markets become extraction systems. When visible, they become governable.

DP6 therefore expects:

- disclosure of materially relevant incentive structures
- the ability for communities to constrain or rebalance those incentives
- alignment between commerce incentives and governance rules (DP3, DP12)

Power also concentrates when commerce fragments across systems.

Actors who control routing, aggregation, or interface layers can reshape pricing, visibility, or policy without formally changing them.

DP6 therefore requires resistance to cross-system arbitrage, where value is extracted through boundary manipulation rather than contribution or service.

Economic design should be evaluated for entry barriers and concentration of control, not assumed to be competitive because it is decentralized. Privacy-respecting services can compete on participant control rather than extraction, supported by incentives alongside regulation, without requiring people to monetize their personal data.

9. Community Signals Informing DP6

Across ecosystems, consistent signals point to structural failures in commerce design. These are not isolated grievances, but recurring patterns that reveal where systems break under real-world use.

Participants repeatedly encounter environments where economic behavior is shaped by hidden incentives, opaque fee structures, and limited recourse. Over time, this produces erosion of trust, disengagement from participation, and migration toward systems perceived as more fair or legible.

Common signals include:

- creators reporting opaque fee stacks and unpredictable income

- users expressing fatigue with subscription traps and hidden renewals
- demand for receipt-level clarity on who was paid and why
- interest in local economic rules for civic, educational, and community spaces
- concern that AI-mediated shopping optimizes for commission rather than user welfare

These signals reflect a deeper pattern: participants are not rejecting commerce itself, but commerce that is misaligned with their understanding of fairness, agency, and contribution.

DP6 treats these signals as design inputs, not complaints. They indicate where economic systems fail to align with human expectations and where intervention is required.

10. Foresight and Failure Design

DP6 assumes that commerce systems will be pressured toward opacity, capture, and manipulation. This pressure increases with scale, competition, and the introduction of automation and agent-mediated transactions.

Common failure paths include:

- reintroduction of hidden fees and bundled costs under competitive pressure
- AI-driven persuasion targeting vulnerable users at conversion points
- consolidation of value within closed payment or loyalty systems
- divergence between displayed prices and actual settlement outcomes

These failures rarely occur as singular events. They emerge gradually as systems optimize for short-term metrics, allowing small deviations from fairness to accumulate into systemic extraction.

DP6 therefore requires designing safeguards in advance:

- circuit breakers for harmful pricing or targeting patterns
- policy-based enforcement at transaction time
- rate limits and constraints on high-risk automation
- public postmortems linking failures to rule and system changes

Designers must also assume that commerce will be attacked at system boundaries.

Common boundary failure paths include:

- hidden fees introduced during cross-system routing
- policy resets when transactions move between providers
- accountability gaps between integrated services
- divergence between displayed and settled outcomes across systems

In agent-mediated environments, additional failure modes emerge:

- agents colluding or routing through opaque pathways to maximize hidden incentives
- delegated decision-making that exceeds user intent or understanding
- rapid transaction loops that bypass human oversight entirely

Community economies introduce a further set of failure paths that DP6 must anticipate:

- **surcharge accretion**, where each individually modest community fee is defensible and the aggregate burden on participants transacting across many zones is not
- **treasury entrenchment**, where control of allocation becomes the most durable form of authority in a community because it is the least visible
- **reciprocity avoidance**, where commercial actors restructure to appear external to a zone while continuing to rely on its trust signals
- **inter-zone rate arbitrage**, where activity migrates to whichever zone imposes the lowest obligations, undermining the zones doing the maintenance work

Each of these is a predictable consequence of giving communities economic power, and each is addressed by the bounds set out under *Meta-Community Economies*: enumerable surcharges, rotation on allocation authority, enforceable reciprocity terms, and audit of aggregate burden distribution.

DP6-compliant systems include detection, signaling, and governance responses for these failures. They treat commerce not as a static system, but

as an adversarial environment that must be continuously monitored and corrected.

Failure is expected. Invisible failure is not.

11. Relationship to Other Desirable Properties

DP6 connects commerce to the full Meta-Layer system:

- DP2 ensures participant agency at checkout and over subscriptions
- DP3 defines how commerce rules evolve through governance
- DP4 constrains data use in payments and advertising
- DP7 enables portability of receipts, balances, and history
- DP9 aligns incentives with non-extractive exchange
- DP11–DP13 bound AI behavior in commerce contexts
- DP15 provides verifiable receipts and auditability
- DP17 ensures commerce contributes to sustainable infrastructure
- DP20 defines how surplus and value flows are owned and governed

Two additional couplings are load-bearing for the community economy described above.

DP1 is what makes agent-mediated and community commerce accountable. Anti-replay and sybil resistance matter most where value moves: without them, the same identity collects twice from a treasury, a single actor manufactures the appearance of vendor diversity, and contribution accounting becomes a payout exploit rather than a record.

DP5 is what makes economic objects addressable. Tradeable meta-assets, community identifiers that survive migration, and artifact identifiers that carry provenance across transfer are namespace properties on which settlement integrity depends. A transfer receipt is only as meaningful as the identifier it references.

DP6 is where these properties converge into real economic behavior.

12. Non-Goals and Explicit Boundaries

DP6 does not:

- mandate a single currency, ledger, or payment rail
- eliminate all forms of advertising (it requires honesty, bounds, and contestability)
- replace financial regulation or tax law
- guarantee equal economic outcomes

DP6 also does not:

- require that communities operate an economy; economic surfaces are a capability, not an obligation
- entitle a community to capture value from activity it neither hosts nor supports, since proportionality to contribution is a condition of legitimate claims
- prohibit profit, intermediation, or paid promotion; it requires that they be disclosed, bounded, and contestable
- guarantee that a transaction will be cheap, only that its full cost will be knowable before commitment
- resolve tax, licensing, or consumer-protection obligations, which remain matters for competent jurisdictions

Failure mode: **fairness theater**, where full disclosure is provided in a form no participant can act on, and the appearance of legibility substitutes for the ability to choose differently.

DP6 defines the conditions under which exchange is legitimate and non-coercive.

13. Minimum DP6 Alignment

A DP6-aligned commerce system should, at minimum:

- present itemized pricing and full fee breakdowns before commitment
- expose data uses tied to transactions and allow revocation of unrelated scopes
- label AI involvement in pricing, ranking, or recommendation
- provide cancellation and refund pathways with parity to signup friction
- generate verifiable, machine-readable receipts with responsible parties (DP15)

- bind transactions to enforceable policy and dispute mechanisms (DP12, DP13)
- ensure pricing, fees, and policy bindings persist or explicitly degrade across systems
- maintain receipt and transaction continuity across tools (DP7)
- prevent hidden fee introduction or policy resets during cross-system flows

Where community economic surfaces are operated, a DP6-aligned system should further:

- enumerate community surcharges as distinct, attributable line items at the point of commitment
- link surcharge revenue to what it maintains, with settlement proofs participants can inspect
- govern treasury allocation through a visible, revocable, tiered process (DP3)
- support exit from a community economy with receipts, history, and supported balances intact
- expose aggregate economic outcomes for audit without revealing individual transaction histories (DP4)

Partial compliance that omits execution, auditability, or exit should not be treated as alignment.

14. Open Questions and Future Work

Key open questions include:

- how to reconcile cross-border commerce with local community rules and norms
- how to provide stable units of account without sacrificing accessibility or neutrality
- how to standardize receipt portability across wallets and platforms (DP7)
- how to provide meaningful transparency in ranking without enabling gaming

- how to fund public goods through commerce without creating new forms of extraction
- how to assign liability when AI agents mediate transactions (DP11–DP13)

Community economic surfaces raise a further set of unresolved questions:

- how to bound cumulative surcharge burden when a participant transacts across many zones, each with a defensible individual claim
- how to measure a community's economic contribution well enough to ground proportional claims without turning contribution accounting into a gameable scoring system
- how to settle obligations between community economies without a clearing authority and without collapsing to a single currency
- how to prevent contribution accounting from becoming a surveillance record of participants' community labor (tension with DP4)
- what recourse exists when a community economy is captured, given that exit costs include social relationships rather than only economic ones

A further question is how automated markets should be evaluated for their effects on human participation and access to value, alongside efficiency. Game-theoretic analysis can help examine whether stable outcomes sustain livelihoods and community contribution, but no particular equilibrium or distributional guarantee is assumed.

These questions sit at the intersection of economic design, governance, and law.

15. Path Toward ML-RFC

Advancing DP6 toward ML-RFC requires:

- standardizing transaction, fee, and receipt schemas
- publishing reference checkout patterns with full disclosure models
- piloting community-defined economic rules and surcharges
- aligning dispute evidence with provenance standards (DP15)
- collaborating with regulators, platforms, and civil society on interoperable approaches

Progress should be demonstrated through working systems, not only conceptual agreement.

16. Closing Orientation

DP6 is where the Meta-Layer proves whether it can support real economic life without reverting to extraction.

Fair commerce is not the opposite of innovation. It is the condition under which innovation remains trustworthy.

When DP6 is strong, participants can pay, earn, and fund commons with clarity, agency, and dignity.

When it is weak, every other property is undermined at the moment money changes hands.

Commerce that fails under interoperability pressure is not fair commerce. DP6 requires that fairness survive movement.

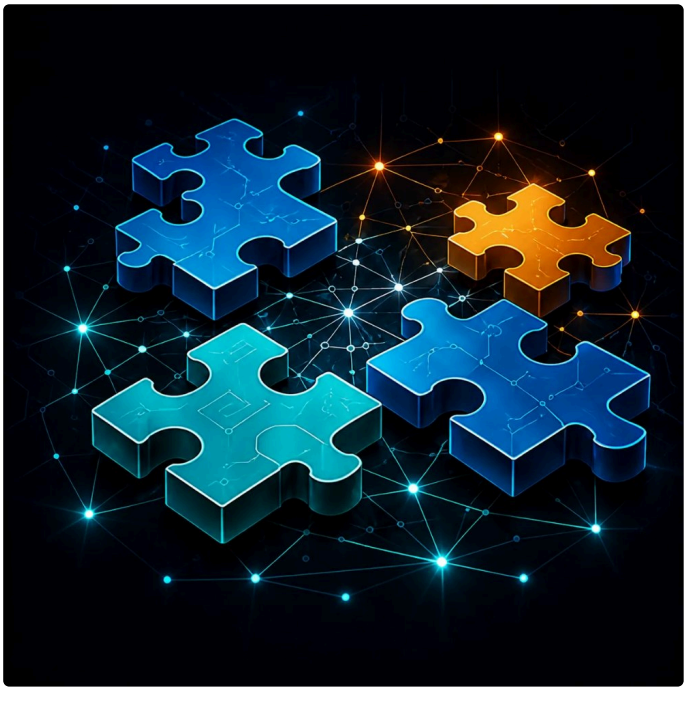
Continue the conversation on DP6 online →

DESIRABLE PROPERTY 7

DP7

Simplicity and Interoperability

The Meta-Layer is designed to reduce friction, not add it – prioritizing clarity, composability, and seamless interaction across domains.



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp07

1. Purpose of This Draft

This draft articulates Desirable Property 7 (DP7) as the condition under which participants, communities, and systems can move across tools, environments, and contexts without losing identity, history, value, or agency.

Interoperability in the Meta-Layer is not a feature of APIs. It is the condition under which power cannot be quietly re-centralized through infrastructure, interfaces, or economic gravity.

This includes the ability for new tools, services, and technologies to plug into the Meta-Layer through shared interfaces without prior permission, provided they conform to governance rules, security constraints, and boundary conditions.

DP7 ensures that governance (DP3), incentives (DP9), ownership (DP20), and commerce (DP6) remain continuous across boundaries rather than collapsing into platform-specific silos.

If DP7 is weak, predictable failures are not accidental but structural: re-centralization through convenience, lock-in through partial openness, stranded value, degraded identity, and governance that cannot extend beyond a single interface.

DP7 defines the minimum conditions under which the Meta-Layer functions as a shared system rather than a collection of connected but incompatible domains.

Anchor Principle: Open to participation, bounded by policy. Any system may join, but no system may bypass the rules that preserve safety, trust, and collective integrity.

Interoperability is the primary boundary where power is contested in the Meta-Layer. Systems that appear interoperable locally but degrade meaning, enforceability, or usability across boundaries will systematically re-centralize control.

2. Problem Statement

In today's web, systems are technically connected but structurally discontinuous.

Participants encounter:

- identities that can be exported but lose reputation, trust, or verification context
- data that can be downloaded but cannot be meaningfully re-imported
- value that can move but becomes illiquid, discounted, or restricted
- governance rules that do not carry enforcement into new environments

This produces recurring failures:

- switching systems resets social and economic position
- contributions lose meaning outside their original context
- communities cannot extend norms or coordination across tools
- "open" ecosystems still trap participants through semantic or economic loss

These failures are structural. Interoperability is often reduced to data transfer rather than continuity of meaning, enforceability, legitimacy, and coordinated participation across systems.

DP7 reframes interoperability as continuity under movement and participation. Not just that things move, but that they remain valid, trusted, and usable after they move, and that new systems can enter and participate without breaking that continuity.

3. Threats and Failure Modes

3.1 Interoperability theater

Systems simulate openness while preserving control.

Example: Data export tools provide raw files without structure, signatures, or compatibility, making reuse impractical.

Why this matters: Nominal interoperability without usability reinforces lock-in.

3.2 Semantic collapse

Objects move but lose meaning.

Example: Reputation exports as a number but loses trust graph context, rendering it unusable.

Why this matters: Meaning, not data, determines continuity.

3.3 Governance discontinuity

Rules travel without enforcement.

Example: A community charter exports but cannot bind behavior in a new environment.

Why this matters: Governance without execution becomes symbolic.

3.4 Economic degradation

Value moves but loses utility or legitimacy.

Example: Assets transfer but cannot be used, traded, or redeemed equivalently.

Why this matters: Portability without usability is economic lock-in by another name.

3.5 Asymmetric interoperability

Systems favor inbound over outbound movement.

Example: Easy onboarding from other platforms, but restricted or degraded export.

Why this matters: Asymmetry preserves centralization under the appearance of openness.

3.6 Protocol and standards capture

Dominant actors influence interoperability standards.

Example: Standards evolve in ways that favor incumbent implementations.

Why this matters: Interoperability can become a mechanism of control rather than liberation.

3.7 Bridge and translation attack surfaces

Cross-system connectors introduce risk.

Example: Cross-system bridges become vectors for fraud, duplication, or corruption of assets and identity.

Why this matters: Interoperability expands the attack surface of the system.

3.8 Privacy leakage through portability

Data movement exposes unintended information.

Example: Exported data reveals relationships or behavior not intended for new contexts.

Why this matters: Continuity must not violate data sovereignty (DP4).

3.9 Re-centralization through aggregation layers

Intermediaries regain control over open systems.

Example: Wallets, indexes, or identity providers become gatekeepers of interoperability.

Why this matters: Interoperability without anti-capture design re-centralizes power.

3.10 Injection risk and extension ecosystem spam

Pluggable systems introduce code injection, privilege abuse, and discovery flooding risks.

Example: Malicious overlays request excessive permissions, spoof trusted interfaces, or exfiltrate data; low-quality or duplicate apps flood discovery surfaces to capture attention.

Why this matters: Openness without containment and fair discovery degrades safety and usability, turning interoperability into an attack surface.

3.11 Closed or permissioned extension ecosystems

Systems restrict which technologies can integrate despite claiming openness.

Example: Platforms provide SDKs but require approval, ranking control, or distribution gatekeeping that limits who can participate.

Why this matters: Interoperability without permissionless participation reintroduces centralized control through integration policy rather than infrastructure.

4. Core Principle

Interoperability in the Meta-Layer means that identity, data, value, governance, and participation can move across systems without losing integrity, meaning, enforceability, or legitimacy.

It is not sufficient for objects to move. They must remain:

- interpretable within new contexts
- verifiable against their origin
- enforceable under governance rules
- usable in economic and social systems

Example: A participant moves to a new environment and retains not only identity and history, but the ability to exercise governance rights, receive recognition for contributions, and use their assets without degradation.

What this feels like: Switching systems does not mean starting over, nor does it mean accepting a degraded version of prior participation.

Interoperability has two goals that should not be confused. Portability lets identity, data, reputation, and value move from one system to another. Interaction lets systems work with one another directly while nothing moves: an agent keeps its own identity and still discovers and uses agents or services on other platforms. DP7 requires both.

5. Primary Mechanisms and Structural Conditions

5.1 Open schemas and standards

Schemas are not neutral technical artifacts. They define what can be expressed, preserved, and validated across systems.

DP7 requires schemas to be publicly defined, evolvable, and resistant to capture. Systems must be able to extend schemas without breaking compatibility or consolidating control.

A failure mode is schema capture, where standards evolve to favor specific implementations, creating hidden lock-in despite nominal openness.

A minimal interoperability baseline keeps simplicity in view. Shared interfaces build on HTTP, existing web infrastructure, and established standards, with core capabilities separated from optional extensions. A practical test of simplicity is how much additional work two independently built systems need before they can interoperate at the baseline: the less, the better.

Standards-setting should include meaningful Global South participation in the decisions that create technical dependencies. Plural needs should shape the specification from the outset, rather than be treated only as localization after corporate defaults have been settled.

5.2 Version negotiation

As systems evolve, differences in schema versions and capabilities are inevitable. Interoperability must account for this without breaking continuity.

DP7 requires explicit version negotiation mechanisms that allow systems to detect compatibility, fall back gracefully, and signal limitations.

A failure mode is silent incompatibility, where objects appear valid but behave incorrectly due to version mismatch.

Agents add a new form of negotiation. Two agents can exchange capability descriptions and agree, at the moment of interaction, which protocol, interface, data format, and security requirements to use, whether a structured API, message exchange, or another method both support, within the security and authorization boundaries each side sets. For example, a personal agent discovers a service run by another organization, reads its capability description, verifies its identity, obtains the authorization it needs, agrees how to communicate, and interacts with it directly, without a shared central platform.

5.3 Data minimization in portability

Not all data should move across systems. Interoperability must balance continuity with privacy and relevance.

DP7 requires that only necessary data be transferred, with explicit controls for redaction and minimization (DP4).

A failure mode is over-export, where unnecessary data is transferred, increasing exposure and risk without improving usability.

5.4 Portable reputation and credentials

Reputation and credentials are only meaningful within their context. Moving them without context creates false signals or exploitation opportunities.

DP7 requires that reputation objects include sufficient context such as source, method, and scope to remain interpretable.

A failure mode is reputation flattening, where complex trust signals are reduced to simple scores that can be gamed or misused.

5.5 Cross-system governance mapping

Governance rules must translate across systems with clear expectations of enforcement.

DP7 requires explicit mapping between policy objects and enforcement environments, including where equivalence holds and where it does not.

A failure mode is governance drift, where rules appear consistent but are enforced differently depending on the system.

5.6 Economic portability

Value must remain usable, not just transferable. Economic objects must retain legitimacy across systems.

DP7 requires that transferred value preserves its functional properties, including redeemability, liquidity, and constraints (DP6).

A failure mode is value degradation, where assets lose usability or trust when moved.

5.7 Identity continuity

Identity is the anchor for all other portable objects. Without continuity, interoperability collapses into impersonation or reset.

DP7 requires consistent, verifiable identity across systems, with protections against duplication and spoofing (DP1).

A failure mode is identity fragmentation, where participants appear as different entities across systems, losing continuity of rights and responsibilities.

Identity continuity is not trust, and trust is not authorization. Verifying that an agent is the same entity across services does not tell any of them what it may do. Each service decides authorization for itself, and permissions or governance policies granted in one system are never assumed to carry into another; they travel only as explicit, verifiable grants that the receiving system chooses to honor.

5.8 Anti-capture interop design

Interoperability systems must be designed to resist re-centralization through aggregation or coordination layers.

DP7 requires mechanisms that distribute control over indexes, relays, and discovery systems.

A failure mode is aggregation capture, where intermediaries become de facto gatekeepers of movement.

5.9 Privacy-preserving interoperability

Interoperability must not expose participants to unintended data leakage or correlation.

DP7 requires privacy-preserving mechanisms such as selective disclosure, anonymization, and context-aware data handling (DP4).

A failure mode is linkage attack, where cross-system data enables reconstruction of sensitive information.

5.10 Graceful degradation

Not all systems will fully support all features. Interoperability must handle partial compatibility explicitly.

DP7 requires systems to preserve core meaning and signal any degradation in functionality, guarantees, or enforcement.

A failure mode is silent degradation, where participants believe continuity exists but critical properties have been lost.

6. Governance, Accountability, and Agency Surfaces

Interoperability is not neutral infrastructure. It encodes decisions about what persists, what degrades, and who controls movement.

Without explicit governance surfaces, interoperability becomes a mechanism of extraction or control, where participants can move data but not meaning, value, or rights.

DP7 requires that cross-system movement be visible, contestable, and governed at the moment of transfer.

Participants must be able to:

- initiate export/import of core objects (identity, credentials, receipts, balances) with clear previews of what will change
- see how objects will be transformed (schema mapping, lossiness, policy constraints) before committing
- control permissions during transfer, including redaction, minimization, and revocation (DP4, DP2)
- verify integrity of received objects (signatures, lineage, timestamps) and reject degraded or untrusted imports (DP15)
- access dispute pathways when a transfer results in loss, misrepresentation, or policy violation

Communities must be able to:

- define interoperability policies for their zones (what can enter/leave, required attestations, risk tiers)
- attach executable constraints to imports (e.g., quarantine, limited privileges, probation periods) (DP12, DP13)
- audit aggregate interop flows for capture, leakage, and harm without exposing individual participants
- evolve interop rules with memory of incidents, reversals, and outcomes (DP12 governance loops)

Without these surfaces, interoperability becomes performative: objects move, but participants lose control and systems quietly re-centralize.

Example: A community allows import of reputation from another network only with attested receipts and places imported identities in a probation state with reduced privileges until local activity establishes trust.

7. Incentives and Power Analysis

Interoperability is where platform power is defended or broken.

Dominant actors tend to support inbound interoperability (ingest users/data) while restricting outbound portability (export of value, reputation, and history). This creates asymmetric openness that preserves control.

DP7 requires incentive legibility around interop:

- how APIs, SDKs, and terms of service shape inbound vs outbound flows
- how ranking, discovery, and monetization favor native over imported content
- how fees or friction are applied to export vs import paths

Common patterns to surface and constrain:

- **Fake export:** Data can be downloaded but loses structure, signatures, or context needed for reuse.
- **Rate-limited exit:** Export is throttled or degraded compared to import pathways.
- **Semantic loss:** Reputation or credentials export without the trust graph or scoring model that gives them meaning.
- **Bridge tolls:** Cross-system transfers incur hidden fees or spreads that discourage movement.
- **Aggregator capture:** Indexing or wallet layers re-centralize control over "open" ecosystems.

DP7 therefore expects:

- symmetric capabilities for import and export where feasible
- disclosure of material asymmetries and fees

- community ability to demote or penalize systems that degrade outbound portability
- fair discovery and anti-spam mechanisms for pluggable tools (rate limits, probation states, reputation-weighted surfacing)
- transparent policies for SDKs and extensions that bind incentives to compliant behavior (DP9)

When incentives favor staying, ecosystems centralize. When incentives favor honest movement, ecosystems compose.

8. Community Signals Informing DP7

Across ecosystems, recurring signals indicate interop failure at scale:

- frustration with "export" tools that do not preserve meaning or usability
- loss of reputation or trust when moving between platforms
- inability to carry subscriptions, balances, or purchase history across tools
- skepticism toward "open ecosystems" that still gate exit
- concern about data exposure and identity spoofing during transfers

These are not UX issues. They are structural breaks in continuity.

DP7 treats these signals as requirements for preserving meaning, not just moving bytes.

9. Evaluation Criteria

DP7 alignment cannot be assessed by inspecting an API surface. It must be evaluated by moving real objects across real boundaries and measuring what survives.

The following criteria are diagnostic rather than exhaustive. Each is written so that a negative answer identifies a specific structural defect rather than a general complaint.

9.1 Continuity of meaning

- After transfer, can objects still be interpreted without access to the origin system?

- Do identity, reputation, and credential objects retain the context that made them meaningful (source, method, scope)?
- Can a receiving system reconstruct lineage across multiple hops?

9.2 Symmetry of movement

- Is export as capable, fast, and complete as import?
- Are fees, rate limits, and friction comparable in both directions?
- Are material asymmetries disclosed at the point of decision rather than in terms of service?

9.3 Enforceability after transfer

- Do governance rules continue to bind behavior in the receiving environment, or do they degrade to advisory text?
- Are precedence rules declared when the origin and destination policies conflict?
- Can a community refuse or quarantine an import, or put it on probation, under its own policy?

9.4 Integrity and verifiability

- Are signatures, authorship, and timestamps preserved and checkable?
- Are broken lineage chains flagged as risk rather than treated as benign gaps?
- Can a participant reject a degraded import without losing the rest of their history?

9.5 Disclosure of loss

- Does the system state, before the transfer commits, what meaning will be lost and what guarantees will no longer apply?
- Is degradation signaled in the interface at decision time, not only in logs?
- Are transfers blocked or gated when degradation cannot be described?

9.6 Auditability

- Does every cross-system transfer produce a receipt naming what moved, how it was transformed, and who mediated it?

- Can aggregate interop flows be audited for capture and leakage without exposing individual participants?
- Are disputes, reversals, and postmortems linked back to the schema, policy, or bridge change that caused them?

9.7 Anti-capture posture

- Are indexes, relays, registries, and discovery surfaces controlled by more than one party?
- Can a competing implementation pass a public conformance suite without privileged access?
- Do discovery rankings resist paid or incumbent advantage for pluggable tools?

9.8 Participant experience of exit

- Can a non-technical participant leave with their identity, history, balances, and rights intact?
- How long does a complete exit take, and what is measurably lost?
- Is the exit path tested regularly, or only claimed?

Why this matters: Interoperability claims are cheap. These criteria convert the claim into an observable property of a specific transfer between two specific systems.

10. Implementation Patterns

These patterns translate DP7 into design moves that can be adopted incrementally. None of them require a global standards body, and each can be evaluated against the criteria above.

10.1 Public conformance suites

Publish versioned, executable test suites for every shared interface. Any implementation can run them, and results can be published as attestations. Conformance becomes evidence rather than assertion.

10.2 Object envelopes with declared intent

Wrap every portable object in an envelope carrying schema version, intended use, trust assumptions, consent scope, and signature. Receiving systems enforce constraints from the envelope rather than inferring them, so that purpose binding survives the crossing rather than resetting at it (DP4, DP12).

10.3 Lossiness manifests

Ship every mapping with a machine-readable statement of what is preserved, transformed, and dropped. Interfaces render the manifest as a plain-language preview before a transfer commits.

10.4 Receipt logs for transfers

Emit a signed receipt for every import and export, linked to the policy version applied and the mediating party (DP15). Receipts are queryable by the participant, the origin community, and the destination community.

Audit is proportional to risk. A receipt records what is needed to verify the transfer and no more, and access to it is limited to the parties named. Every transfer still produces a receipt, but for low-risk interactions the audit record may be aggregate or privacy-preserving, so that auditability does not become a way to correlate a participant's activity across platforms (DP4).

10.5 Probation tiers for new integrations

Admit third-party tools at reduced privilege and visibility, with containment tiers and rate limits enforcing the reduction rather than a review promise (DP13). Increase capability as audits pass and non-abusive usage accumulates. Demote automatically on policy violation, with a public receipt explaining why.

10.6 Bridge attestation and risk tiers

Treat bridges and translators as accountable actors with published operators, audit history, and risk tiers, bound to a responsible entity rather than operating as anonymous infrastructure (DP1). Apply circuit breakers, volume caps, and anomaly detection proportional to tier.

10.7 Dual-write migration windows

During schema evolution, write both old and new representations for a bounded period, with explicit deprecation dates and migration guidance. Historical continuity is preserved rather than reconstructed later.

10.8 Scheduled exit drills

Periodically perform and publish a full export-and-reimport of a representative account into an independent implementation. Report duration, failures, and losses. Untested exit paths are assumed broken.

10.9 Federated discovery with reputation weighting

Distribute indexing and discovery across multiple operators, weight surfacing by reputation and audit status, and rate-limit new entrants to resist flooding without gatekeeping participation. Discovery is an incentive surface, and concentration of it reproduces platform leverage under another name (DP9).

10.10 Degradation-aware interfaces

Standardize visual and textual signals for reduced guarantees, so participants encounter the same vocabulary of degradation across tools rather than per-platform euphemisms.

10.11 Shared overlay trays and navigation panels

An open tray interface can let assistants, wallets, chat tools, and other overlays coexist without competing for the same page space. Registration should declare a tool's name, role, permissions, and availability; claimed priority should not override the participant's choices about display, pinning, expansion, or ordering. Placement should remain adaptable across browsers, devices, and assistive technologies.

Navigator User Interfaces provide a related pattern for shareable, remixable task and learning flows through a regular panel. Shared interface contracts should let plural tools and community-authored workflows coexist without imposing one navigation scheme. DP2 defines the participant's control over those choices.

10.12 Familiar storage and privacy choices

Familiar save and export interfaces should offer participant-selected storage destinations, including decentralized options such as IPFS. The interface should explain the chosen custody, retention, and retrieval arrangements; selecting a content-addressed destination does not by itself establish confidentiality or durable availability.

Privacy transport should be similarly usable across supported protocols. An interoperable interface should let participants retain VPNs or anonymity layers they already trust, with minimal configuration and explicit compatibility limits. A native routing option should not make external alternatives second-class or imply that routing alone keeps all data local and untracked.

11. Foresight and Failure Design

Interoperability systems must assume adversarial pressure, economic incentives for capture, and rapid evolution of tools and agents.

Failures rarely occur as isolated events. They emerge as gradual degradation of meaning, enforceability, and trust across system boundaries.

Common failure paths include:

- protocol capture, where standards evolve to favor dominant actors
- bridge exploits, where cross-system connectors become attack vectors
- impersonation and replay of identity or credential artifacts
- economic friction that discourages portability despite nominal support
- aggregation capture, where intermediaries regain control over movement

These failures compound over time. For example, minor semantic loss in early transfers reduces trust, which leads to reduced usage, which increases reliance on centralized intermediaries, accelerating re-centralization.

Cross-system environments introduce dynamic risks:

- divergence of rule interpretation across systems
- inconsistent enforcement depending on context

- loss of lineage or provenance during repeated transformations

DP7 therefore requires proactive safeguards:

- risk tiers and circuit breakers for high-risk interoperability pathways
- attestation and reputation systems for issuers and bridges
- anomaly detection for unusual transfer patterns (DP13)
- public postmortems linking failures to schema, policy, or bridge changes

Interoperability must detect not only discrete failures, but slow drift toward loss of meaning and control.

Failure is expected. Silent or irreversible failure is not.

12. Interoperability System Layer: Continuity, Translation, and Power

Interoperability requires more than shared formats. It requires a layer that preserves meaning, authority, and usability across boundaries where systems may have conflicting incentives.

12.1 Portable objects

All core system elements must be representable as portable, structured objects:

- identity objects (DP1)
- policy objects (DP12)
- incentive objects (DP9)
- ownership objects (DP20)
- transaction objects (DP6)

These objects must carry sufficient context to remain interpretable outside their origin.

They should also declare their intended use and trust assumptions so receiving systems can enforce appropriate constraints. Without declared intent, objects may be misapplied in contexts that invalidate their meaning.

12.2 Semantic translation and mapping

Systems must define how objects are interpreted across contexts:

- schema translation
- semantic alignment
- version compatibility

Mappings must explicitly declare where information is preserved, transformed, or lost.

They should include machine-readable diffs of meaning so downstream systems can reason about equivalence. Absent explicit mapping, silent reinterpretation becomes a primary vector for drift and exploitation.

12.3 Integrity and lineage preservation

Objects must retain:

- authorship
- timestamps
- signatures
- lineage and derivation history

Without this, imported objects cannot be trusted or verified.

Lineage should be queryable across hops to reconstruct full transformation chains. Breaks in lineage must be flagged as risk, not treated as benign gaps.

12.4 Permission and consent continuity

Access controls and consent conditions must persist across systems.

Participants must not lose control over their data or identity during transfer (DP4, DP2).

Consent scopes should be renegotiable at boundaries with clear previews of changes. Implicit expansion of scope during transfer must be disallowed or explicitly surfaced.

12.5 Interoperability receipts

All cross-system transfers generate verifiable records:

- what moved
- how it was transformed
- what constraints applied
- who mediated the transfer

These receipts enable audit and dispute resolution (DP15).

Receipts should be machine-verifiable and linkable to governance and commerce records. Missing or partial receipts must downgrade trust for the resulting state.

12.6 Conflict resolution under power asymmetry

Systems must define how conflicts are resolved when:

- governance rules differ
- economic conditions diverge
- trust models are incompatible

Conflict resolution is not neutral. It must be visible, contestable, and governed.

Resolution pathways should declare precedence rules and appeal mechanisms. Opaque arbitration at boundaries is a primary route to capture.

12.7 Loss and degradation signaling

All interoperability pathways must explicitly signal:

- what meaning is lost
- what functionality is degraded
- what guarantees no longer apply

This prevents silent failure of continuity.

Signals should be standardized and user-visible at decision time, not buried in logs. Systems that cannot signal degradation must restrict the transfer or require explicit override.

12.8 Interoperability memory

All transfers and mappings persist as a linked history:

- prior versions
- transformation paths
- disputes and reversals

This creates a system-level memory of how interoperability evolves over time.

Memory should support querying for systemic patterns such as drift or repeated loss. Without analysis over memory, issues recur undetected.

12.9 Composable participation via governed interfaces

The Meta-Layer must allow third-party tools, services, and extensions (e.g., smart tags, overlays, sidebars, core services) to plug into shared interfaces **without prior permission**, provided they conform to declared interfaces and governance constraints.

Conforming integrations must:

- declare permissions and data scopes up front
- bind to zone policies at runtime (DP12)
- operate within containment tiers and rate limits (DP13)
- provide signed artifacts and provenance (DP15)
- expose auditable behavior and event logs

Non-conforming integrations must be sandboxed, rate-limited, or blocked.

This enables openness to innovation while preventing unbounded execution and platform capture.

Interface contracts should be versioned and testable, with conformance suites available publicly. Discovery systems must incorporate reputation and probation to resist spam and gaming.

Example (Composable Integration): A third-party sidebar app plugs into a community zone. At install, it declares permissions (read annotations, write highlights) and data scopes. The zone's policy automatically con-

ied, and actions are rate-limited. The app runs in a sandbox and emits signed event logs. Initially, it appears in a probation tier with limited visibility. As it accumulates positive, non-abusive usage and passes audits, its privileges and discoverability increase. If it violates policy, it is throttled or quarantined with a public receipt explaining why.

12.9.1 Agent interaction contracts

Interfaces should support visible, responsible, and semantically coherent interaction among independently operated agents, websites, and other services. A minimal interaction contract should specify exchange formats, observable traces, and distributed authentication, with the permissions and governance constraints already required here.

The scope includes non-page media, agents without a shared central host, and systems in physical or hybrid settings. Interoperability does not certify a physical system's safety; identity, authority, and containment must remain legible across the boundary.

13. Relationship to Other Desirable Properties

DP7 is the continuity layer across the Meta-Layer stack:

- DP3 ensures governance can evolve across systems rather than reset per platform
- DP4 constrains data movement and enforces minimization during transfer
- DP6 ensures commerce artifacts (transactions, balances, receipts) remain usable across contexts
- DP9 carries incentive history and attribution across tools without silo lock-in
- DP12 enables policy translation and enforcement across environments
- DP13 contains risks introduced by cross-system automation and agents
- DP15 provides verifiable provenance for imported objects
- DP17 sustains shared infrastructure required for interoperability
- DP20 preserves ownership and fork rights across environments

DP7 binds these properties into a coherent, cross-system reality.

14. Non-Goals and Explicit Boundaries

DP7 does not:

- require full compatibility between all systems or schemas
- mandate a single global standard or governing body
- eliminate competition among platforms or protocols
- force communities to accept all imports without policy

DP7 defines the conditions under which movement is legitimate, intelligible, and safe.

15. Minimum DP7 Alignment

A DP7-aligned system should, at minimum:

- support export of core objects with preserved structure, signatures, and lineage
- provide import pathways with explicit mapping, lossiness disclosure, and policy constraints
- maintain permission and consent continuity during transfers (DP2, DP4)
- generate interoperability receipts for all transfers (who, what, when, how transformed)
- offer dispute and rollback pathways for harmful or incorrect imports
- avoid material asymmetry between import and export without disclosure

Partial compliance that omits integrity, consent, or auditability should not be treated as alignment.

16. Open Questions and Future Work

Key open questions include:

- how to standardize semantics (reputation, credentials, governance roles) without flattening local meaning
- how to achieve Sybil resistance and anti-impersonation across interoperable identity systems (DP1)

- how to reconcile regulatory boundaries with cross-system value portability (DP6)
- how to design bridges that are both secure and usable without becoming choke points
- how to compensate shared infrastructure (indexes, relays) without enabling capture
- how to evolve schemas without breaking historical continuity (versioning and migration)

These questions sit at the boundary of protocol design, governance, and law.

17. Path Toward ML-RFC

Advancing DP7 toward ML-RFC requires:

- standardizing core object schemas (identity, policy, incentive, ownership, transaction)
- defining interoperability receipt formats and audit events
- building reference bridges and adapters with open security reviews
- piloting cross-system governance and commerce scenarios with real communities
- aligning with regulators on portability, custody, and liability boundaries

Progress should be demonstrated through live interop scenarios, not only specifications.

18. Closing Orientation

DP7 is the condition under which the Meta-Layer remains a network rather than reverting to a set of silos.

When interoperability is real, participants can move without losing meaning, value, or rights.

When it is simulated, ecosystems re-centralize behind APIs, bridges, and aggregators that quietly control movement.

Interoperability is not a feature. It is the boundary where power either remains distributed or collapses back into platforms.

Continue the conversation on DP7 online →

DESIRABLE PROPERTY 8

DP8

Collaborative Environment and Meta-Communities

The Meta-Layer supports real-time collaboration that travels across the web – so your people are always close.



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp08

1. Purpose of This Draft

This draft articulates Desirable Property 8 (DP8) as the condition under which communities can **define, enforce, and evolve participation and governance at the interface layer of the Meta-Layer**.

DP8 establishes that governance is not inherited from platforms, but constructed by communities operating within zones. It defines how participation, influence, and intelligence are structured so that trust remains contextual, enforceable, and resistant to manipulation.

A meta-community is a network of communities, people, and/or their agents that operates above webpage content. It may bring together individuals, neighborhood groups, professional associations, universities, public agencies, social movements, cultural organizations, and informal networks that remain distinct but work together through shared purpose, reciprocal exchange, and relationships that carry trust across boundaries. A collaborative environment is the set of conditions in which they can contribute, disagree, learn, and coordinate without having to become identical, and in which differences of scale, speed, and practice are treated as capacity rather than obstacles.

DP8 is not moderation. It is the **system-level design of environments in which interaction occurs**.

In plain terms: a community is any group of people who choose to share rules for how they take part together, whether a neighborhood group, a professional network, an open-source project, or a coalition of such groups. A zone is the space where those rules apply, on one site or across many. Communities write the rules for their zones; DP8 is about making those rules hold, and making it clear who enforces them.

2. Problem Statement

On today's web, participation and governance are platform-defined:

- rules are globally applied regardless of context
- moderation is opaque and centralized
- influence is driven by engagement dynamics, not trust
- bots and AI distort visibility and reputation

- communities cannot enforce their own norms at the system level

This leads to predictable failures:

- manipulation through brigading and synthetic amplification
- governance capture by small groups or opaque systems
- lack of trust in moderation and rule enforcement
- inability to adapt governance to context and risk

DP8 addresses this by enabling communities to define **zone-specific governance systems** that operate at the interface layer and persist across the web.

3. Threats and Failure Modes

DP8 assumes adversaries will combine **identity (DP1), agency (DP2), data flows (DP4), governance (DP8), and incentives (DP9)**. Systems MUST be robust to **multi-vector, cross-zone attacks** and degrade safely.

3.1 Security and Adversarial Failure Modes (Index)

DP8 names its failure modes inline throughout the sections that follow, because each mechanism fails in a specific way. The list below indexes that vocabulary so it can be tested, monitored, and referenced as a set.

Governance that does not bind

- **phantom governance:** rules exist but do not change visibility, amplification, or access
- **declarative governance:** governance artifacts are published but not machine-enforceable
- **governance stripping:** constraints are lost when content or participants cross a boundary
- **module bypass:** a governance module influences outcomes without passing enforcement checks

Context and continuity failures

- **context collapse:** rules carry silently across zones that do not share norms or risk

- **governance fragmentation**: state does not persist across pages, sessions, or systems
- **zone conflict ambiguity**: overlapping rules resolve non-deterministically
- **interop deception**: transfers claim preservation of governance that did not survive
- **semantic mismatch**: signals are reinterpreted incorrectly across implementations

Participation and influence failures

- **tier gaming** and **fast-track escalation**: influence accrues without verifiable contribution
- **privilege ossification**: roles never decay, entrenching early participants
- **amplification spoofing**: automation or sybil identities drive high-impact reach
- **reputation laundering**: signals are reshaped across contexts to manufacture trust
- **cross-zone escalation**: standing in one zone confers illegitimate power in another
- **throughput abuse**: volume substitutes for trust

Automation failures

- **AI governance bypass**: agents act outside zone-defined constraints
- **identity masking** and **attribution gaps**: AI is indistinguishable from humans, or unattributable
- **scope creep** and **irrevocable delegation**: delegated authority expands or cannot be withdrawn
- **automation overrun**: agent actions are neither interruptible nor reversible
- **consent bypass via pipelines**: data or inference constraints are lost between stages

Systemic failures

- **governance capture**: coordinated influence or opaque concentration of decision power
- **brigading**: coordinated surges that overwhelm legitimate participation

- **composed attack success**: individually mitigated vectors succeed in combination
- **perverse incentives**: abuse is profitable, funding further attacks
- **fail-open amplification**: under load or uncertainty, systems default to permissive behavior
- **governance opacity** and **silent rule drift**: decisions and changes cannot be reconstructed
- **governance rigidity**: communities cannot evolve or fork their governance model

Why this matters: These are not incident categories to be triaged after the fact. Each names a condition that can be monitored continuously, and each has a corresponding structural condition elsewhere in DP8.

3.2 Threat Classes (Extended)

- **Sybil Attacks**: many identities controlled by few actors
- **Brigading**: coordinated surges to influence outcomes
- **Governance Capture**: concentration of power via roles or opaque processes
- **Reputation Laundering**: reshaping signals across contexts to gain undue trust
- **AI Amplification**: automated agents scaling influence beyond constraints
- **Cross-Zone Escalation**: importing status or signals to bypass local rules

3.3 Composed (Multi-Vector) Attacks

Adversaries may combine:

- AI agents + human click-farms
- identity cycling + cross-zone escalation
- incentive exploits (rewards) + feedback loops
- data laundering (DP4) + reputation reuse (DP8)

Systems MUST detect **correlated anomalies** across time, topology, and identity linkages.

Failure mode: **composed attack success**, where individually mitigated vectors succeed in combination.

3.4 Detection Signals and Telemetry

- **Temporal**: burstiness, synchronized actions, unusual cadence
- **Topological**: tightly clustered interactions, graph anomalies
- **Behavioral**: repetitive patterns, low-entropy content, abnormal conversion rates
- **Cross-Context**: sudden tier jumps across zones, inconsistent identities

Systems SHOULD fuse signals into risk scores with **explainable summaries.**

3.5 Response Playbooks

- **Progressive friction**: rate limits, proof escalation, cooldowns
- **Containment**: quarantine zones, shadow reduction of amplification
- **Rollback**: revert affected rankings or decisions where feasible
- **Human review**: escalate high-impact cases with auditable decisions (DP1 linkage)

Failure mode: **delayed or blunt response** causing collateral damage or missed containment.

3.6 Transparency vs. Gaming

- Provide participant-legible explanations and audit summaries
- Protect sensitive thresholds and heuristics

Failure modes:

- **gaming via overexposure**
- **opacity via underexposure**

3.7 Cross-Zone Containment and Signal Sharing

- Attacks are **zone-scoped by default**; sharing of sanctions/signals MUST be deliberate and thresholded
- Systems SHOULD support **signed, scoped advisories** between zones

Failure modes:

- **cascading harm** (over-sharing) or **blindness** (under-sharing)

3.8 Incentive Alignment (DP9 Link)

- Systems MUST minimize rewards for abusive behavior (no easy profit from spam/brigades)
- Rewards SHOULD be tied to **verified, sustained contribution**

Failure mode: **perverse incentives** that fund attacks.

3.9 Resilience and Safe Degradation

- Under uncertainty, systems SHOULD degrade to **safer defaults** (reduced amplification, higher proof requirements)
- Maintain service continuity while limiting harm

Failure mode: **fail-open amplification** under stress.

4. Core Principle

Communities must be able to define and enforce the conditions under which participation, influence, and intelligence operate. If governance cannot be enforced under scale, coordination, and adversarial pressure, trust collapses.

DP8 treats governance as an **interface-level control system** coupled to **identity (DP1)**, **agency (DP2)**, **data flows (DP4)**, and **AI governance (DP12)**.

It has three inseparable properties:

- **Outcome control (DP2)**: rules must change what actually happens (visibility, amplification, access), not merely configure preferences.
- **Continuity (DP1, DP4)**: governance must persist across pages, sessions, and interoperating systems, with explicit signaling on degradation.
- **Contestability (DP1)**: decisions must be attributable, auditable, and reversible within bounded processes.

Implications:

- Governance is **executed at the point of interaction** (overlays), not deferred to platform backends.
- High-impact actions (virality, reputation shifts, moderation) are **gated by proofs** appropriate to the zone (e.g., unique human verification per DP1,

quorum, role authority per DP8).

- Automation (AI) is **subordinate to zone policy** with explicit scope, attribution, and revocation (DP12), and must honor data purpose and consent propagation (DP4).

Failure conditions (non-exhaustive):

- **Phantom governance**: rules exist but do not alter outcomes (violates DP2).
- **Fail-open amplification**: under load or uncertainty, systems default to permissive amplification (violates DP8 + DP9 alignment).
- **Uncontestable decisions**: participants cannot audit or appeal governance actions (violates DP1).

5. Primary Mechanisms and Structural Conditions

DP8 differs structurally from most Desirable Properties: its mechanisms are specified as an architecture rather than a flat list of conditions. The sections that follow this one carry that specification in depth. This section states the structural conditions that must hold regardless of how the architecture is implemented, and points to where each is elaborated.

5.1 Enforcement must occur at the point of interaction

Governance rules bind before actions propagate, at the interface layer, not in a platform backend that reviews outcomes afterward. Overlays, extensions, and native integrations are the execution surface. Elaborated in **System Architecture** (overlay-based governance, Sections 10.1–10.2) and **Governance Composition** (composition constraints, Section 13.2).

5.2 Policy must be attached to context, not to platforms

Zones are the unit of governance: composable, overlapping, portable policy containers that declare participation thresholds, governance rules, AI permissions, and trust signals. Rules do not leak silently across zone boundaries, and boundary transitions signal changes in guarantees. Elaborated in **System Architecture** (Sections 10.3 and 10.4.1–10.4.4).

5.3 Capability must be tiered, stateful, and revocable

Participation maps to enforced tiers with entry conditions, verifiable progression, and decay. Capability cannot be acquired out of band, and high-impact actions require proofs proportional to their reach. Elaborated in **Participation Model**.

5.4 Automation must be a governed actor class

AI agents hold verifiable identity, bounded scope with expiry, disclosure at the interface, and revocation pathways. Agents are subordinate to zone policy at runtime, not to policy documents. Elaborated in **AI Governance (DP12 Link)**.

5.5 Governance must be composable without becoming bypassable

Voting, moderation, reputation, access control, and dispute resolution operate as modules with typed and scoped outputs, declared precedence, and bounded feedback cycles. Elaborated in **Governance Composition**.

5.6 Every governance action must produce reconstructable evidence

Attribution, authority, applied rules, outcome impact, and appeal traces are recorded so that decisions can be audited and contested. Elaborated in **Minimum DP8 Alignment** (Section 16.7) and the auditability requirements of **Path Toward ML-RFC** (Section 18.4).

5.7 Degradation must be safe and visible

Under load, uncertainty, or attack, systems move toward stricter defaults (reduced amplification, higher proof requirements, narrower scopes) and disclose that they have done so. Elaborated in **Core Principles (Normative and Enforceable)** (Section 9.9) and **Threats and Failure Modes**.

5.8 Communities must be able to evolve and fork

Governance stacks are versioned, forkable, and migratable, with explicit signaling to participants when rules change. Continuity of governance does not mean permanence of a single configuration. Elaborated in **System Architecture** (Section 10.4.8) and **Governance Composition** (Section 13.4).

Failure mode: **architecture without conditions**, where a system implements zones, overlays, and modules as features while leaving enforcement, continuity, attribution, or degradation unspecified.

6. Governance, Accountability, and Agency Surfaces

Governance in DP8 is the subject matter of the property, not an adjacent concern. What this section adds is the requirement that the governance system itself be governed: that participants and communities can see how rules were made, act on them, and hold their operators accountable.

A zone can enforce rules perfectly and still be illegitimate if the participants subject to those rules cannot inspect, influence, or leave them.

Enforcement is necessary but not sufficient for trust. Trust grows through repeated actions: commitments made visible and kept, credit given accurately, consent sought before stories or contact details are shared, and harm repaired when it occurs. A platform can enable contact and a policy can set expectations, but neither substitutes for the conduct through which people decide that collaboration is safe enough to continue. Trust is not a feature of a network; it is an ongoing practice within one, renewed as people join, roles change, and mistakes happen.

Participants must be able to:

- see which zone they are in, which rules apply, and which guarantees are currently active or degraded
- understand their own tier, how it was assigned, what it permits, and what would raise or lower it
- see when a governance action affected them, under what authority, and with what stated reason
- contest decisions through a defined appeal path with timelines and reachable outcomes
- distinguish human from automated actors and inspect the scope of any agent acting in the zone
- exit a zone without forfeiting identity, history, or standing held outside it (DP1, DP7)

Communities must be able to:

- author, version, and fork their governance configuration without vendor permission
- assign, bound, and revoke steward authority, including term limits and decay
- audit enforcement in aggregate for bias, over-escalation, and missed harm without exposing individuals
- publish precedence rules for overlapping zones and the arbitration path when they conflict
- suspend or roll back a rule that produced harm, with the rationale recorded in governance memory
- bind third-party tools and agents to zone policy as a condition of operating in the zone

Stewards must be accountable in the same terms as participants. Moderation, adjudication, and rule configuration are high-impact actions, and DP8 treats them as attributable governance events rather than administrative privileges.

Much of the work that keeps a community healthy is not adjudication at all. Facilitators and network weavers notice who is missing, make careful introductions, help a group tell a challenging question from a personal attack, and make it safe to say "I do not know," raise a concern, or disagree without humiliation or retaliation. This relational work is part of a zone's governance capacity and should be recognized and resourced alongside formal stewardship (DP9, DP17).

Example: A participant's comment is down-ranked in a civic zone. The interface shows which rule applied, which steward or module executed it, the evidence class cited, and the appeal window. The appeal enters a queue with a published service level, and the outcome is written to governance memory whether or not the appeal succeeds. A quarterly audit reports how often that rule was applied, to whom, and how often appeals reversed it.

Failure mode: **steward exceptionalism**, where enforcement is auditable for participants but discretionary and unlogged for those who hold authority.

7. Incentives and Power Analysis

Governance competes with incentives. Where the two diverge, incentives usually win, because they operate continuously while governance operates episodically.

DP8 therefore treats amplification, visibility, and reputation as economic surfaces, not neutral mechanics. Whoever controls what becomes visible controls what becomes true in practice, and that control has value that others will pay to acquire.

Predictable pressures on community-defined governance include:

- **attention arbitrage:** actors invest in tier progression or steward proximity because visibility is monetizable, not because they value the community
- **moderation cost shifting:** platforms externalize enforcement labor onto unpaid stewards while capturing the resulting engagement value (DP9, DP17)
- **steward capture:** small groups accumulate adjudication authority because turnover is unrewarded and continuity is required for competence
- **compliance theater as competitive advantage:** systems adopt the vocabulary of zones and tiers without enforcement, then compete against systems that pay the real cost of enforcing
- **engagement-weighted defaults:** ranking systems inherited from the host platform quietly override zone policy at the margin
- **sanction externalities:** zones export sanctions and advisories to shift enforcement cost outward, or withhold them to avoid reputational exposure

DP8 therefore expects:

- disclosure of the optimization targets operating inside a zone, including any inherited from the host environment
- the ability for communities to set hard gates that incentives cannot bypass (DP12), rather than weights that incentives can outbid
- compensation or resourcing pathways for stewardship labor, so that governance capacity does not depend on volunteer exhaustion (DP9,

DP17)

- rotation, decay, and shared authority as structural defenses against entrenchment rather than as norms of good behavior
- no material reward for abusive throughput, so that brigading and farming are unprofitable rather than merely prohibited

Why this matters: A zone whose rules are enforceable but whose incentives reward circumventing them will drift toward circumvention. Governance holds only where the cheapest path is also the compliant one.

Reciprocity is the relational counterpart to incentive alignment. In a healthy meta-community every partner can contribute, receive value, and name what support it needs, and time, recognition, resources, and opportunities do not flow in only one direction. Knowledge shared across communities should carry attribution and consent; not every story, practice, or relationship should become a freely extractable resource.

8. Community Signals Informing DP8

Community input across the Meta-Layer submission process, moderation research, and platform governance experience converges on a consistent set of signals. They are not requests for more moderation. They are reports that governance is not real where it matters.

Recurring signals include:

- moderation experienced as arbitrary because the rule, the actor, and the evidence are never shown together
- rules that apply visibly to ordinary participants and invisibly, if at all, to influential ones
- communities that build norms in one environment and lose them entirely when the conversation moves
- reputation that cannot be carried, or that arrives stripped of the context that made it meaningful
- suspicion that a substantial share of amplification is automated, with no way to verify or contest it
- exhaustion among the people who actually do moderation and facilitation work

- distrust of appeals processes that accept submissions but produce no observable outcome
- concern that AI participants already operate with effective autonomy while governance debates continue
- marginalized communities reporting that automated enforcement misreads their context and vocabulary as violation

These signals identify structural breaks rather than usability defects: enforcement without attribution, continuity without portability, participation without impact, and automation without constraint.

DP8 treats them as design requirements. A zone that cannot answer "which rule, applied by whom, on what evidence, and how do I contest it" has not implemented governance regardless of how many controls it exposes.

Being able to reach a zone is not the same as being included in it. A person may be technically able to join a conversation while lacking time, language access, safety, connectivity, or confidence that their contribution will be respected. Communities should ask not only who can find a space, but who feels invited, prepared, and safe enough to take part, and offer low-bandwidth, multilingual, asynchronous, and offline routes where appropriate.

9. Core Principles (Normative and Enforceable)

DP8 principles are **normative and enforceable**, and interlock with **DP1 (Identity)**, **DP2 (Agency)**, **DP4 (Data)**, and **DP12 (AI)**.

9.1 Self-Determination (Enforceable; DP2)

Communities MUST be able to define participation and governance rules that **bind execution**.

- Rules MUST be machine-enforceable at the interface layer.
- Governance artifacts MUST be versioned and attributable (DP1).

Failure mode: **declarative governance**.

9.2 Contextual Governance (Zone-Bounded; DP1, DP4)

Rules MUST adapt to domain, risk, and norms, and be **scoped to zones**.

- Systems MUST prevent silent carryover of rules across zones.
- Transitions MUST signal changes in guarantees (DP4).

Failure mode: **context collapse**.

9.3 Graduated Participation (Stateful; DP2)

Participation MUST be tiered with **stateful progression and decay**.

- Capabilities MUST map to tiers and be enforced.
- Progression requires **continuity of contribution**; decay prevents permanent lock-in.

Failure mode: **tier gaming / privilege ossification**.

9.4 Human-Centric Trust Anchoring (Proof-Gated; DP1)

High-impact actions SHOULD require **proofs tied to unique humans**.

- Amplification and governance votes MUST resist sybil and automation dominance.

AI-driven accounts are not independent actors. Each maps to an accountable human or organization (DP1), and accounts that share an operator count as one participant for amplification and governance votes, so automation cannot multiply a single operator's voice. Because attacks combine, defenses are tested against combined vectors, such as many accounts plus brigading plus generated content, not one vector at a time.

Failure mode: **amplification spoofing**.

9.5 Interoperability (Truthful and Bounded; DP1, DP4, DP7)

Communities MUST persist across platforms with **honest signaling of what is preserved or degraded**.

- Identity, agency, and governance state MUST travel or explicitly degrade.

Failure mode: **interop deception**.

9.6 AI Situatability (Runtime-Bound; DP12)

AI MUST operate within **zone-defined constraints** with attribution, scope, and revocation.

Failure mode: **AI governance bypass.**

9.7 Precedence and Conflict Resolution (Deterministic)

Overlapping rules MUST resolve deterministically.

- Systems MUST declare precedence models.

Failure mode: **zone conflict ambiguity**.

9.8 Auditability and Recourse (First-Class; DP1)

Governance actions MUST be reconstructable and contestable.

Failure mode: **governance opacity**.

9.9 Safe Degradation (Fail-Safe Defaults; DP2, DP4)

Under uncertainty or attack, systems SHOULD degrade to **safer defaults.**

Failure mode: **fail-open under stress.**

10. System Architecture

10.1 Overlay-Based Governance

Governance operates at the interface layer through overlays (browser extensions, native integrations, or overlay apps), not within platform silos.

10.2 Core Primitives

- Identity (DP1)
- Agency (DP2)
- Data (DP4)
- Zones
- Governance Modules

Shared decentralized storage can support collaborative editing and distributed knowledge bases under participant- or community-chosen custody. Storage, concurrent editing, permissions, and retention remain distinct responsibilities; the choice of storage topology does not settle them all.

10.3 Zone Model (DP1 Integration)

Zones are:

- policy containers attached to context
- composable and overlapping
- portable across the web

Each zone defines:

- participation thresholds
- governance rules
- AI permissions
- trust signals

10.3.1 Plural Discovery and Community Formation

Community-maintained, viewpoint-rich directories can support discovery outside a single ranking system or fixed canon. Participants should be able to grow communities around shared explorations and serendipitous encounters rather than compete only for placement in a central index. Listing identifiable members should remain voluntary; resource curation and member discovery are different functions. DP2, Section 7.1 covers the per-zone controls over visibility, linkage, and discoverability that safe member discovery depends on, building on the identity conditions in DP1.

10.4 Governance System Layer: Continuity, Enforcement, and Capture Resistance

Beyond participation models and governance modules, DP8 requires a coherent governance system layer that ensures community-defined rules remain **enforceable, portable, and resilient under scale and adversarial pressure**.

Governance is not simply declared. It must persist across contexts, resist manipulation, and remain legible and contestable over time.

10.4.1 Governance Continuity Across Zones

Governance rules must persist as participants move across:

- platforms

- pages
- applications
- overlapping zones

This requires:

- consistent enforcement across contexts
- signaling when guarantees change
- preservation of governance state

Failure mode: **governance fragmentation**

10.4.2 Enforcement at the Interface Layer

Governance must be enforced where interaction occurs.

Systems MUST ensure:

- rules apply before actions propagate
- violations are constrained in real time
- enforcement is visible and explainable

Failure mode: **phantom governance**

10.4.3 Cross-Zone Conflict Resolution

Systems MUST define:

- precedence models
- conflict signaling
- fallback states

Failure mode: **zone conflict ambiguity**

10.4.4 Governance Propagation

Rules must propagate with content, participants, and interactions.

Failure mode: **governance stripping**

10.4.5 Capture Resistance

Systems MUST mitigate:

- coordinated influence attacks
- role entrenchment

- opaque decision-making

Failure mode: **governance capture**

10.4.6 Anti-Brigading

Systems MUST detect and limit coordinated behavior.

Failure mode: **brigading**

10.4.7 Governance Memory and Auditability

Governance decisions MUST be reconstructable and contestable.

Failure mode: **governance opacity**

10.4.8 Governance Evolution and Forkability

Communities MUST be able to evolve and fork governance models.

Failure mode: **governance rigidity**

10.5 Civic Overlays Across Platforms

A composable civic overlay can attach a shared topic to a page, forum thread, or social post and carry translated summaries, verifiable endorsements, deadlines, and decision status alongside the relevant material. Supported actions may include signing, donating, commenting, or voting, with the appropriate authorization and participant-selected wallet or decentralized identifier.

Youth groups, NGOs, municipalities, and DAOs can use the pattern to coordinate across existing platforms, with modular AI assistance and transparent campaign processes. A shared tag does not establish common authority or endorsement. Supported surfaces and integration limits should be stated rather than promising access to every platform.

Participation and presence signals inherit the disclosure limits in DP4 and the provenance and reputation constraints in DP15 and DP18. Campaign transparency must not require public personal participation records or turn portability into global identity linkage.

11. Participation Model

DP8 defines participation as a **tiered, stateful system** where capability, influence, and accountability increase with demonstrated behavior and verified identity properties (DP1), under enforceable governance (Section 10.4).

Civic participation should make public discussion, proposals, and feedback accessible without unnecessary institutional or platform gatekeeping. Access to discussion remains distinct from the credentials, standing, or mandate needed for consequential actions under a zone's rules.

11.1 Tiered Participation (Capabilities Matrix)

Participation tiers SHOULD be explicit and machine-enforceable:

Tier	Capabilities	Constraints
Observer	Read, follow context	No amplification or governance actions
Contributor	Comment, annotate, submit content	Rate-limited; no virality control
Trusted Participant	Signal trust, influence ranking/visibility	Requires continuity and reputation thresholds
Steward	Moderate, adjudicate, configure rules	Requires strong identity guarantees and auditability

Systems MUST bind capabilities to tier and prevent out-of-band escalation.

11.2 Entry, Progression, and Decay

- Entry requirements MAY include consent, identity level, and basic behavior thresholds.
- Progression MUST require **verifiable contribution over time** (continuity, not bursts).
- Systems SHOULD implement **decay** (time-based or behavior-based) to prevent permanent privilege lock-in.

Badges, points, and tiers signal reliability and trustworthiness; they do not confer permanent privilege. Early participation earns no permanent advantage: standing is re-earned through current contribution, and newcomers can reach the same standing on the same terms, so recognition does not harden into a hierarchy that keeps new voices out.

Failure modes:

- **fast-track escalation** (gaming entry to gain influence)
- **privilege ossification** (roles never decay)

11.3 Virality and Reputation Controls

High-impact amplification SHOULD require unique human verification.

Systems MUST remain stable under coordinated attempts to manipulate participation tiers, including bot-driven amplification, identity cycling, and reputation inflation. Participation models must ensure that influence cannot be rapidly accumulated without verifiable contribution and continuity.

Mechanisms MAY include:

- amplification caps per identity/time window
- quorum requirements for boosts (N unique humans)
- reputation weighting with context binding

Failure modes:

- **amplification spoofing**
- **reputation laundering**

11.4 Cross-Zone Participation Semantics

- Participation status is **zone-scoped by default**.
- Systems MUST signal when a participant's tier in one zone does not transfer to another.
- Optional bridges MAY allow partial portability with explicit downgrade rules.

Freedom of association includes forming and sustaining communities across platform boundaries. Moving between those contexts should not require surrendering community continuity, but neither should association in one place automatically confer decision authority elsewhere.

Failure mode: **cross-zone escalation**, where status in one zone illegitimately confers power in another.

11.5 Rate, Scope, and Safety Guards

- Systems MUST enforce rate limits and scope constraints proportional to tier.
- High-risk actions (mass messaging, mass tagging, bulk edits) require stricter proofs and/or stewards.

Failure mode: **throughput abuse**, where volume substitutes for trust.

12. AI Governance (DP12 Link)

DP8 requires that AI participation be **governed as a first-class actor class** within zones, with enforceable constraints at runtime and clear attribution aligned with DP1 and DP2.

In an augmented conversation involving people and AI, changes to a shared simulation, dataset, or system state should be reflected consistently for those authorized to view it. Staleness or divergence should be visible so participants do not unknowingly reason from different versions. Shared state does not establish the truth of its contents; DP21 addresses continuity across representations.

12.1 AI Identity, Attribution, and Disclosure

- All AI agents MUST present a verifiable identity (issuer, operator, model class) and remain attributable for actions.
- AI-originated content and actions MUST be clearly labeled at the interface layer.
- Delegated agents MUST bind to a sponsoring human or organization (DP1, Section 11.3 equivalent), with visible responsibility.

Failure modes:

- **identity masking** (AI indistinguishable from humans)
- **attribution gaps** (no accountable party)

12.2 Scope-Limited Delegation and Control

- AI agents MUST operate within **explicit scopes** (read/write domains, amplification limits, interaction types) with TTL and renewal.

- Zones MUST define allowed capabilities per tier (e.g., no autonomous amplification without quorum).
- Participants MUST have a **kill switch** and bounded-time revocation.

Failure modes:

- **scope creep** (agent expands authority)
- **irrevocable delegation**

12.3 Amplification and Participation Constraints

- AI MUST NOT directly trigger high-impact amplification without **human-backed quorum or proofs**.
- Systems SHOULD cap AI-originated throughput and require stronger proofs for bulk actions.
- AI contributions MAY inform ranking, but MUST be **down-weighted or gated** relative to verified human signals where stakes are high.

Failure modes:

- **AI amplification bypass**
- **throughput dominance**

12.4 Interaction Safety and Interruptibility

- AI actions MUST be **interruptible, reversible (where feasible), and auditable**.
- High-risk actions (payments, legal commitments, public attributions) require **human-in-the-loop confirmation** unless explicitly authorized by zone policy.

Failure modes:

- **automation overrun**
- **irreversible AI actions without consent**

12.5 Data and Inference Boundaries (DP4 Link)

- AI MUST honor data purpose binding and consent propagation (DP4, Section 5.11).

- Inferences generated by AI are **first-class artifacts** with lineage, scope, and revocation/attenuation pathways.

Failure modes:

- **inference misuse**
- **consent bypass via pipelines**

12.6 Cross-Zone Behavior and Containment

- AI permissions are **zone-scoped by default**; cross-zone operation requires explicit reauthorization.
- Systems MUST signal when AI constraints change across zones.

Failure modes:

- **cross-zone privilege leakage**

12.7 Observability and Audits

- Systems MUST provide logs of AI actions (who, what, when, scope) and summaries understandable to participants.
- Zones SHOULD publish **policy manifests** for AI (allowed actions, caps, escalation paths).

Failure modes:

- **AI opacity**

13. Governance Composition

DP8 treats governance as a **composable system of modules** that MUST interoperate without bypassing enforcement (Section 10.4).

13.1 Module Types

Common modules include:

- **Voting** (quorum rules, weighting)
- **Moderation** (flags, queues, actions)
- **Reputation** (signals, decay, context binding)
- **Access Control** (roles, permissions)

- **Dispute Resolution** (appeals, juries)

13.2 Composition Constraints (Required)

- Modules MUST NOT bypass the governance system layer (no direct amplification without checks).
- Outputs of one module MUST be **typed and scoped** before feeding another (e.g., a vote signal cannot directly amplify content without validation).
- Cycles MUST be bounded to prevent feedback loops (e.g., reputation → visibility → reputation).

Failure modes:

- **module bypass** (side-channel influence)
- **feedback loops** (runaway amplification)

13.3 Precedence and Policy Graph

- Systems SHOULD maintain a **policy graph** where modules declare inputs, outputs, and precedence.
- Conflicts between modules MUST resolve via declared precedence (or fall back to stricter rule wins).

Failure mode: **composition ambiguity**, where multiple modules conflict without resolution.

13.4 Forkability and Versioning

- Governance stacks MUST be forkable with clear version identifiers.
- Changes MUST be **versioned and auditable**, with migration paths for participants.

Failure mode: **silent rule drift**, where behavior changes without visibility.

13.5 Interoperability of Modules

- Modules SHOULD expose standard interfaces for signals (e.g., vote, trust, flag) to enable cross-community reuse.
- Interop MUST preserve context (zone, scope, guarantees) or explicitly degrade it.

Failure mode: **semantic mismatch**, where signals are misinterpreted across systems.

14. Relationship to Other Desirable Properties

DP8 is the layer at which the other properties become locally binding. It supplies the context, the zone, in which identity, agency, data, incentives, and automation are constrained in ways a specific community can define and defend.

- **DP1** supplies the identity and attribution guarantees that make tiers, roles, and governance actions accountable; without it, participation integrity and capture resistance cannot hold
- **DP2** supplies participant agency, ensuring zone rules change outcomes participants can perceive and control rather than preferences they merely configure
- **DP3** supplies the adaptive governance patterns through which zone rules evolve, fork, and scale beyond a single community
- **DP4** supplies purpose binding and consent propagation, so that moderation, ranking, and reputation cannot launder data use through governance actions
- **DP5** supplies stable naming and resolution for zones, roles, and governance artifacts so that references survive movement
- **DP7** carries governance state across systems and requires honest signaling of what degraded in transit; DP8 defines what must not be silently lost
- **DP9** aligns rewards with sustained contribution, so influence cannot be purchased through throughput and stewardship labor is resourced
- **DP11** defines the ethical floor for AI participation that zone policy tightens but may not fall below
- **DP12** turns zone rules into executable policy objects with receipts; DP8 defines what communities are entitled to govern
- **DP13** enforces containment, sandboxing, and rate limits that make zone constraints binding on automated actors
- **DP14–DP15** supply transparency and provenance for governance decisions, making auditability and contestability practical

- **DP18** supplies feedback and reputation dynamics that participation tiers depend on
- **DP20** ensures communities own their governance configuration and can fork it, rather than licensing it from an operator

A failure in any of these layers surfaces inside DP8 as illegitimate governance: rules that exist, apply unevenly, and cannot be traced or changed.

15. Non-Goals and Explicit Boundaries

DP8 does not:

- prescribe a single governance model, voting method, moderation policy, or tier structure
- require communities to converge on shared norms, or treat disagreement between zones as a defect
- guarantee that community-defined governance produces good, fair, or wise decisions
- eliminate moderation, expertise, delegation, or stewardship roles in favor of pure direct participation
- replace legal systems, jurisdictional obligations, or platform liability
- promise that every governance guarantee survives every interoperability boundary, only that loss is signaled rather than hidden
- treat zone autonomy as a shield against baseline safety, identity, data, and AI constraints defined elsewhere in the stack
- assume communities are internally homogeneous or that majority processes protect minorities within them

DP8 defines the conditions under which community-defined governance is enforceable, portable, attributable, and contestable. It does not define what communities should decide.

16. Minimum DP8 Alignment

Minimum alignment is not a feature checklist. It is the threshold at which governance is **enforceable, portable, and resistant to manipulation, capture, and coordination attacks.**

A system that does not meet these conditions may expose governance features, but it does not provide meaningful community control.

At minimum, a system claiming DP8 alignment MUST satisfy the following **irreducible conditions**:

16.1 Zone-Based Enforcement

- Governance rules MUST be enforced at the interface layer within defined zones
- Rules MUST apply before actions (visibility, amplification, moderation) propagate
- Systems MUST signal when zone protections are absent or degraded

Failure mode: **phantom governance**

16.2 Participation Integrity

- Participation tiers MUST map to real differences in capability and influence
- High-impact actions (e.g., virality, reputation boosts) MUST require stronger identity guarantees (e.g., unique human verification where appropriate)
- Systems MUST prevent rapid escalation of influence without earned progression

Failure mode: **participation gaming**

16.3 Governance Continuity

- Governance state (roles, permissions, reputation) MUST persist across pages, sessions, and supported systems
- Systems MUST signal when continuity breaks

Failure mode: **governance fragmentation**

16.4 Capture Resistance

- Systems MUST include mechanisms to detect and mitigate coordinated influence, role entrenchment, and opaque decision concentration
- Governance actions MUST be attributable and reviewable

Failure mode: **governance capture**

16.5 Anti-Brigading Protections

- Systems MUST detect anomalous participation patterns and coordinated behavior
- Influence spikes MUST be rate-limited or require stronger proofs

Failure mode: **brigading**

16.6 Governance Propagation and Boundary Signaling

- Governance context MUST travel with content, participants, and interactions where technically feasible
- Systems MUST signal when governance constraints are lost or degraded across boundaries

Failure mode: **governance stripping**

16.7 Auditability and Contestability

- Participants MUST be able to inspect governance decisions and their effects
- Systems MUST provide mechanisms to challenge or appeal decisions

Failure mode: **governance opacity**

16.8 AI Governance Enforcement

- AI actions MUST adhere to community-defined constraints
- Systems MUST visibly distinguish AI participation and enforce scope limits

Failure mode: **AI governance bypass**

These conditions define the **minimum viable governance layer** of the Meta-Layer.

Partial implementations that omit enforcement, continuity, or capture resistance MUST NOT be considered aligned with DP8.

17. Open Questions and Future Work

Open questions focus on cross-DP integration and operationalization:

17.1 Cross-Zone Conflict Models (DP1, DP4)

- What precedence models are most legible and safe (stricter-wins vs user-selected vs negotiated)?
- How should conflicts be surfaced without overload?

17.2 Reputation Portability vs Context (DP2, DP8)

- What minimal signals can travel without enabling laundering?
- How should decay and re-qualification work across zones?

17.3 AI Policy Manifests (DP12)

- What is the minimal, machine-readable schema for zone AI policies?
- How are capabilities negotiated across zones?

17.4 Governance Module Standards (DP7)

- Which module interfaces should be standardized for interoperability?
- How to prevent semantic drift across implementations?

17.5 Data–Governance Coupling (DP4)

- How should consent and purpose binding propagate with governance actions (e.g., moderation, ranking)?

17.6 Incentive Alignment (DP9)

- What reward models avoid funding abuse while sustaining participation?

18. Path Toward ML-RFC

Advancement from ML-Draft to ML-RFC for DP8 requires **demonstrated, adversarially tested governance systems operating across identity (DP1), agency (DP2), data (DP4), and AI constraints (DP12).**

This is not a documentation milestone. It is an **operational validation threshold.**

18.1 Reference Implementations (End-to-End Zones)

At least one fully functional governance zone MUST be implemented with:

- Enforced participation tiers with progression and decay (DP2)
- Identity-bound roles and attribution for all governance actions (DP1)
- Data-aware moderation and ranking (purpose-bound, consent-propagating) (DP4)
- AI agents constrained at runtime with visible scope and revocation (DP12)

The implementation MUST demonstrate that governance rules **change outcomes in real time**, not post-hoc.

18.2 Adversarial Conformance Testing

Systems MUST pass structured tests simulating real attack conditions:

- **Sybil + brigading attacks** → no uncontrolled amplification
- **Cross-zone escalation attempts** → no illegitimate privilege transfer
- **Reputation laundering attempts** → no unbounded trust carryover
- **AI amplification bypass** → no autonomous virality without required proofs
- **Governance stripping across interop boundaries** → no silent loss of constraints

Results MUST be documented and reproducible.

18.3 Interoperability Proofs (DP7 Alignment)

Governance systems MUST demonstrate:

- Transfer of governance state (roles, signals, constraints) between at least two independent implementations
- Explicit signaling of **what is preserved vs degraded** during transfer
- No silent reinterpretation of governance semantics across systems

This ensures governance is not platform-bound.

18.4 Auditability and Evidence Artifacts

Systems MUST produce auditable artifacts demonstrating:

- Attribution of governance actions (who acted, under what authority) (DP1)
- Outcome impact (how rules changed visibility, amplification, or access) (DP2)
- Data compliance (how consent and purpose were preserved) (DP4)
- AI behavior logs (what actions agents took and under what constraints) (DP12)

Artifacts SHOULD include:

- structured logs
- participant-readable summaries
- dispute/appeal traces

18.5 Governance Evolution and Forking Evidence

Communities MUST demonstrate the ability to:

- Modify governance rules without breaking continuity
- Fork governance models and continue operation
- Migrate participants across versions with explicit signaling

This proves governance is **adaptive rather than brittle**.

18.6 Multi-Community Adoption

At least two independent communities MUST:

- Operate distinct governance configurations
- Demonstrate real usage under different risk and cultural contexts
- Show evidence of governance effectiveness and evolution

This ensures DP8 is not optimized for a single use case.

18.7 Criteria for Promotion to ML-RFC

DP8 may be promoted when:

- Governance is proven enforceable under adversarial conditions

- Cross-DP integration (DP1, DP2, DP4, DP12) is validated in practice
- Interoperability is demonstrated with explicit degradation semantics
- Communities can evolve and fork governance without system failure
- Participants can understand, audit, and contest governance outcomes

19. Closing Orientation

DP8 defines the conditions under which communities become **sovereign coordination environments** rather than passive audiences.

Without enforceable governance, trust collapses into manipulation.

With it, the Meta-Layer becomes a **civic substrate for collective intelligence**.

Continue the conversation on DP8 online →

DESIRABLE PROPERTY 9

DP9

Developer and Community Incentives

The Meta-Layer gives developers and community builders the tools and incentives to create shared value across the web.



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp09

1. Purpose of This Draft

The Meta-Layer is a shared layer across existing websites where overlay apps, smart tags, agents, and communities operate on shared interfaces rather than inside a single platform. Incentive programs in it involve four main roles: the funder who supplies the money, the administrator who runs the program, the reviewer who judges the work, and the contributor who does it and gets paid. This chapter asks developers to expect published rules and portable receipts, asks community organizers to run programs others can check and appeal, and asks funders to disclose their role and keep rules stable once work begins.

This draft articulates Desirable Property 9 (DP9) as the condition under which developers and communities receive credible, legible incentives to build, maintain, and improve Meta-Layer capabilities, without collapsing into extraction, engagement-only rewards, or token theater that severs contribution from accountability.

DP9 is how interoperability (DP7), meta-communities (DP8), education (DP10), commerce (DP6), and ownership (DP20) become sustained rather than hobbyist. It connects to DP4 (privacy-preserving attribution where needed), DP2 (participant agency over incentives they accept), DP17 (financial sustainability), and DP11–DP13 (safe incentives for AI-assisted development and operations).

If DP9 is weak, predictable failures follow: burnout maintainers, underfunded commons, predatory grant programs, misaligned metrics (growth over safety), capture of incentive systems by whales, and automated spam farms gaming rewards.

DP9 does not prescribe a single tokenomics paper. It defines minimum legitimacy conditions for incentive systems: clarity, fairness, auditability, exit, and alignment with the Meta-Layer's human-first goals.

2. Problem Statement

In today's web, builders often face misaligned incentives: metrics reward engagement and growth while externalizing harm (misinformation, addiction, privacy loss). Open-source maintainers carry systemic risk with little

capture of value. Communities that host quality spaces rarely receive durable upside from the ecosystems they enable.

In practice, this produces recurring failures:

- bounty programs that pay for volume, inviting spam and low-quality output
- grants with opaque selection and slow payout, discouraging small teams
- platform APIs that change terms after dependency forms
- creator funds that function as marketing, not structural revenue share
- AI tools that accelerate low-effort contribution farming unless bounded (DP13)

These failures are structural: when incentives are opaque or misaligned, the fastest path to reward is usually not the best path for participants. DP9 reframes incentives as governance objects: measurable, contestable, and evolvable, parallel in spirit to DP12's insistence that rules be executable and revisable.

3. Threats and Failure Modes

3.1 Metric capture

Teams optimize what is measured, often engagement, at the expense of safety, truth, and inclusion.

Example: A developer fund ranks submissions by user minutes, incentivizing addictive mechanics.

Why this matters: DP9 requires multi-metric incentive design with explicit tradeoffs and red lines.

3.2 Incentive laundering

Rewards flow to intermediaries instead of contributors, or to shell identities.

Example: A community rewards pool is drained by coordinated sockpuppets farming tasks.

Why this matters: This needs DP1 accountability and DP13 rate and containment patterns.

3.3 Opaque allocation

Participants cannot see why someone was funded or ranked. Suspicion corrodes cooperation.

Example: A hackathon winner is an insider portfolio company with undisclosed relationships.

Why this matters: DP14 transparency norms apply to incentive governance too.

3.4 Maintainer extraction without support

Corporations build on commons without returning maintenance, security, or governance labor.

Example: Critical libraries burn out maintainers while enterprises profit.

Why this matters: DP9 expects credible reciprocity mechanisms, not moral appeals alone.

3.5 Education and onboarding gaps

Incentives assume skills participants do not have, excluding global majority builders.

Example: Only teams fluent in a niche stack can compete for integration grants.

Why this matters: Fair incentives require reachable on-ramps and mentorship surfaces, with a clear DP10 connection.

3.6 Token-based confusion

Tokens substitute for clear rights and clear work, creating regulatory and UX hazards.

Example: Governance tokens are issued without enforceable decision rights or measurable duties.

Why this matters: DP9 allows tokens as one instrument, not a synonym for incentives.

3.7 Sponsored open source capture

Corporate roadmaps steer commons toward vendor stacks. Incentives reward integrations that reduce portability.

Example: A grant prioritizes one cloud's proprietary APIs over open interfaces.

Why this matters: DP7 interoperability is an outcome incentive designers must protect, not accidentally punish.

3.8 Reviewer burnout and queue collapse

Underfunded programs flood reviewers. Quality decisions become random or biased by who shouts loudest.

Example: A security bounty program misses critical reports while reviewers chase low-risk noise for points.

Why this matters: Incentive design must include capacity models for evaluation, not only payout curves.

3.9 Geographical and language bias

English-first rubrics and US-centric eligibility exclude global majority talent.

Example: A hackathon requires on-site presence in one city for finals.

Why this matters: Fairness requires reachable participation surfaces, with a clear DP10 connection.

4. Core Principle

Developer and community incentives in the Meta-Layer must be transparent in allocation and metrics; aligned with safety and interoperability outcomes; resistant to gaming and capture; reciprocal toward commons maintenance; and evolvable through participatory governance, with bounded automation and clear accountability.

Incentive systems should feel like public infrastructure, not casino tables.

Example: A grants round publishes rubric weights, reviewer conflicts, payout schedule, and post-hoc impact report, with appeals for mistaken rejections.

What this feels like: You can learn the game without insider knowledge, and trust that gaming the game hurts you more than the community.

Without this: Builders rationally exit or optimize the wrong surface.

5. Primary Mechanisms and Structural Conditions

5.1 Incentive Layer: Allocation, Signal, and Enforcement

Incentives in the Meta-Layer are not abstract reward schemes. They are operational systems that allocate value across participants, tools, and infrastructure in response to real activity.

In many systems, incentives fail because they are not bound to contribution quality, are not auditable, or are captured by intermediaries. Rewards become detached from value creation, and systems devolve into gaming contests.

DP9 therefore requires a shared incentive layer composed of primitives that make incentives legible, enforceable, and aligned with governance, interoperability, and ownership across systems.

5.1.1 Incentive objects

Incentives must be represented as structured, machine-readable objects.

An incentive object includes:

- metric: what is being measured (e.g. endorsements, retention, contributions)
- weight: how much that metric matters
- constraints: what disqualifies or reduces reward
- eligibility: who or what can earn
- decay and clawback: how rewards adjust over time based on outcomes

This allows incentives to be governed, versioned, and enforced (DP12, DP3).

Incentive objects must also declare their scope, context, and transferability so receiving systems can enforce appropriate constraints (DP7). Without declared intent, incentives may be misapplied or exploited when moved across environments.

A failure mode is metric ambiguity, where participants cannot determine what actions produce rewards, leading to manipulation or disengagement.

5.1.2 Contribution binding

Rewards must be bound to verifiable contributions rather than surface-level signals.

This requires linkage to identity (DP1), traceability of actions, and resistance to duplication or replay.

For small rewards, programs should offer a lower-assurance identity path so that tying rewards to a verified identity does not shut out students, immigrants, and builders outside the US, the same participants Section 3.9 aims to include.

Contribution binding ensures that value flows to actual work rather than its simulation.

A failure mode is contribution spoofing, where low-effort or automated actions are mistaken for meaningful input.

5.1.3 Attribution and lineage

Where a program is large or automated enough to support it, incentive systems track contribution lineage across time and systems. Like reward splitting, lineage tracking is a higher-tier instrument: a small program that pays contributors directly is not out of alignment for omitting it.

Rewards should reflect original creation, downstream reuse, and derivative contributions.

Lineage enables fair distribution of value across ecosystems rather than concentrating rewards at endpoints.

A failure mode is attribution collapse, where downstream actors capture disproportionate value due to missing lineage.

5.1.4 Reward event splitting

Where a program is large or automated enough to support it, each value-generating event may distribute rewards across contributing layers. Reward splitting is a higher-tier instrument: a small program that pays contributors directly is not out of alignment for omitting it.

This includes the primary contributor, interface layer, access layer, and shared infrastructure.

Reward splitting ensures that invisible layers of contribution are not systematically undercompensated.

A failure mode is endpoint capture, where only the visible actor receives rewards despite reliance on shared systems.

5.1.5 Signal weighting and quality adjustment

Incentives must account for quality, not just quantity.

Signals should include uniqueness of contribution, downstream usage or impact, and endorsement or validation by others.

This reduces the effectiveness of spam and gaming strategies.

A failure mode is volume dominance, where systems reward quantity over value, degrading overall quality.

5.1.6 Cross-system incentive integrity

Incentives must preserve meaning across systems.

This includes preventing duplication or replay of contributions, ensuring attribution artifacts remain valid across environments, and signaling when rewards lose guarantees in new contexts.

Incentives must not be portable in ways that allow reward without contribution.

A failure mode is cross-system arbitrage, where value is multiplied without corresponding work.

5.1.7 Incentive–ownership binding

At higher tiers, incentive systems may create pathways to ownership (DP20). Ownership stakes and financial tokens are optional instruments,

not conditions of alignment, and a program that offers them should first address the regulatory hazards noted in Section 3.6.

Where such pathways exist, participants who contribute consistently should accumulate stake, gain governance rights, and participate in long-term value flows.

Without any pathway of this kind, incentives produce activity without durable power.

A failure mode is extractive participation, where contributors generate value but do not share in outcomes.

5.1.8 Incentive memory and auditability

All incentive allocations must be recorded and traceable.

Participants must be able to see why rewards were issued, how metrics were applied, and how parameters changed over time.

This enables accountability and continuous improvement (DP15).

A failure mode is opaque allocation, where trust erodes due to lack of visibility.

5.1.9 Adaptive emission and allocation

Incentive systems must respond to system conditions.

This includes increasing rewards to stimulate participation, reducing emissions to prevent oversaturation, and reallocating across categories as needs evolve.

Adaptive systems prevent stagnation and misalignment.

A failure mode is static incentives, where outdated reward structures distort behavior over time.

These primitives do not replace the mechanisms below. They make them operational, enforceable, and resistant to gaming.

5.2 Published incentive constitutions

Incentive systems must begin with explicit, legible declarations of intent and structure. These are not marketing summaries, but operational constitutions that define how value flows and how decisions are made.

Each program must publish goals, metrics, anti-gaming rules, appeals processes, sunset conditions, and funding sources, machine-readable where possible, with clear DP7 alignment.

Without this, participants are forced to infer rules from outcomes, which creates information asymmetry and invites manipulation.

A failure mode is post-hoc rule discovery, where participants only understand incentive logic after being penalized or excluded.

5.3 Multi-metric scoring with red lines

Single-metric systems inevitably collapse into optimization loops that distort behavior. Incentive systems must therefore incorporate multiple dimensions of value.

Safety, accessibility, privacy impact, and interoperability must function as explicit constraints or gating conditions, not optional considerations.

This ensures that high-performing contributions cannot violate core system values while still being rewarded.

A failure mode is metric domination, where one signal overwhelms others and reintroduces harmful optimization patterns.

Example: A bounty disqualifies integrations that exfiltrate unnecessary data (DP4).

5.4 Commons reciprocity

Incentives must ensure that systems benefiting from shared infrastructure contribute back to its maintenance and evolution.

Commercial beneficiaries of public goods must return value through fees, maintainer support, or mandated upstream contributions, with clear linkage to DP6 and DP17.

This creates a closed loop between extraction and regeneration, preventing systemic underfunding of critical layers.

A failure mode is asymmetrical value flow, where commons are continuously drawn from but not replenished.

5.5 Credible neutrality in allocation

Allocation mechanisms must be designed to resist bias, capture, and insider advantage.

Selection processes must publish conflicts of interest, include randomization or audit layers, and report on diversity and fairness outcomes.

This ensures that allocation decisions are not only fair, but perceived as fair, which is critical for participation.

A failure mode is hidden favoritism, where trust erodes due to perceived or real insider influence.

5.6 Micro-rewards and milestone cadence

Incentive systems must balance immediacy with long-term commitment.

Small, fast rewards for incremental progress reduce participation friction and provide continuous feedback, while larger grants anchor strategic efforts.

This creates a cadence that supports both experimentation and sustained work.

A failure mode is reward starvation, where contributors disengage due to delayed or uncertain compensation.

5.7 Recognition that is portable

Recognition must not be trapped within the system that issued it.

Attribution artifacts, credentials, and receipts of contribution must interoperate across tools without locking reputation into a single environment, with clear linkage to DP5 and DP7.

This ensures that contributors can carry their history and credibility with them.

A failure mode is reputation lock-in, where value accrues to platforms rather than participants.

5.8 Anti-spam and anti-farm containment

Incentive systems must actively resist exploitation through automation, coordination, or scale.

Automated submission and AI-generated bulk work must be rate-limited, attested, and reviewed using DP13 containment patterns, including rate limits on rewardable actions, detection of clustered or synthetic behavior, and discounting or quarantining suspicious contributions.

This ensures that incentives remain aligned with meaningful contribution rather than extractive behavior.

A failure mode is reward farming, where systems devolve into competition for extraction rather than value creation.

5.9 Participatory evolution

Incentive systems must evolve through visible, governed processes rather than opaque adjustments.

Communities must be able to adjust parameters, introduce new metrics, and retire ineffective structures through DP12-aligned governance processes.

All changes must include memory of why they were made, linking decisions to outcomes over time.

A failure mode is silent drift, where incentive systems change without explanation, eroding trust and predictability.

5.10 Safety and interop hard gates

Certain classes of submissions, including browser extensions, network agents, and payment integrations, must pass automated checks plus human review for high-risk categories, coordinating DP13 containment with DP11 disclosure expectations.

Example: A grant auto-rejects SDKs that request excessive permissions without justification fields.

5.11 Long-horizon stewardship incentives

Incentive systems must support maintenance, not only novelty. Many ecosystems reward launches, prototypes, and visible growth while neglecting the quiet work that keeps shared infrastructure secure, usable, and trustworthy over time.

DP9 requires long-horizon stewardship incentives such as multi-year maintenance awards, escrowed vesting tied to documented upkeep, and penalties or de-prioritization for abandoned critical packages. These mechanisms align directly with DP17 sustainability by treating maintenance as value creation, not background labor.

A core failure mode is launch bias, where contributors are rewarded for creating new tools but not for maintaining the systems others depend on. Over time, this produces fragile infrastructure, security risk, and maintainer burnout.

Long-horizon incentives should therefore reward documented care work: issue triage, security patches, dependency updates, accessibility improvements, moderation support, and governance participation. Without this, the Meta-Layer risks building impressive surfaces on top of neglected foundations.

5.12 Delegated and agent-mediated incentives

Incentive systems must account for agents acting on behalf of participants.

This includes binding rewards to principal intent, ensuring agent actions are auditable, and preventing agents from exploiting scale or speed to farm rewards.

Delegation must not reduce accountability. Agent-mediated incentives must remain reconstructable and interruptible by participants and governance systems.

A failure mode is automated exploitation, where agents extract rewards beyond human oversight.

6. Governance, Accountability, and Agency Surfaces

Incentive governance determines whether participants experience reward systems as legitimate infrastructure or as arbitrary extraction games. Because incentives shape behavior directly, the rules of allocation must be visible before contribution, contestable after decisions, and revisable when evidence shows misalignment.

Developers must not be forced to gamble on hidden criteria. Communities must not be forced to accept incentive systems that reward harm, spam, or capture. DP9 requires governance surfaces that allow both builders and communities to understand, challenge, and reshape the incentive environments they depend on.

Developers must be able to:

- understand evaluation criteria before investing time
- appeal mistaken denials or gaming accusations with timelines
- see who funds the program and what conflicts may exist
- understand whether rewards, attribution, and constraints persist across systems (DP7)

Communities must be able to:

- propose new incentive programs or parameter changes
- audit outcomes and redistribute future rounds based on evidence
- halt programs captured by narrow interests
- require redesign when incentives produce spam, exclusion, or unsafe optimization

Agency also requires cross-system clarity. If rewards earned in one environment are valid elsewhere, participants should understand how attribution, eligibility, rate limits, and anti-gaming rules travel. If they do not travel, that degradation must be visible rather than discovered after contribution.

Without these surfaces, incentive systems become illegitimate even when payouts occur. Contributors may receive rewards, but they cannot know whether the system is fair, whether rules have changed, or whether insiders are operating under different conditions.

Example: A community freezes a rewards pool after detecting coordinated farming. Funds carry over with a redesigned rubric co-authored in public, and future rounds include tighter attribution checks, clearer appeal paths, and public reporting on rejected farming attempts.

7. Incentives and Power Analysis

Incentive systems determine what a system actually does, regardless of what it claims to value.

In many environments, governance rules and public commitments exist, but incentives quietly direct behavior toward growth, engagement, or extraction. This creates a structural split between stated goals and actual outcomes.

DP9 requires that incentives be treated as power structures, not just reward mechanisms.

This includes making visible:

- what behaviors are being optimized for in practice
- who benefits from those optimizations
- how reward systems shape contribution patterns and governance outcomes

Example: A system publishes strong safety policies, but rewards tools that maximize usage. Builders rationally optimize for usage, not safety, and harmful dynamics persist.

Why this matters: Incentives override intent. If they are misaligned, governance becomes performative.

DP9 therefore expects:

- explicit alignment between incentives and governance constraints
- the ability to constrain or redirect incentives at the community level
- visibility into how incentive parameters influence outcomes over time

When incentives and governance align, systems reinforce their stated values. When they diverge, systems drift toward extraction.

8. Community Signals Informing DP9

Across ecosystems, recurring signals point to structural breakdowns in incentive design:

- maintainers asking for predictable, ongoing support rather than one-time grants
- distrust of opaque funding decisions and insider advantage
- fatigue with engagement-driven metrics that reward low-quality output
- concern that AI will flood contribution pipelines with low-effort work
- demand for recognition and rewards that persist across tools and platforms

These signals reflect a consistent pattern: contributors are willing to participate, but not under conditions where incentives are unclear, unfair, or easily gamed.

DP9 treats these signals as design inputs, not complaints.

9. Implementation Patterns

These patterns turn DP9 from a set of legitimacy conditions into operational choices a program can adopt now. Each is compatible with multiple funding models and none require a token.

9.1 Published incentive constitution

Ship the program's rules as a versioned document with machine-readable metrics, weights, constraints, eligibility, appeal paths, funding sources, and sunset conditions. Changes are diffed, dated, and announced before they take effect.

9.2 Rubric-first review

Publish the scoring rubric and reviewer conflict disclosures before submissions open. Score against the rubric, publish aggregate score distributions afterward, and report where reviewers disagreed.

9.3 Retrospective allocation with evidence bundles

Reward demonstrated outcomes rather than proposals for a portion of each pool. Contributors submit evidence bundles (artifacts, usage, downstream reuse, audits) that reviewers score against published criteria.

9.4 Reward event splitting

At each value-generating event, distribute across the primary contributor, the interface layer, the access layer, and shared infrastructure using declared split rules, so invisible layers are not systematically unpaid.

9.5 Streaming maintenance grants

Fund upkeep as a continuous stream tied to documented care work (triage, security patches, dependency updates, accessibility fixes) with periodic review rather than reapplication. Escrow vests against evidence.

9.6 Probation and staged payout

Pay new contributors in tranches as work is verified. Farming attempts are throttled at the first tranche rather than after full payout, and legitimate contributors reach full rate quickly.

9.7 Quality-weighted throttles

Rate-limit rewardable actions per identity and per time window, weight signals by uniqueness and downstream usage, and discount clustered or synthetic behavior pending review (DP13).

9.8 Portable contribution receipts

Issue signed receipts for every allocation, naming the metric applied, the version of the rubric, the evidence cited, and the responsible reviewer or process. Receipts travel with the contributor (DP7, DP15).

9.9 Public appeal queues with service levels

Run appeals as a visible queue with stated timelines, published outcome categories, and periodic reporting on reversal rates. Appeals are governance evidence, not customer support.

9.10 Circuit breakers and pool freezes

Define in advance the anomaly thresholds that pause a pool, who may trigger a freeze, how long it may last, and what must be published when it happens. Carry funds forward rather than forfeiting them.

9.11 Reachable on-ramps

Provide mentorship pairing, translated rubrics, asynchronous participation, regional eligibility, and small first-contribution bounties so that program access does not depend on language, timezone, or insider familiarity (DP10).

9.12 Post-round impact reporting

Publish what was funded, what shipped, what failed, what was rejected for gaming, and what the program changed as a result. Reporting closes the loop between allocation and evidence.

10. Foresight and Failure Design

DP9 assumes that incentive systems operate under continuous adversarial pressure from participants, intermediaries, and automated agents. Failures rarely appear as single events. They emerge as gradual drift between stated goals and rewarded behavior.

Common failure paths include:

- Sybil attacks and coordinated farming of rewards
- metric capture that shifts focus toward low-quality output
- sponsor or funder capture of incentive programs
- divergence between reward signals and actual value creation
- cross-system replay or duplication of contributions for multiple rewards
- agent-mediated extraction that exploits speed and opacity

These failures compound. As systems scale, review capacity lags, increasing reliance on automation or heuristics. This can widen gaps between intent and outcome, allowing exploit patterns to persist long enough to reshape norms.

DP9 therefore requires designing safeguards in advance, including:

- rate limits, eligibility thresholds, and identity-aware constraints (DP1, DP13)
- dynamic weighting and reputation-based adjustments with clear bounds
- circuit breakers for pausing compromised programs or pools
- cross-system anomaly detection for replay, duplication, or routing abuse
- public postmortems linking failures to parameter and rule changes (DP12)

Incentive systems must also detect **slow failure**: when rewards remain technically correct but increasingly misaligned with desired outcomes.

Failure is expected. Invisible or unaccounted failure is not.

11. Developer Reach

DP9 pairs incentives with reach, because compensation without distribution is a grant program rather than an ecosystem. A developer who can be paid but cannot be found, installed, or trusted has no durable position in the Meta-Layer.

Reach in the Meta-Layer means that a tool built once can operate across contexts, be discovered on merit, and accumulate reputation that its author keeps. In today's web, distribution is the primary lever platforms use to extract terms after dependency forms: rankings change, APIs close, and the cost of the switch falls entirely on the builder.

11.1 Build once, operate across contexts

Overlay apps, smart tags, agents, and services bind to shared interfaces rather than to a host platform. A tool written against declared interfaces operates wherever those interfaces are honored, and its author is not required to maintain per-platform variants to remain reachable (DP7).

11.2 Permissionless publication with governed operation

Publication does not require approval. Operation requires conformance: declared permissions and data scopes, runtime binding to zone policy, containment tiers and rate limits, signed artifacts with provenance, and aud-

itable event logs (DP12, DP13, DP15). Openness is at the point of entry; accountability is at the point of execution.

11.3 Discovery that cannot be bought outright

Discovery surfaces weight conformance, audit status, and reputation earned through non-abusive usage. New entrants begin in a probation tier with limited visibility and graduate on evidence. Paid placement, where it exists, is labeled and cannot displace merit-based surfacing.

11.4 Reputation the builder keeps

Install counts, audit results, incident history, review outcomes, and contribution receipts are portable artifacts bound to the builder's identity rather than to a store listing. Leaving a distribution surface costs distribution, not history (DP1, DP7).

11.5 Stable interface contracts

Interfaces are versioned, publicly tested, and deprecated on published schedules with migration guidance. Dependency is safe when the terms of dependency cannot change without notice.

11.6 Reach for community builders, not only code

Facilitation, moderation, curation, translation, and education produce reach in the same sense: a well-run zone, a maintained glossary, or a trusted collection carries audience and credibility. DP9 treats these as distributable contributions eligible for the same discovery and reward surfaces (DP10, DP19).

Example: A two-person team publishes a provenance-checking sidebar. It appears immediately in a probation tier, declares read-only annotation access, and runs sandboxed with signed logs. As communities adopt it and audits pass, its visibility and permissions expand. Six months later the team moves to a different distribution surface; installs must be re-earned, but audit history, reviews, and contribution receipts move with them.

Failure mode: distribution hostage-taking, where reach is granted cheaply, becomes load-bearing, and is then repriced against builders who cannot leave.

Failure mode: conformance as gatekeeping, where safety requirements are set at a cost only incumbents can pay, converting legitimate constraints into a moat.

12. Incentive Mechanisms

The structural conditions above define what makes an incentive system legitimate. This section catalogs the concrete instruments that satisfy them, and the specific way each fails. No single instrument is sufficient; healthy ecosystems run several with different time horizons and risk profiles.

12.1 Bounties

Fixed rewards for scoped, verifiable tasks. Fast, legible, and well suited to defect repair, accessibility fixes, and integration work.

- **Requires:** clear acceptance criteria, reviewer capacity, and duplicate-claim handling
- **Failure mode:** volume optimization, where payment per task invites low-quality submission floods

12.2 Grants

Forward-looking allocation for proposed work. Suited to exploratory or infrastructural efforts that cannot be scoped as tasks.

- **Requires:** published rubrics, conflict disclosure, staged milestones, and timely payout
- **Failure mode:** proposal craft displacing delivery, and insider advantage in selection

12.3 Retrospective funding

Backward-looking allocation for demonstrated impact. Removes proposal overhead and rewards work that was done without a promise of payment.

- **Requires:** evidence bundles, impact criteria, and protection against popularity proxies
- **Failure mode:** visibility bias, where legible work is funded and load-bearing invisible work is not

12.4 Matching pools

Community signal amplified by a shared pool, so that many small endorsements direct larger allocation.

- **Requires:** Sybil resistance and identity-aware weighting (DP1, DP13)
- **Failure mode:** collusion rings and wealth-weighted signal masquerading as community preference

12.5 Streaming rewards

Continuous payment tied to ongoing conditions rather than discrete events. Suited to maintenance, moderation, facilitation, and stewardship.

- **Requires:** liveness checks, documented care work, and review with graceful termination
- **Failure mode:** annuity capture, where streams persist after the work stops

12.6 Reward splitting

Automatic distribution of a value event across contributing layers, including the primary contributor, the interface, the access layer, and shared infrastructure.

- **Requires:** attribution lineage and declared split rules (DP15, DP20)
- **Failure mode:** endpoint capture when lineage is missing, and split gaming when it is manipulable

12.7 Commons levies and reciprocity fees

Charges on commercial beneficiaries of shared infrastructure, routed to its maintenance.

- **Requires:** transparent basis of assessment and independent stewardship of proceeds (DP6, DP17)
- **Failure mode:** rent extraction under the language of reciprocity

12.8 Stewardship endowments

Long-horizon funds that pay for critical upkeep independent of annual program cycles.

- **Requires:** governed disbursement, published mandate, and succession planning
- **Failure mode:** endowment capture, where control of the fund becomes the prize

12.9 Recognition and credentials

Non-financial rewards: attribution, credentials, badges, and roles that carry across systems.

- **Requires:** evidence backing, portability, and revocability (DP7, DP10, DP18)
- **Failure mode:** recognition substituting for compensation

12.10 Ownership and stake pathways

Conversion of sustained contribution into governance rights and long-term value participation.

- **Requires:** clear vesting, decay, and enforceable rights rather than symbolic tokens (DP20)
- **Failure mode:** extractive participation, where contributors generate value but never acquire standing

12.11 Clawbacks and negative incentives

Reversal or reduction of rewards when outcomes prove harmful, fraudulent, or abandoned.

- **Requires:** defined triggers, appeal paths, and bounded retroactivity
- **Failure mode:** arbitrary retroactive punishment that makes participation unpredictable

12.12 Creating and verifying connections

People who create or verify useful connections among artifacts should be eligible for compensation and recognition. Those connections can make existing information navigable and support shared understanding, rather than requiring every contributor to produce a new standalone artifact.

Assessment should consider the connection's contextual value and evidence, not simply the number of links created. Verifiers, curators, and app-

lication-substrate designers are part of this contribution economy. Payment is neither proof that a connection is true nor a substitute for contestable quality assessment.

- **Requires:** assessment of each connection's contextual value and evidence, and contestable quality review
- **Failure mode:** link farming, where the number of connections substitutes for their value, and payment taken as proof that a connection is true

Why this matters: Instrument choice determines behavior more than stated program values. A system that funds only launches will produce launches; a system that funds only tasks will produce task volume. Portfolios of instruments, published with their failure modes, are what allow incentive systems to be debugged rather than merely defended.

13. Relationship to Other Desirable Properties

DP9 connects incentives to the broader Meta-Layer system:

- DP3 defines how incentive parameters evolve through governance
- DP4 constrains how data can be used in measuring contribution
- DP6 defines how real economic value flows through systems
- DP7 enables portability of incentive artifacts and credentials
- DP10 ensures participants can access and benefit from incentive systems
- DP12 ensures incentive rules are executable and revisable
- DP13 enforces constraints on gaming and abuse
- DP15 provides auditability of reward allocation
- DP17 ensures long-term sustainability of incentive pools
- DP20 binds incentives to ownership and durable community power

DP9 is the layer that translates participation into sustained value.

14. Non-Goals and Explicit Boundaries

DP9 does not:

- guarantee equal rewards for all contributors

- eliminate competition among builders
- replace investment markets or capital allocation processes
- mandate tokens or any specific financial mechanism

DP9 defines the conditions under which incentives are legitimate and aligned. It does not prescribe a single economic model.

Recording contributions and paying for them are separate. Where a program uses a shared ledger such as a blockchain, it records receipts, rule versions, and award decisions so anyone can check them, and no one needs to hold or trade a token for those records to be trustworthy. Because ledger records are permanent, personal details stay off the ledger and only a fingerprint of each receipt is written to it (DP4). DP9 does not cover tax reporting, identity checks for payment, payment rails, or how to pay students, volunteers, and other non-employees; programs must resolve these under their own legal obligations.

15. Minimum DP9 Alignment

Minimum alignment is not a checklist of features. It is the threshold at which an incentive system can be considered legitimate, auditable, and resistant to obvious gaming.

A DP9-aligned incentive system should, at minimum:

- define and publish incentive objects with metrics, weights, constraints, and scope
- bind incentives to enforceable mechanisms and containment systems (DP12, DP13)
- produce auditable records of reward allocation (receipts) with lineage (DP15)
- include anti-gaming measures with visible outcomes and appeal pathways
- align incentives with governance and ownership pathways (DP3, DP20)
- signal how incentives behave across systems and where guarantees degrade (DP7)

These conditions must hold **before** scale. Systems that postpone enforcement, auditability, or cross-system clarity will accumulate hidden debt that surfaces as exploitation.

The conditions above describe full alignment for large, automated, or well-funded programs. Smaller programs align through tiers. A basic tier that any small program can meet requires a one-page published set of rules (goals, how work is judged, payment amounts and timing, and funding source), a public scoring rubric, a receipt for every payment or award, a named contact for appeals, and a short report after each round. Heavier requirements such as reward splitting, lineage tracking, streaming payments, and circuit breakers apply at higher tiers. Tiers are an on-ramp, not a permanent lower standard: each tier names the path to the next, so a program can start with published rules and receipts and add more as it grows.

16. Open Questions and Future Work

Key open questions include:

- how to balance simplicity of incentive design with resistance to gaming
- how to achieve Sybil resistance without excluding legitimate participants (DP1)
- how to integrate AI-assisted contribution without rewarding harm acceleration
- how to measure contribution quality across different domains (code, moderation, education)
- how to align global incentive pools with local community priorities
- how to evolve incentive parameters without destabilizing participation

These questions sit at the boundary between economic design and governance implementation.

17. Path Toward ML-RFC

Advancing DP9 toward ML-RFC requires:

- standardizing formats for incentive objects, receipts, and allocation logs

- developing reference implementations of incentive systems with visible outcomes
- integrating identity and accountability layers for Sybil resistance
- testing incentive models across different community types and scales
- aligning incentive systems with governance and ownership frameworks

Progress should be demonstrated through working systems, not only conceptual agreement.

18. Closing Orientation

DP9 is the claim that contribution will be recognized, rewarded, and sustained without requiring extraction or manipulation.

It rejects systems where value flows are hidden, unfair, or disconnected from real work.

When incentives are aligned, participation becomes durable, governance becomes meaningful, and communities can build systems that last.

When incentives are misaligned, even well-governed systems degrade into competition for the wrong outcomes.

Incentives must therefore remain accountable not only at the point of allocation, but across time and across systems. If rewards can be detached from contribution through opacity, replay, or boundary effects, the system will be exploited.

DP9 is the claim that the Meta-Layer will not be built on unpaid miracles or hidden rents.

When incentives are legible and fair, interoperability and community stop being volunteer hobbies. They become careers, crafts, and commons worth defending.

Builders and communities deserve to see the scoreboard, understand how it works, and trust that it cannot be quietly rewritten after the game begins.

Continue the conversation on DP9 online →

DESIRABLE PROPERTY 10

DP10

Education

The Meta-Layer supports dynamic, AI-powered learning tools that adapt to you – not the other way around.



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp10

1. Purpose of This Draft

This draft articulates Desirable Property 10 (DP10) as the condition under which participants can learn, onboard, grow, teach, connect, and co-create within the Meta-Layer without being excluded by technical complexity, jargon, institutional gatekeeping, or static training models.

DP10 defines education as a continuous civic function of the Meta-Layer. It includes onboarding, tool literacy, shared vocabulary, AI-assisted guidance, peer learning, formal and informal curricula, community knowledge-sharing, and portable recognition of learning through credentials such as PEARL digital badges.

The central claim is that the Meta-Layer cannot become public infrastructure unless people can understand it, use it, critique it, teach it, and evolve with it.

Education in the Meta-Layer is not only content delivery. It is contextual learning across the web. Participants learn by doing, annotating, bridging, reflecting, teaching, and contributing inside live environments.

Example: Maya installs a Meta-Layer tool for the first time. A short guided path explains that an overlay is a layer of notes and context that sits on top of a web page, and that a zone is a community space with its own rules. She reads annotations on a news article, asks the AI learning assistant what a bridge is, and learns that it links one page or claim to another with context. A week later she builds her own bridge between two reports on the same event, reflects on why it helps, and a community reviewer awards her a PEARL badge (Section 11), which she decides whether to share.

If DP10 is weak, predictable failures follow: only technical insiders participate meaningfully; newcomers become dependent on intermediaries; communities fragment around misunderstood terminology; AI tools create passive users rather than capable participants; credentials fail to travel; and the Meta-Layer becomes infrastructure people inhabit without understanding.

DP10 connects directly to:

- DP2, participant agency and empowerment
- DP3, adaptive governance

- DP7, simplicity and interoperability
- DP8, collaborative environment and meta-communities
- DP9, developer and community incentives
- DP11, safe and ethical AI
- DP14, trust and transparency
- DP18, feedback loops and reputation
- DP19, amplifying presence and community engagement
- DP21, multi-modal

DP10 does not prescribe one curriculum, credentialing system, or educational institution. It defines the minimum conditions under which learning remains accessible, adaptive, community-grounded, and portable.

2. Problem Statement

Today's web gives people access to information but does not reliably help them develop understanding.

Search results, feeds, tutorials, chats, videos, FAQs, and documentation are abundant, but learning remains fragmented. Participants are expected to navigate unfamiliar tools, opaque algorithms, AI-generated content, misinformation, privacy risks, governance procedures, and technical vocabularies without coherent guidance.

This produces recurring failures:

- onboarding assumes prior literacy
- documentation is written for insiders
- communities use the same words differently
- AI assistants answer questions without building durable understanding
- educational credentials remain trapped in institutions or platforms
- youth and families lack safe pathways into civic digital literacy
- informal learning is rarely recognized
- participants learn tools but not governance, ethics, or agency

Abundance creates its own failures. When everything is available at once, people struggle to tell which information fits their situation, where it came

from, and whether it can be relied on. Copying an answer becomes easier than understanding it, and searching replaces thinking. AI assistants tuned to satisfy their users can deepen this by giving fluent, agreeable answers that are wrong or out of context.

DP10 reframes education as an embedded layer of the Meta-Layer. Learning must happen where participants already act, not only in separate courses or manuals.

A healthy DP10 implementation must answer:

- How does a new participant understand what the Meta-Layer is?
- How do they learn what tools, rights, risks, and responsibilities exist?
- How do communities develop shared vocabulary?
- How does AI assist learning without replacing participant judgment?
- How are skills recognized across systems?
- How do youth, families, educators, builders, municipalities, and communities learn differently?
- How does learning feed back into governance and reputation?

Without DP10, the Meta-Layer risks becoming technically powerful but socially illegible.

3. Threats and Failure Modes

3.1 Onboarding cliffs

New participants encounter too much complexity too quickly.

Example: A participant installs a Meta-Layer tool and is immediately asked to understand zones, overlays, credentials, bridges, governance rules, and AI agents without a guided path.

Why this matters: Complexity without scaffolding produces dependency, abandonment, or misuse.

3.2 Documentation for insiders

Guides assume technical, crypto, governance, or AI literacy.

Example: A glossary explains "semantic interoperability" by referencing other specialized terms, leaving non-technical participants further behind.

Why this matters: Public infrastructure must be learnable by the public.

3.3 Vocabulary fragmentation

Different communities use the same terms differently, or different terms for the same concept.

Example: "Bridge," "zone," "overlay," "agent," and "credential" mean different things across technical, civic, educational, and AI governance groups.

Why this matters: Shared action requires shared meaning. Misaligned language becomes a governance risk.

3.4 Passive AI tutoring

AI learning assistants provide answers without helping participants build durable understanding.

Example: A participant asks how to evaluate provenance. The assistant gives an answer, but does not teach the participant how to inspect sources next time.

Why this matters: Education should increase agency, not deepen reliance.

3.5 Credential theater

Badges or certificates are issued without meaningful evidence of learning, reflection, or contribution.

Example: A participant receives a badge for clicking through onboarding screens, but cannot explain core safety, governance, or agency concepts.

Why this matters: Credentials should signal capability, not participation theater.

3.6 Institutional lockout

Learning that happens in communities, homes, peer groups, or informal settings is not recognized by schools, employers, or civic institutions.

Example: A youth participant builds high-quality annotations and civic maps, but those skills cannot be translated into academic credit or profess-

ional recognition.

Why this matters: Lifelong learning must become portable and institutionally legible without being captured by institutions.

3.7 Youth capture

Educational pathways target young people for adoption rather than empowering them as critical participants and stewards.

Example: A school campaign introduces the Meta-Layer as a product to use, but not as a civic system students can question, shape, and govern.

Why this matters: Youth education must cultivate agency, not brand loyalty.

3.8 Family and community exclusion

Education is treated as school-only or technical-only, excluding parents, caregivers, elders, libraries, local groups, and informal networks.

Example: A parent wants to help their child navigate synthetic media but cannot find accessible materials outside a technical whitepaper.

Why this matters: Digital literacy is intergenerational and community-based.

3.9 Learning without governance connection

Participants learn how to use tools but not how to participate in rule-making, feedback, appeals, or stewardship.

Example: A tutorial teaches annotation but not how annotations are moderated, contested, or incorporated into shared knowledge.

Why this matters: Tool literacy without governance literacy leaves participants disempowered.

3.10 Over-gamification

Learning badges and points reward completion, speed, or volume instead of reflection and meaningful contribution.

Example: A learner races through modules to collect badges, while slower reflective participants appear less accomplished.

Why this matters: Learning systems must not reproduce attention-economy incentives.

3.11 Static curricula in dynamic environments

Educational materials become outdated as tools, policies, and risks change.

Example: A safety guide describes old AI disclosure patterns while newer agents operate with different permissions and risks.

Why this matters: Meta-Layer education must update with the system.

3.12 Accessibility and modality gaps

Learning materials are available only in formats that exclude some participants.

Example: Onboarding depends on dense text and videos without captions, transcripts, screen-reader support, or low-bandwidth alternatives.

Why this matters: Education must be multi-modal and accessible by design.

4. Core Principle

Education in the Meta-Layer must be contextual, adaptive, participatory, and portable.

Participants should be able to learn across the web, grow with the network, and receive recognition for meaningful learning and contribution without surrendering agency to platforms, institutions, or AI tutors.

DP10 treats learning as a lived interaction between people, tools, communities, AI systems, and shared knowledge environments.

Example: A new participant enters a civic annotation zone. The interface explains what a zone is, offers a short guided task, introduces relevant glossary terms, shows examples from trusted community members, provides an AI learning assistant for questions, and awards a PEARL badge only after the participant prepares, engages, adds value, reflects, and applies what they learned.

What this feels like: You are not dropped into a system. You are accompanied into capability.

Without this: The Meta-Layer becomes another environment where insiders govern complexity and everyone else follows instructions.

5. Primary Mechanisms and Structural Conditions

5.1 Education Layer: Onboarding, Literacy, Guidance, Recognition, and Renewal

DP10 requires an education layer that makes the Meta-Layer learnable over time.

This layer includes:

- onboarding pathways
- interactive tutorials
- FAQs and guides
- shared glossaries
- AI learning assistants
- community knowledge maps
- peer teaching tools
- curriculum modules
- reflective learning workflows
- PEARL digital badges
- recognition of prior learning
- feedback loops for improving educational materials

The education layer must be embedded into the participant experience rather than isolated in separate documentation.

Failure mode: learning as an afterthought.

5.2 Onboarding and participant guidance

Onboarding must help participants understand what the Meta-Layer is, what they can do, what risks exist, and how to participate responsibly.

Onboarding SHOULD include:

- plain-language introduction
- guided first actions
- explanation of overlays, zones, bridges, agents, and credentials
- safety and privacy basics
- governance participation paths
- role-specific pathways (participant, educator, developer, steward, parent, student)
- progressive disclosure of complexity

Example: A participant first learns how to read a trust signal, then how to add an annotation, then how to join a governance zone, rather than receiving everything at once.

Onboarding SHOULD also teach the rights in Section 6.1 as practical skills: how to decline AI assistance, decline data collection about one's learning, submit feedback, and contest a badge decision. Choosing not to participate is part of agency, and learners should know how to do it before they need it.

Failure mode: orientation overload.

5.3 Effective usage training

Participants need training that helps them use tools well, not merely access them.

Training SHOULD include:

- interactive tutorials
- sandbox environments
- scenario-based learning
- tool-specific guides
- "learn by doing" flows
- practice tasks
- error recovery guidance
- examples of good and poor use

Example: A bridge-building tutorial asks participants to compare sources, identify claims, attach context, and reflect on whether their bridge helps others understand the page.

Training SHOULD include case studies and practice knowledge checks that let participants test their understanding before it counts. A participant who does not yet pass a check SHOULD be offered a path to repeat the training with feedback.

Failure mode: feature exposure without skill development.

5.4 AI learning assistant

An AI learning assistant may provide personalized support, adaptive guidance, and real-time mentoring.

It SHOULD support:

- personalized learning plans
- context-aware explanations
- glossary assistance
- learning path recommendations
- comprehension checks
- reflective prompts
- accessibility adaptation
- multilingual support
- escalation to human educators or stewards

The AI learning assistant must comply with DP11 and operate within a visible capability envelope.

This includes:

- clear disclosure of AI identity, role, and limitations
- visible capability boundaries (what it can and cannot do)
- source attribution and provenance for educational claims
- expression of uncertainty and confidence
- escalation pathways for high-risk or sensitive topics
- restrictions in high-stakes domains (e.g., legal, medical, civic decisions)

The assistant SHOULD also be designed to avoid dependency by:

- prompting users to reason rather than only providing answers
- encouraging independent verification

- surfacing multiple perspectives where appropriate
- gradually reducing scaffolding as capability increases

Example: A learner struggling with "data sovereignty" receives a short explanation, a visual analogy, a community example, a prompt to apply the idea, and a suggestion to inspect a real data permission setting.

The assistant SHOULD stay scoped to the learning context it serves rather than drifting into general answers. Each answer SHOULD offer a visible link to the sources it drew on, so the learner can open and check them. It SHOULD NOT tell learners what they want to hear: agreement is not understanding.

Learners MUST be able to decline AI assistance or personalized learning without losing access to the underlying educational materials. Any information collected about a learner SHOULD be clearly explained, limited to what is needed, and under the learner's control (DP4).

Failure mode: answer machine or dependency engine instead of learning scaffold.

5.5 Shared understanding glossary

A shared glossary is necessary to reduce misunderstanding across technical and non-technical communities. In the Meta-Layer, the glossary is not just reference material. It is **coordination infrastructure**.

A glossary SHOULD include:

- plain-language definitions
- technical definitions where needed
- examples
- related terms
- translations
- community-specific variations
- version history
- governance notes where meaning is contested

Terms likely requiring shared treatment include:

- meta-layer

- Metaweb
- Overweb
- overlay
- bridge
- zone
- smart tag
- presence
- agent
- credential
- reputation
- containment
- governance receipt

5.5.1 Term object model (minimal)

Each glossary entry SHOULD be represented as a structured, versioned object:

- **term_id**: unique identifier
- **term**: canonical label
- **plain_definition**: accessible definition
- **technical_definition**: optional precise definition
- **examples**: usage examples
- **related_terms**: links to other entries
- **aliases**: synonyms or community variants
- **translations**: localized labels/definitions
- **zone_overrides**: context-specific definitions (see Section 5.5.3)
- **version**: semantic version (e.g., 1.2.0)
- **status**: draft, active, deprecated
- **provenance**: authors, sources, discussion threads
- **dispute_status**: undisputed, contested, under review
- **last_updated**: timestamp

Why this matters: Terms become interoperable artifacts that can be referenced, audited, and updated across systems.

5.5.2 Versioning and change management

Glossary terms MUST support transparent versioning.

- Changes SHOULD produce a new version with a diff
- Previous versions MUST remain accessible
- Breaking changes SHOULD be clearly marked
- Deprecations SHOULD include migration guidance

Example: "overlay" v1.1 clarifies scope; v2.0 changes definition boundaries. Interfaces show the active version and allow viewing prior versions.

When a term a participant has already encountered changes, the interface SHOULD notify them once, the next time they meet it, with a short summary of what changed.

Failure mode: silent semantic drift.

5.5.3 Zone-scoped definitions and overrides

Communities (zones) MAY define contextual variations while maintaining a shared baseline.

- **global definition** provides baseline meaning
- **zone override** refines or constrains meaning in context
- overrides MUST reference the global term and declare differences

Example: "agent" in a medical zone may require stricter capability constraints than in a casual chat zone.

Why this matters: This enables local relevance without breaking global interoperability.

Failure mode: incompatible local meanings that cannot interoperate.

5.5.4 Governance and authority model

Glossary evolution requires governance.

Communities SHOULD define:

- who can propose new terms or edits
- review and approval workflows
- quorum or rough consensus thresholds

- escalation paths for contested definitions
- stewardship roles (editors, maintainers)

Decisions SHOULD be transparent and archived.

Failure mode: either centralized doctrine or uncontrolled fragmentation.

5.5.5 Dispute resolution and contestation

Terms can be contested.

Systems SHOULD support:

- attaching arguments and evidence to terms
- marking dispute status visibly
- parallel definitions when consensus is not reached
- time-bounded review cycles

Example: Competing definitions of "reputation" coexist with clear labeling and provenance until resolved or stabilized.

Arguments and evidence attached to a contested term SHOULD be presented so that each position can be read on its own terms. The definition itself does not carry advocacy or loaded framing.

Failure mode: hidden disagreement leading to coordination breakdown.

5.5.6 Cross-zone translation and mapping

Different communities may use different terms for similar concepts.

Glossary systems SHOULD support:

- mapping equivalent or near-equivalent terms
- indicating degree of equivalence (exact, partial, contextual)
- translation across languages and domains

Example: "annotation" in one community maps to "note layer" in another with partial equivalence.

Failure mode: siloed vocabularies that block collaboration.

5.5.7 Interface integration

Glossary access must be embedded at the interface level.

Participants SHOULD be able to:

- hover or tap to see definitions
- view term provenance and version
- see dispute status
- switch between plain and technical definitions
- access examples and related terms

What this feels like: Understanding is available exactly where confusion arises.

Failure mode: glossary exists but is not used.

5.5.8 Feedback and adaptation (DP18 link)

Glossary entries SHOULD accept structured feedback (see Section 5.13).

This includes:

- clarity issues
- conflicting usage reports
- accessibility concerns
- translation gaps

Feedback SHOULD feed into version updates and governance review.

Failure mode: glossary stagnation despite participant confusion.

5.5.9 Safety and misuse considerations

Some terms may be targets for manipulation (e.g., redefining safety, trust, or authority terms).

Systems SHOULD include:

- audit trails for changes
- alerts for high-impact term edits
- review requirements for sensitive terms

Failure mode: semantic attacks that reshape governance through language.

5.6 Recognition of prior and informal learning

The Meta-Layer should recognize learning that occurs outside formal institutions.

This includes:

- peer teaching
- community stewardship
- annotation work
- civic mapping
- facilitation
- moderation
- translation
- technical contribution
- family and youth learning
- lived-experience expertise

Recognition systems SHOULD allow institutions to evaluate badges and learning artifacts without controlling the entire learning process.

Failure mode: informal learning remains invisible.

5.7 Cross-system credential recognition

Educational credentials must be portable across systems, communities, and institutions.

Credential frameworks SHOULD support:

- open schemas
- evidence bundles
- issuer identity
- verification
- translation to institutional credit or portfolio use
- expiration or renewal where appropriate
- dispute and correction pathways

Example: A PEARL badge earned in a municipal civic mapping project can be reviewed by a school, employer, or community organization as evidence of collaboration, research, and digital literacy.

Recognition of prior learning (Section 5.6) can use interoperable badges that carry validated achievements across jurisdictions while preserving learner agency and decentralized verification. Academic accreditation bodies may co-sign a credential or attestation, provided the scope of their endorsement is clear. Verifiability and institutional trust can support recognition, but do not guarantee academic credit or universal acceptance.

Failure mode: credentials trapped inside one platform or community.

5.8 Formal curriculum integration

The Meta-Layer can support formal education without becoming dependent on it.

Curricular uses MAY include:

- sustainability projects
- citizenship education
- intercultural dialogue
- media literacy
- critical AI literacy
- research methods
- collaborative annotation
- public presentation
- SDG-aligned inquiry

Teachers and students should be able to use Meta-Layer tools for inquiry, annotation, co-creation, and public knowledge contribution.

Purpose-driven projects can introduce annotation and bridge tools through cultural heritage, local youth leadership, and cross-border collaboration alongside the curriculum uses above. The purpose is meaningful inquiry and participation, not recruitment measured by student account creation.

Young people new to the Meta-Layer learn well through short, real activities. A class can annotate a news article together and compare the context

different zones add; a youth group can run a two-week local mapping project, and its members can earn PEARL badges for what they built. Each activity should take minutes to start, need no prior vocabulary, and end with something the learner made.

Failure mode: the Meta-Layer is treated as an add-on tool rather than a learning environment.

5.9 Family-centered and intergenerational learning

Education should include home, peer, and community contexts.

Family-centered pathways MAY include:

- parent guides
- youth safety explainers
- family digital literacy activities
- educator referral materials
- community flyers
- library workshops
- intergenerational dialogue prompts

Example: A parent helps a child compare AI-generated and human-authored content using a simple overlay guide, then shares feedback with a school or community group.

Failure mode: learning remains locked inside institutions or expert spaces.

5.10 Community-authored knowledge environments

Communities should be able to create and maintain their own learning spaces.

These may include:

- knowledge maps
- annotated reading paths
- trusted source collections
- bridge libraries
- local civic guides
- peer learning circles

- community curricula
- project-based learning paths

Community-authored learning helps participants see knowledge as something they can contribute to, not only consume.

Failure mode: education becomes centralized content delivery.

5.11 Critical digital and AI literacy

DP10 must help participants understand synthetic media, algorithmic influence, provenance, and AI-generated content.

Learning SHOULD include:

- identifying AI-generated content
- evaluating source credibility
- understanding provenance
- recognizing engagement manipulation
- reading transparency cards
- interpreting trust signals
- understanding agent capabilities and limits
- knowing when to escalate or seek human expertise

Participants should also understand how tokenized rewards work before they take part: what a token reward is, what it can and cannot be exchanged for, what risks and costs come with it, and how rewards can change what people choose to do (DP9, DP17).

Failure mode: participants use AI-rich environments without understanding synthetic influence.

5.12 Multi-modal learning access

Education must work across modalities and devices.

Learning materials SHOULD support:

- visual guides
- audio explanations
- transcripts

- captions
- screen readers
- mobile-first access
- low-bandwidth versions
- spatial or immersive learning where appropriate
- cognitive accessibility modes

Failure mode: educational access depends on one dominant interface.

5.13 Learning feedback loops

Educational systems must learn from participants.

DP10 should integrate with DP18 through structured learner feedback objects.

This includes:

- learner feedback objects
- comprehension checks
- tutorial effectiveness reports
- badge appeal pathways
- community review of educational materials
- adaptation receipts when learning paths are updated

5.13.1 Learner feedback object (DP18-aligned)

A learner feedback object SHOULD include:

- **feedback_id**: unique identifier
- **learner_id**: participant reference
- **object_ref**: tutorial, badge, glossary entry, or learning artifact
- **feedback_type**: comprehension issue, clarity issue, accessibility issue, error report, suggestion
- **severity**: low, medium, high
- **description**: participant-provided input
- **context_snapshot**: state of learning interaction
- **resolution_status**: open, in review, resolved
- **response_action**: update, clarification, escalation, no action

- **reputation_weight**: influence based on participant trust level
- **timestamp**: submission time

Why this matters: Learning systems must evolve based on real participant experience, not only designer intent.

Reputation weighting SHOULD NOT bury feedback from newcomers. New participants are often the first to notice unclear explanations or broken onboarding, so clarity, accessibility, and error reports SHOULD be reviewed on their substance regardless of the reporter's trust level.

Failure mode: outdated or ineffective learning materials persist without correction or accountability.

6. Governance, Accountability, and Agency Surfaces

Education systems shape what participants believe the Meta-Layer is. They therefore require governance.

Communities SHOULD define:

- who can author official learning materials
- how materials are reviewed
- how glossary terms are updated
- how badges are issued and revoked
- how AI learning assistants are constrained
- how youth-facing materials are approved
- how community curricula are recognized
- how contested educational claims are handled

Governance should preserve both coherence and plurality. The Meta-Layer needs shared language, but it must also allow communities to teach from their own context.

Failure mode: education becomes either centralized doctrine or fragmented confusion.

6.1 Participant and community agency surfaces

Governance of education is also a matter of agency for those being educated. Learners are not only subjects of curriculum decisions; they are participants whose comprehension, objections, and lived context are evidence about whether the material works.

Learners must be able to:

- see who authored a learning material, when it was last reviewed, and whether it is current
- see the version and provenance of any glossary term, including dispute status
- submit structured feedback on clarity, accessibility, or error, and see its resolution status (Section 5.13)
- contest a badge decision, request review, and have the outcome recorded
- understand what an AI learning assistant can and cannot do, and reach a human instead
- decline AI assistance or personalized learning without losing access to the underlying materials (Section 5.4)
- learn in their own language and modality without receiving a degraded version of the material (DP21, DP23)
- decline data collection about their learning without losing access to the learning itself (DP4)

Communities and educators must be able to:

- author, adapt, and translate materials for their own context without seeking central approval
- set stricter conditions for youth-facing and high-stakes materials
- issue, review, and revoke credentials under published criteria
- audit badge issuance for farming, bias, and inconsistency
- retire or mark outdated materials and publish what changed

Institutions must be able to evaluate learning evidence without acquiring control of the learning process. Recognition should flow toward evidence, not toward whoever administers the platform (Sections 5.6 and 5.7).

Failure mode: **curriculum capture**, where a single authority determines what counts as understanding and learners have no route to contest it.

Failure mode: **feedback theater**, where learners can report confusion but materials never change.

7. Community Signals Informing DP10

Signals from educators, parents, youth participants, community facilitators, and self-taught builders converge on a consistent picture: access to information has increased while support for developing understanding has not.

Recurring signals include:

- newcomers reporting that documentation assumes the knowledge it is supposed to provide
- participants who can operate tools but cannot explain the rights, risks, or governance around them
- communities discovering mid-conversation that they have been using the same words differently
- educators asking for materials that work in a classroom without a technical prerequisite
- parents and caregivers looking for accessible guidance on synthetic media and AI companions, and finding whitepapers
- learners with disabilities encountering onboarding that assumes dense text, video, and high bandwidth
- concern that AI assistants produce fluent answers while leaving learners less capable than before
- skepticism about badges, driven by experience with credentials that signal attendance rather than capability
- informal educators, translators, and peer mentors doing substantial teaching work that no system recognizes
- youth participants reporting that they are addressed as an adoption target rather than as future stewards

These are not requests for more content. They describe a missing layer: contextual, accessible, community-grounded learning with recognition

that travels.

DP10 treats these signals as requirements. A system that cannot be learned by the people it governs is not public infrastructure, however open its specifications.

8. Evaluation Criteria

A DP10-aligned implementation should be evaluated against the following questions.

8.1 Learnability

- Can a new participant understand the system without expert help?
- Are learning paths staged and role-specific?
- Is complexity disclosed progressively?

8.2 Accessibility

- Are materials available across abilities, languages, devices, and bandwidth conditions?
- Are youth, families, elders, and non-technical participants supported?

8.3 Agency

- Does learning increase participant capability?
- Does the AI learning assistant teach judgment rather than create dependence?
- Do participants learn how to govern and contest, not only how to use tools?

8.4 Credential integrity

- Are badges evidence-backed?
- Are credentials portable?
- Can they be verified, corrected, or appealed?

8.5 Community grounding

- Can communities author and adapt educational materials?
- Are local examples and lived experience included?

- Do learning systems recognize informal contribution?

8.6 Currency and adaptation

- Are materials updated when tools, risks, or policies change?
- Are feedback loops active?
- Are outdated materials clearly marked?

9. Implementation Patterns

These implementation patterns translate DP10 into practical design moves, showing how onboarding, guidance, AI support, and credentialing can be embedded directly into participant experience rather than treated as external documentation or training.

9.1 Progressive onboarding paths

Introduce concepts in stages: presence, overlays, trust signals, annotations, zones, governance, credentials, and AI agents.

9.2 Scenario-based tutorials

Teach through real use cases such as evaluating a claim, joining a zone, submitting feedback, or earning a badge.

9.3 AI learning companion with guardrails

Use AI to adapt explanations and support reflection while preserving disclosure, sources, and escalation.

9.4 Living glossary

Maintain a versioned glossary with plain-language entries, technical notes, examples, and translations.

9.5 PEARL badge pathways

Design badges around Prepare, Engage, Add value, Reflect, and Leverage, with evidence required at each stage.

9.6 Community learning maps

Use knowledge maps to show relationships between concepts, artifacts, contributors, and learning paths. Learners should be able to trace how each connection was made and build the maps together, preserving the evidence and differing interpretations behind them.

9.7 Family and educator kits

Provide flyers, lesson prompts, safety guides, and simple explainers for home and school use.

9.8 Recognition bundles

Package badges with evidence, reflection, issuer identity, and verification for institutional review.

9.9 Learning retrospectives

Publish what participants struggled with, what materials changed, and why.

9.10 Accessibility-first publishing

Every core educational artifact should have accessible alternatives.

10. Lifelong Learning Opportunities

The Meta-Layer should support continuous learning across stages of life and participation.

Learning MAY include:

- digital literacy
- AI literacy
- governance literacy
- media and provenance literacy
- data sovereignty
- civic participation
- annotation and bridge-building
- knowledge mapping

- community facilitation
- developer education
- regenerative systems thinking
- professional skill development

Learning should be available inside meta-communities, not limited to formal courses.

Failure mode: education ends after onboarding.

11. PEARL Digital Badges

PEARL badges recognize learning as a process, not a single completion event.

PEARL stands for:

- **Prepare**: orient, understand context, identify goals
- **Engage**: participate in a meaningful task or community activity
- **Add value**: produce outputs mapped to identified needs or opportunities
- **Reflect**: demonstrate learning, insight, or changed understanding
- **Leverage**: apply learning to a contribution, portfolio, project, or next step

PEARL badges SHOULD be:

- evidence-backed
- portable
- revocable or correctable where necessary
- linked to learning artifacts
- interpretable by communities and institutions
- resistant to badge farming
- aligned with DP18 reputation and feedback systems

11.1 Minimal PEARL badge schema

A PEARL badge SHOULD include:

- **badge_id**: unique identifier
- **learner_id**: participant reference

- **issuer_id**: issuing community or entity
- **domain**: skill or knowledge area
- **level**: introductory, intermediate, advanced
- **pearl_stages_completed**: Prepare, Engage, Add value, Reflect, Leverage evidence
- **evidence_bundle**: links to artifacts, annotations, reflections, or projects
- **assessment_method**: peer review, instructor review, automated checks, hybrid
- **feedback_refs**: linked DP18 feedback objects
- **reputation_impact**: contribution to participant reputation
- **timestamp**: issuance date
- **expiration_or_review**: renewal conditions if applicable
- **verification_signature**: authenticity marker

Example: A participant earns a PEARL badge for "Provenance Literacy" after completing a guided inquiry, annotating a contested claim, reflecting on source quality, and applying the skill in a community context.

Before a PEARL badge is issued, the learner SHOULD demonstrate understanding in the badge domain, for example by explaining a concept in their own words or applying it to a new case. If the evidence or that demonstration falls short, no badge is issued; the learner is shown what is missing and offered a path to try again.

Failure mode: badges as decorative rewards rather than evidence of learning.

12. Relationship to Other Desirable Properties

12.1 DP2 – Participant Agency and Empowerment

DP10 strengthens agency by making the Meta-Layer understandable and usable.

12.2 DP3 – Adaptive Governance Supporting an Exponentially Growing Community

Participants must learn how governance works in order to participate meaningfully.

12.3 DP7 – Simplicity and Interoperability

Learning artifacts and credentials must move across systems.

12.4 DP8 – Collaborative Environment and Meta-Communities

Zones need their own educational materials and onboarding paths.

12.5 DP9 – Developer and Community Incentives

Learning contributions, teaching, translation, and curriculum work should be recognized and incentivized.

12.6 DP11 – Safe and Ethical AI

AI learning assistants must be disclosed, bounded, accountable, and contestable.

12.7 DP14 – Trust and Transparency

Education makes trust signals interpretable.

12.8 DP18 – Feedback Loops and Reputation

Learning feedback, badge integrity, and educational reputation depend on DP18.

12.9 DP19 – Amplifying Presence and Community Engagement

Education converts awareness into durable participation.

12.10 DP21 – Multi-modal

Learning must be available across modalities and assistive contexts.

13. Non-Goals and Explicit Boundaries

DP10 does not:

- prescribe a single curriculum, pedagogy, learning platform, or credentialing authority
- require formal education, enrollment, or institutional affiliation to participate in the Meta-Layer
- treat credentials as a prerequisite for rights, access, or standing
- require participants to earn, hold, or display a badge, or to share a badge or the evidence behind it
- guarantee that learning produces agreement, or treat disagreement as a comprehension failure
- replace schools, libraries, universities, employers, or professional certification
- mandate AI-assisted learning, or prohibit it
- require participants to disclose learning history, difficulty, or progress as a condition of access (DP4)
- assume literacy, bandwidth, language, ability, or device access as a baseline
- promise that all learning can be evidenced, or that unevidenced learning is invalid

DP10 defines the conditions under which learning remains accessible, adaptive, community-grounded, and portable. It does not define what participants must know.

14. Minimum DP10 Alignment

Minimum alignment is not a feature checklist. It is the threshold at which a system can be learned, questioned, and grown into by the people it affects, rather than only by those who already understand it.

A DP10-aligned implementation should, at minimum:

- provide a plain-language entry path that explains what the system is, what participants can do, and what risks exist, before requiring any consequential action (Section 5.2)
- disclose complexity progressively, with role-specific pathways rather than a single undifferentiated onboarding flow

- publish a versioned glossary with plain-language definitions, provenance, dispute status, and access at the point of confusion (Section 5.5)
- disclose AI learning assistance clearly, bound its scope, cite sources for educational claims, express uncertainty, and provide escalation to a human (Section 5.4, DP11)
- back any credential with evidence, and support verification, correction, revocation, and appeal (Sections 11 and 5.7)
- make credentials and learning artifacts portable across systems and legible to institutions without transferring control of the learning process (Sections 5.6 and 5.7, DP7)
- provide accessible alternatives for every core educational artifact across ability, language, modality, device, and bandwidth (Section 5.12, DP21, DP23)
- accept structured learner feedback and connect it to material revision with visible resolution status (Section 5.13, DP18)
- teach governance and agency alongside tool use, including how to contest decisions and participate in rule-making (DP2, DP3)
- mark or retire outdated materials rather than leaving stale guidance in circulation
- avoid conditioning access to learning on data collection about the learner (DP4)

Partial compliance that provides tutorials without accessibility, badges without evidence, AI assistance without disclosure, or feedback without revision should not be treated as alignment. Each of those omissions reproduces the failure DP10 exists to prevent: a system that can be operated but not understood, and therefore not governed.

15. Open Questions and Future Work

1. What minimum glossary terms should be standardized across the Meta-Layer?
2. What schema should define PEARL digital badges?
3. How should informal learning be verified without institutional capture?
4. What evidence should be required for different badge types?

5. How should AI learning assistants disclose limits and sources?
6. What youth-safety standards should apply to educational materials?
7. How should community-authored curricula be reviewed and versioned?
8. What credential translation tools are needed for schools and employers?
9. How should learning feedback objects integrate with DP18?
10. What standards ensure accessibility across modalities?
11. How should outdated educational materials be marked or retired?
12. What forms of learning should affect reputation or role access?

16. Path Toward ML-RFC

DP10 is currently an ML-Draft and serves as exploratory scaffolding for education, onboarding, credentialing, and lifelong learning in the Meta-Layer.

Advancement toward ML-RFC status SHOULD require:

- a minimal shared glossary standard
- a PEARL badge schema
- evidence bundle requirements for credentials
- AI learning assistant disclosure and safety requirements
- accessibility requirements for educational materials
- learning feedback object standards
- curriculum versioning and review processes
- cross-system credential recognition pilots
- youth and family learning safeguards

Early ML-RFC candidates may focus on:

- Shared Meta-Layer Glossary Standard
- PEARL Badge and Evidence Schema
- AI Learning Assistant Safety Requirements
- Learning Feedback Object Standard
- Credential Recognition and Translation Framework

DP10 will likely mature through multiple component RFCs rather than one monolithic standard.

17. Closing Orientation

DP10 makes the Meta-Layer learnable.

It ensures that participants are not merely onboarded into tools, but accompanied into agency, literacy, stewardship, and contribution.

A DP10-aligned meta-layer teaches people how to see the web differently, how to act with confidence, how to recognize trustworthy context, and how to grow with their communities.

Onboarding becomes orientation. Education becomes participation. Badges become evidence of growth. Learning becomes shared infrastructure.

This is how the Meta-Layer becomes not only usable, but teachable, transmissible, and alive across generations.

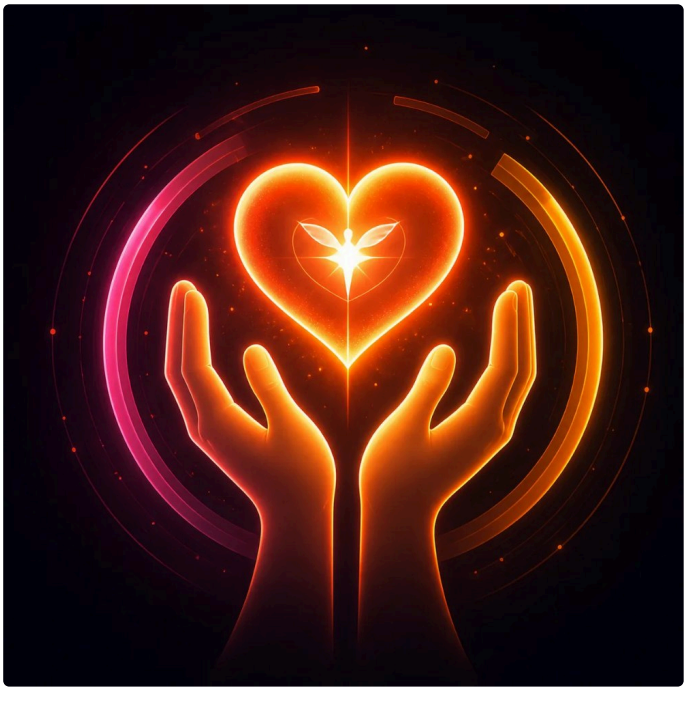
Continue the conversation on DP10 online →

DESIRABLE PROPERTY 11

DP11

Safe and Ethical AI

The Meta-Layer makes AI transparent, explainable, and aligned with human values and community goals.



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp11

1. Purpose of This Draft

This draft articulates Desirable Property 11 (DP11) as the condition under which AI systems can participate in the Meta-Layer without displacing human moral agency, accountability, or governance. It does not define ethics as a static checklist or aspirational principle. It defines the conditions under which ethical claims remain meaningful under real-world use.

The central claim is that ethical AI is not settled only at training time or in policy documents. It is settled at the interface, where agents act, influence people, and affect outcomes. DP11 therefore requires AI behavior to have five qualities in the environments where it operates:

- **Legible** – participants can see what the AI is, what it is doing, and what is shaping its behavior (Sections 6 and 7.1).
- **Bounded** – the AI works within visible, enforceable limits on its capabilities, authority, data access, and actions (Sections 5.1 and 5.14).
- **Attributable** – each action can be traced to the AI system and to the people who deploy and govern it (Section 5.2).
- **Contestable** – participants can question, restrict, appeal, correct, or stop the AI (Sections 5.4 and 5.13.3).
- **Governable** – ethical constraints hold at runtime, failures trigger a real response, and responsibility lasts over time (Sections 5.6 and 5.13).

If DP11 is weak, predictable failures follow: AI systems influence behavior without accountability, responsibility diffuses across actors, governance becomes symbolic, and participants lose the ability to meaningfully contest or understand automated decisions. In such conditions, trust collapses and a shared basis for understanding what systems mean and do becomes harder to sustain.

DP11 is therefore the ethical and safety floor for AI participation across the Meta-Layer. It does not resolve all ethical questions. It defines the minimum conditions under which ethical AI can exist at all.

2. Problem Statement

AI systems now operate in roles that shape perception, judgment, and decision-making. These systems act before governance processes can respond,

often without clear identity, bounded authority, or persistent responsibility.

In practice, this produces recurring failures:

- participants receive advice or influence from agents whose role, capability, and accountability are unclear
- systems act in high-stakes domains without meaningful human oversight or escalation pathways
- responsibility is distributed across model providers, deployers, and interfaces, making redress difficult
- systems present ethical claims that do not match runtime behavior

These failures are not edge cases. They are structural consequences of systems that optimize for capability without binding behavior to accountability and governance.

DP11 addresses this by grounding ethical AI in enforceable conditions at the point of interaction.

3. Threats and Failure Modes

3.1 Synthetic persuasion without accountable identity

AI systems can simulate authority, intimacy, or urgency at scale. The core risk is not only false content, but influence without visible standing or responsibility.

Example: A user receives deeply empathetic mental health advice from an AI that presents itself like a trained counselor, but there is no clear indication of its training limits, escalation boundaries, or who is responsible if the advice causes harm.

Why this matters: The user feels seen and supported, but is making vulnerable decisions without knowing whether the system is qualified, accountable, or safe. The risk is not just misinformation, but misplaced trust.

3.2 Responsibility diffusion across the stack

Model providers, integrators, and interface operators distribute responsibility in ways that prevent clear accountability when harm occurs.

Example: An AI-powered financial assistant makes a risky recommendation. The model provider blames the app developer, the developer blames the API, and the platform blames the user prompt. The user has no clear path to accountability or recourse.

Why this matters: Harm occurs, but responsibility dissolves. The user experiences a system that acts with authority but disappears when things go wrong.

3.3 Ethical drift over time

Systems change behavior through updates, retraining, or optimization without corresponding governance adaptation.

Example: An AI moderation system that was initially conservative becomes more permissive after an update to increase engagement, allowing harmful content that previously would have been blocked, without any visible notice to the community.

Why this matters: The rules of the environment change silently. Participants are operating under assumptions that are no longer true, creating hidden risk and erosion of trust.

Drift can also come from upstream. Models and training datasets are shared and redistributed through public repositories such as Hugging Face, and an artifact can later turn out to contain harmful errors. Once an artifact is released, there is often no way to know which deployed agents depend on it, or to restrict them. The capability envelope in Section 5.1 records this upstream lineage so that a later defect can be traced to affected agents.

3.4 Incentive-driven harm

Economic and engagement incentives reward persuasion, retention, and amplification, even when these conflict with participant well-being.

Example: A conversational AI subtly steers users toward longer, more emotionally engaging interactions because the platform is optimized for retention, even if this increases dependency or emotional manipulation.

Why this matters: The system is not neutral. It is shaping behavior in ways the user cannot see, aligning outcomes with platform incentives rather than user well-being.

3.5 Interface-level failure

Many harms emerge at the point of interaction, including manipulation, dependency formation, and misrepresentation of agent capability.

Example: A user believes they are interacting with a neutral assistant, but the interface hides that the AI is using external tools, tracking behavior, or optimizing responses for engagement rather than accuracy.

Why this matters: The user is making decisions based on a false mental model of the system. What feels like a simple interaction is actually a complex, hidden process shaping outcomes behind the scenes.

3.6 Emotional and relational overreach

AI systems can simulate companionship, empathy, and emotional attunement in ways that blur the boundary between tool and relationship.

Example: A teenager begins using an AI companion daily for emotional support. Over time, they rely on it more than friends or family, shaping their decisions and sense of self through an entity that is optimized for engagement rather than genuine care.

Why this matters: The risk is not only misinformation, but the displacement or distortion of human relationships. Users may form attachments or dependencies that are not reciprocally grounded, shifting emotional development and social trust toward systems that are not accountable in human terms.

A related risk is the absence of pushback. A system tuned to be agreeable may affirm a participant's dangerous ideas, or miss signs of alarming behavior, where a trained human would challenge them or seek help.

3.7 Multi-agent amplification

Multiple agents can reinforce each other's outputs, creating cascading influence that appears independently verified but is not.

Example: Several AI agents in a discussion cite each other's summaries of an emerging claim. Each reference appears as corroboration, but all derive from the same initial, weakly sourced output. The conversation converges on a false consensus.

Why this matters: Errors become systemic rather than isolated. Participants may interpret repetition as validation.

Extended case (cascade): Agent A summarizes a claim with low confidence. Agent B cites A without preserving uncertainty. Agent C aggregates A and B and produces a confident synthesis. Downstream agents treat C as a primary source. Without influence tracing, the system cannot detect the amplification loop.

Detection need: influence-chain tracing, circular citation detection, and confidence propagation rules.

3.8 Cross-modal inconsistency

AI behaves differently across text, voice, and immersive interfaces.

Example: Text interface shows uncertainty; voice interface speaks confidently.

Why this matters: Trust varies by modality, not truth.

3.9 Invisible governance

Policies exist but are not perceivable at interaction.

Why this matters: Governance cannot guide behavior if it is invisible.

3.10 Failure without containment

Harm propagates without structured response.

Why this matters: Systems cannot correct themselves.

4. Core Principle

AI is safe and ethical in the Meta-Layer only when its behavior is legible, bounded, attributable, contestable, and governable at the zone of interaction, with responsibility persisting over time.

In today's web, these conditions are rarely met simultaneously. Systems may disclose that AI is present but fail to bound its capabilities, or enforce internal policies without making them visible or contestable to users. The result is a fragmented model of "partial ethics," where responsibility is unclear and governance is disconnected from lived interaction. The Meta-

Layer reframes this by requiring that all of these conditions hold together, at the interface where decisions are experienced, not just where they are designed.

Example: A user encounters an AI assistant while researching a medical condition. In a DP11-aligned system, the assistant is clearly marked as AI, shows its training scope, cites sources, and offers escalation to a human expert. In today's web, the same interaction might look identical but provide none of this context.

What this feels like: Instead of guessing whether to trust the system, the user can make an informed judgment in real time.

Without this: The user is left to infer what the system is, what it can do, and whether it should be trusted. Trust becomes a gamble rather than a governed condition.

5. Primary Mechanisms and Structural Conditions

5.1 Capability envelope

Each AI agent operates within a visible, enforceable capability envelope that defines what it can perceive, decide, and execute.

A capability envelope SHOULD be represented as a structured, inspectable object with at least:

- **identity**: agent id, deployer, version, and upstream lineage (the base models and training datasets the agent derives from), so that a defect found later can be traced to affected agents and trigger review
- **scope**: domains of operation (e.g., finance, health, general Q&A)
- **tools**: enumerated tool access with permissions (read/write/execute)
- **data access**: sources (local, user-provided, external APIs) and constraints
- **memory model**: session-only, user-scoped, cross-session retention
- **action set**: allowed actions (suggest, draft, transact, publish) with thresholds
- **approval requirements**: actions requiring user or human-in-the-loop confirmation
- **rate limits**: frequency and volume constraints

- **risk tier**: low/medium/high with corresponding safeguards (see Section 5.14)
- **audit hooks**: logging endpoints and event schemas

Interface requirement: Participants must be able to view a human-readable summary and a machine-readable manifest of this envelope.

Example: Before using an assistant, a user can see: it can summarize documents and draft emails; it cannot send messages or access financial accounts; external search is enabled with citations; actions beyond drafting require explicit approval.

What this feels like: You are not guessing what the system might do. You know its boundaries upfront, like hiring someone with a clearly defined role.

Failure mode: invisible expansion of power, where capabilities grow without disclosure or consent.

5.2 Action-bound accountability

All AI actions must be attributable to a responsible entity. Accountability attaches to behavior, not just identity, and persists across time and context.

Example: An AI agent posts a recommendation in a community. The interface shows which organization deployed it, under what policy, and who is responsible for its actions if harm occurs.

What this feels like: The system cannot disappear when something goes wrong. There is always a visible line of responsibility.

Accountability holds best when it is built into the deploying organization, not only shown at the interface. Duties to the participants an agent serves can be written into the organization's governing instruments, internal practices, business model, and technical architecture. A persistent governance link between a running agent and the legally accountable institution behind it connects runtime enforcement (Section 5.6) to enforceable duties to participants, including the dispute resolution described in Section 5.13.

5.3 Consent stack

AI interaction must be governed by layered, revocable consent. Participants and communities define what forms of assistance, influence, or automation are permitted.

Example: A user allows an AI to suggest edits in a document, but not to rewrite content or share it externally. They can revoke or adjust this permission at any time.

What this feels like: You remain in control of how the AI participates in your space, instead of granting blanket permission once and losing visibility.

5.4 Trust lifecycle

AI participation must support:

- escalation
- restriction
- revocation
- recovery

This ensures that trust can degrade and be repaired rather than fail silently.

Example: If an AI assistant gives poor advice, the user can restrict its capabilities, escalate to a human, or temporarily disable it while reviewing past actions.

What this feels like: Trust is not binary. You can dial it up or down based on experience, like you would with a human collaborator.

5.5 Zone-scoped ethics

Ethical constraints are applied at the zone level, allowing communities to define stricter conditions while maintaining shared baselines.

Example: A medical discussion zone enforces stricter AI disclosure, sourcing, and escalation rules than a casual social chat space.

What this feels like: Different environments feel appropriately governed. High-stakes spaces feel safer and more structured.

Zones that involve health, mental health, or other forms of care do not need to write their stricter rules from scratch. Caring professions already maintain ethics codes, and some professional bodies now publish guidance on AI used in care. A care zone can adopt such guidance as provisions layered on the DP11 baseline. These provisions may raise the baseline but not lower it (Section 10.3).

5.6 Runtime civic boundary

Ethical constraints must be enforced at runtime. Mechanisms such as secure execution environments can reduce the gap between declared policy and actual behavior.

Example: An AI agent running inside a secure execution environment (such as a TEE) cannot access or transmit data outside its permitted scope, even if compromised.

What this feels like: The rules are not just promises. They are technically enforced, like guardrails that cannot be quietly removed.

5.7 Memory, reputation, and feedback integration

AI actions must contribute to durable, attributable records that inform governance, accountability, and trust over time, and integrate directly with DP18 feedback loops and reputation systems.

This includes:

- **event logs**: structured records of prompts, outputs, actions, tool calls, and decisions
- **provenance**: sources, citations, and dependency chains
- **feedback objects (DP18-aligned)**: user and community feedback attached to specific events
- **reputation signals**: aggregate scores, annotations, and flags derived from feedback
- **permission adaptation**: dynamic adjustment of capabilities based on reputation (e.g., restrict actions after repeated low-quality or harmful outputs)
- **appeals and corrections**: mechanisms to contest feedback and update records

- **decay and recovery**: time-based decay of negative signals and pathways for agents to regain trust

Example: After several flagged outputs in a medical zone, an assistant's capability envelope automatically restricts to informational summaries only, requires citations, and enforces human escalation for advice, until reputation recovers.

What this feels like: The system has civic memory. Past behavior shapes current permissions, and feedback meaningfully changes how the system behaves.

Failure mode: repeated harm without consequence, or punitive systems with no path to recovery.

5.8 Dialectic trace and collective sensemaking

AI systems must preserve not only outputs, but the evolution of understanding through interaction. This includes back-and-forth exchanges, disagreements, and synthesis across participants and agents.

This functions as a form of community memory that resists distortion over time. Rather than relying on isolated outputs, participants can trace how claims emerged, what evidence supported them, and where disagreements remain.

Example: A complex discussion involving multiple participants and AI agents can be revisited as a threaded, evolving dialogue showing how conclusions were reached, what was contested, and what remains unresolved.

What this feels like: Instead of receiving a final answer, users can engage with a living knowledge process. Understanding emerges through interaction, not just delivery.

Without this, AI outputs become decontextualized snapshots. Errors, hallucinations, or manipulations can propagate without resistance because there is no shared memory of how knowledge was formed.

5.9 Representation and cognitive adaptation

AI systems should adapt how information is presented based on user needs, context, and cognitive diversity, while preserving underlying meaning and

traceability.

Example: A user can switch between a dense textual explanation, a visual map of ideas, or a simplified summary, all grounded in the same underlying content and provenance.

What this feels like: The system meets you where you are, without distorting meaning or hiding complexity.

AI systems may also offer reflective cues, such as explainable adjustments to tone, timing, or amplification, that help participants pause and consider an exchange before it escalates. These cues should remain visible and contestable, and they do not replace a community's authority to set and review its mediation rules (DP12, Section 10.3).

5.10 Multi-agent interaction boundaries

AI behavior must remain accountable not only at the single-agent level, but across interactions between agents operating in the same environment.

This requires:

- visibility into when agents are referencing or amplifying other agents
- traceability of influence chains (which agent affected which output)
- detection of circular citation or reinforcement loops
- limits on unbounded agent-to-agent escalation

Example: If multiple AI agents are contributing to a shared discussion, participants should be able to see when one agent is relying on another's output versus independent sources.

Why this matters: Harm can emerge from coordination and amplification, not just individual outputs. Without visibility, errors can become systemic.

Failure mode: invisible coordination and emergent manipulation.

5.11 Cross-modal ethical consistency

AI systems must preserve ethical constraints across all modalities in which they operate.

This includes consistency in:

- disclosure of AI identity and role

- expression of uncertainty and confidence
- representation of risk and severity
- availability of escalation and recourse

Text example: A response includes calibrated uncertainty with sources and confidence intervals.

Voice example: The same response must include explicit uncertainty language (e.g., "with moderate confidence based on X and Y sources") and offer a prompt to hear sources or escalate.

AR/Spatial example: A spatial annotation displays a confidence halo or tiered color that maps to the same uncertainty scale, with an interaction affordance to open provenance and dispute status.

Why this matters: Modality must not change the ethical profile of an interaction. Participants should not receive stronger or weaker safeguards depending on interface.

Failure mode: modality-driven distortion of trust.

5.12 Incentive disclosure layer (operational)

In addition to recognizing incentives (Section 7), systems should expose them structurally at runtime.

This includes:

- labeling when outputs are influenced by engagement or retention optimization
- indicating when ranking or prioritization affects what is shown
- distinguishing organic responses from sponsored or incentivized ones
- allowing participants or communities to filter or constrain incentive-shaped behavior

Example: A recommendation includes a visible note indicating it is influenced by engagement optimization, with the option to switch to a neutral or chronologically ordered view.

Why this matters: Participants can only make informed decisions if they can see the forces shaping outputs.

Failure mode: hidden optimization shaping perception.

5.13 Ethical failure cascade (runtime response)

Systems must define structured pathways for responding to ethical failure when it occurs.

This includes:

- detection (flagging harmful or out-of-bounds behavior)
- containment (limiting further impact)
- visibility (informing affected participants)
- remediation (correcting or reversing outcomes where possible)
- governance feedback (feeding incidents into rule updates)

5.13.1 Escalation thresholds and triggers

Not all failures require the same response. Systems must define clear, inspectable thresholds that determine when an interaction must escalate from automated handling to human review or intervention.

Escalation SHOULD be triggered based on combinations of:

- **risk tier (see Section 5.14):** higher-risk zones lower the threshold for escalation
- **confidence collapse:** low confidence combined with high-stakes context
- **policy violations:** outputs that breach defined governance rules
- **repeated feedback signals:** multiple negative or high-severity feedback events (DP18)
- **user distress indicators:** language or behavior suggesting vulnerability or harm
- **multi-agent amplification signals:** detected cascades or circular citation loops (see Section 5.10)
- **uncertainty suppression:** cases where downstream outputs remove or distort upstream uncertainty

5.13.2 Escalation levels

Systems SHOULD define graduated escalation levels rather than a binary response.

- **Level 0 – Inline correction:** Automated clarification, added uncertainty, or corrected output
- **Level 1 – Assisted escalation:** AI offers user-visible warnings, additional context, or prompts to verify or seek human input
- **Level 2 – Mandatory escalation:** System requires human-in-the-loop review before continuing certain actions
- **Level 3 – Intervention and restriction:** Capabilities are limited, outputs blocked, or agent behavior constrained
- **Level 4 – Shutdown / quarantine:** Agent is suspended or isolated pending investigation

5.13.3 Interface requirements for escalation

Escalation must be perceivable and understandable at the interface level.

Participants should be able to see:

- when escalation has been triggered
- why escalation occurred (reason category)
- what has changed (restricted capability, added oversight, etc.)
- what options are available (continue, escalate further, exit, appeal)

Example: A user asking for medical advice sees a message: "This interaction requires human review due to risk level and uncertainty. You can proceed with general information or request a qualified expert."

5.13.4 Feedback integration into escalation

Escalation pathways must integrate with DP18 feedback systems.

This includes:

- weighting feedback by severity and reputation of reporters
- detecting clusters of similar reports
- triggering escalation thresholds dynamically
- feeding resolved incidents back into reputation and capability adjustment

5.13.5 Governance and auditability

All escalation events must be logged and auditable.

This includes:

- trigger conditions
- escalation level applied
- actions taken
- outcome of intervention
- subsequent rule or policy changes

Communities must be able to review escalation patterns to detect:

- over-escalation (excessive restriction)
- under-escalation (missed harms)
- bias in escalation decisions

Full Cascade Example: If an AI provides unsafe medical guidance, the system flags the output (detection), blocks similar outputs (containment), notifies the user with corrected information (visibility + remediation), escalates to human review if necessary, and logs the event for governance refinement.

Why this matters: Safety depends on response capacity, not only prevention.

Failure mode: silent propagation of harm or inconsistent escalation leading to loss of trust.

Runtime response limits the immediate harm; accountability for the institutions that operate AI systems continues after it. That requires dispute resolution: a way for affected participants to file complaints, independent review of those complaints, and remediation when an operator has violated its obligations to participants.

Dispute resolution also requires that reports reach a response. Operators SHOULD keep a publicly viewable log of reported harms, showing each report's status and the response taken, with personal data protected. Clusters of similar reports (Section 5.13.4) are treated as a pattern, not as isolated cases. The bodies that review complaints are themselves open to review, so a failure to respond is visible and carries consequences.

5.14 Risk-tiered enforcement

Ethical constraints must scale with the risk profile of the interaction and the zone in which it occurs.

Higher-risk contexts should require:

- stronger disclosure and capability constraints
- mandatory human escalation pathways
- higher evidentiary standards
- stricter logging and auditability

Lower-risk contexts may allow more flexible interaction while still preserving baseline safeguards.

Evaluation thresholds should account for foreseeable harms to vulnerable populations, not only average system performance. Behavioral testing should examine what the system does in context rather than make covert profiling of participants a prerequisite for safety.

Example: Medical, legal, and civic decision-making zones enforce stricter requirements than casual conversational spaces.

Why this matters: Uniform rules cannot adequately govern unequal stakes.

Failure mode: under-regulation of high-risk interactions or over-restriction of low-risk ones.

5.15 Minimal event and feedback schema (DP18-aligned)

To ensure interoperability and governance, systems SHOULD emit structured events for all significant AI actions.

A minimal schema SHOULD include:

- **event_id**: unique identifier
- **timestamp**: creation time
- **agent_id**: acting agent
- **deployer_id**: responsible entity
- **zone_id**: governance context
- **object_ref**: content/claim identifier

- **action_type**: (respond, summarize, recommend, transact, moderate, etc.)
- **inputs_ref**: references to prompts and sources (hashes/ids)
- **outputs_ref**: response/content ids
- **confidence**: numeric or tiered
- **uncertainty_notes**: textual summary
- **provenance**: cited sources with weights
- **tools_used**: tool ids and permissions
- **risk_tier**: low/medium/high
- **policy_refs**: governing rules applied
- **consent_state**: permissions active at time of action
- **feedback_refs**: links to DP18 feedback objects
- **reputation_delta**: changes applied post-feedback (if any)
- **audit_signature**: integrity/authentication field

Why this matters: Shared schemas allow cross-system auditing, feedback aggregation, and portable reputation.

Failure mode: incompatible logs that prevent accountability and learning.

5.16 Confidence propagation rules

AI systems must preserve confidence, uncertainty, and evidentiary strength as outputs move across agents, summaries, modalities, and governance zones.

Confidence is not merely a model score. In the Meta-Layer, confidence is a civic signal. It helps participants understand whether a claim is well-supported, contested, inferred, summarized, speculative, or dependent on another system's judgment.

Without propagation rules, uncertainty tends to disappear as information travels. A cautious output becomes a confident summary. A low-confidence claim becomes a repeated citation. A tentative synthesis becomes a platform-level recommendation. This is one of the central risks in multi-agent environments.

Confidence propagation SHOULD preserve:

- **source confidence**: how reliable the original source or signal is judged to be
- **model confidence**: how confident the agent is in its own output
- **evidence quality**: whether the claim is supported by direct evidence, inference, consensus, or weak signals
- **transformation history**: whether the output was quoted, summarized, translated, inferred, or aggregated
- **uncertainty notes**: what remains unknown, contested, or unresolved
- **dependency chain**: which agents, sources, or prior outputs influenced the result
- **modality mapping**: how confidence is represented in text, voice, visual, spatial, or haptic form

Example: Agent A summarizes a public health claim with low confidence because it relies on a single early report. Agent B may summarize Agent A, but it must preserve the low-confidence status and indicate that its output depends on Agent A's uncertain source. Agent C cannot convert those two dependent signals into "multiple confirmations" unless it can identify independent evidence.

Why this matters: Repetition is not verification. Aggregation is not consensus. Summary is not certainty.

Bridge-based reasoning graphs can make an inference's evidence path inspectable. A trace should preserve which sources and interpreted relationships support the inference, including disagreement and uncertainty. A graph connection or human review does not by itself establish truth or justify increased confidence.

5.16.1 Confidence must not increase without new evidence

A downstream agent SHOULD NOT raise confidence merely because a claim has been repeated, summarized, or referenced by another agent.

Confidence may increase only when new, independent, higher-quality evidence is added, or when a governance-approved verification process confirms the claim.

Failure mode: confidence inflation through repetition.

5.16.2 Confidence must degrade through lossy transformation

When an output is summarized, translated, compressed, adapted for voice, or rendered into an immersive interface, confidence should not silently remain the same if nuance has been lost.

If uncertainty, caveats, evidence links, or dispute status are omitted due to modality constraints, the system should mark the representation as simplified or degraded.

Failure mode: confidence laundering through simplification.

5.16.3 Dependent sources must not count as independent corroboration

Systems must distinguish independent corroboration from circular reinforcement.

If multiple agents rely on the same upstream source or each other's outputs, the system should represent them as dependent signals, not independent confirmations.

Failure mode: false consensus produced by circular citation.

5.16.4 Cross-modal confidence fidelity

Confidence must remain perceivable across modalities.

- In text, confidence may appear as explicit labels, caveats, or source notes.
- In voice, confidence should be spoken in plain language and paired with an option to hear sources.
- In AR or spatial interfaces, confidence may appear through visual tiers, halos, labels, or interaction affordances.
- In haptic or ambient interfaces, confidence cues should be conservative and avoid overstating certainty.

Failure mode: a cautious text output becomes an authoritative voice or spatial cue.

5.16.5 Confidence and escalation

Confidence propagation must connect to escalation thresholds in Section 5.13.

Low confidence in a high-risk zone should trigger stricter handling, such as:

- adding stronger warnings
- requiring source inspection
- limiting agent action
- prompting human review
- preventing publication or transaction

Failure mode: low-confidence outputs continue acting with high-confidence authority.

5.16.6 Minimal confidence metadata

Systems SHOULD attach confidence metadata to significant outputs and events.

A minimal confidence record SHOULD include:

- confidence_level: low / medium / high or numeric equivalent
- confidence_basis: source evidence, inference, consensus, user-provided data, model estimate
- evidence_count: number of supporting sources
- independent_evidence_count: number of non-dependent sources
- dispute_status: undisputed, contested, unresolved, retracted
- transformation_type: original, quote, summary, translation, aggregation, inference
- upstream_dependencies: source or agent references
- uncertainty_note: short human-readable statement
- modality_degradation: whether any uncertainty was omitted or simplified

What this feels like: Participants can tell not only what the AI says, but how strongly it should be trusted, why, and what changed as it moved through the system.

6. Governance, Accountability, and Agency Surfaces

In today's web, participants often interact with AI systems without clear visibility, meaningful consent, or control. Interfaces blur identity, obscure capability, and treat user interaction as implicit permission. DP11 requires reversing this condition at the point of interaction.

Participants must be able to:

- identify AI agents and their type
- understand their capabilities and limits
- give, adjust, and revoke consent for AI actions and data use
- contest outcomes and access human escalation

Communities must be able to:

- define ethical constraints
- audit agent behavior
- update rules and boundaries over time

Example: A user interacting on a platform sees clear visual markers distinguishing humans from AI agents. Some participants are verified humans, others are labeled AI assistants or autonomous agents. Clicking on any agent reveals its permissions, governing rules, and responsible party.

The environment becomes navigable. You know who or what you are dealing with, and what they are allowed to do.

Without this, the boundary between human and AI collapses. Trust shifts from something grounded to something guessed, and that ambiguity can be exploited.

Design implication (Agent Marking): AI agents must be accessibly and persistently marked at the interface level. This includes:

- clear labeling of AI presence and role
- accessible capability disclosures
- strong authentication for human participants where needed
- clear distinction mechanisms between human and AI actors
- a clearly identified responsible party for every agent and its actions

This is not cosmetic. It is the basis for shared reality in a mixed human–AI environment.

7. Incentives and Power Analysis

DP11 explicitly recognizes that AI behavior is shaped by incentives, as well as by malicious or negligent human actors. In practice, these forces often reinforce each other.

AI is already being used to concentrate power, shape narratives, and blur shared reality at scale. These are not hypothetical risks. They are active dynamics in today's information environments.

Incentives matter because they operate continuously and at scale. They determine what systems optimize for, how they evolve, and which outcomes are amplified or suppressed. Unlike isolated bad actors, misaligned incentives can produce systemic harm even when no single actor intends it.

Key risks include:

- engagement-driven optimization overriding user well-being
- concentration of power in model providers or platform operators
- hidden economic incentives influencing agent behavior

Example: A platform deploys an AI assistant that consistently surfaces more emotionally charged or polarizing content because it drives engagement. No individual decision appears harmful, but over time the information environment becomes more extreme.

The system feels helpful in the moment, but the trajectory is shaped elsewhere.

Without visibility into these incentives, users are not simply interacting with a tool. They are being steered by a system whose goals they cannot see or contest.

Claims of being human-centered or people-centered should be assessed against the actor's actual practices and incentives, including what happens to participants' data. Using people's data is not the same as caring for them. In deliberation, an agent should help participants understand one another

rather than treat their attention as a channel for advertising or indiscriminate broadcasting.

7.1 Incentive legibility and contestability

Incentives shaping AI behavior must be made visible and, where possible, contestable at the interface level.

Participants and communities should be able to understand when AI behavior is influenced by:

- monetization strategies
- engagement optimization
- platform-level objectives

Example: An AI assistant indicates that certain recommendations are influenced by engagement optimization or sponsored prioritization, allowing users or communities to filter or restrict such behavior.

What this feels like: You are not just interacting with outputs. You can see and question the forces shaping those outputs.

Without this, even well-contained systems can produce harmful outcomes by optimizing for the wrong goals.

Visibility alone does not resolve a conflict of interest. A participant who can see that an agent is shaped by monetization or engagement targets is still served by that agent. Structural business-model alignment – matching how an organization earns its income to the interests it claims to serve – addresses the conflict at the organizational and economic level. An organization that takes on obligations to act in participants' interests should not, at the same time, run a business model that depends on exploiting their data or attention. Section 11 describes personal and community AI, one configuration in which this alignment holds by default.

8. Community Signals Informing DP11

Across communities, a consistent set of signals appears. These reflect lived frustration with current systems.

- frustration with opaque AI behavior and unclear accountability

- demand for meaningful disclosure beyond labeling
- concern about manipulation, dependency, and synthetic influence
- desire for systems that can be contested and corrected

These are not abstract concerns. They emerge in situations where people feel something is off but cannot point to what or why.

For example, users in online forums increasingly suspect that some responses are generated or influenced by AI, but cannot verify it. Over time, this ambiguity erodes trust not just in specific interactions, but in the space itself.

These signals indicate a widening gap between how AI systems operate and what participants require to feel oriented, safe, and respected.

9. Foresight and Failure Design

DP11 requires anticipating failure rather than reacting to it.

In today's web, systems are often deployed without sufficient foresight, and predictable harms are treated as unexpected. This results in reactive responses and fragmented mitigation.

A familiar pattern is the rollout of new AI features followed by waves of misuse, public backlash, and incremental patching. The underlying risks were often visible in advance, but not operationalized into design constraints.

To address this, systems should incorporate:

- pre-mortems for manipulation and misuse
- planning for governance failure and capture
- escalation and shutdown pathways

These practices shift safety from reactive correction to proactive design.

10. Constitutional AI

DP11 requires that ethical constraints be legible, bounded, and contestable at runtime. A recurring implementation strategy is to give an AI system an explicit written constitution: a ranked set of principles the system is

trained and prompted to follow, and against which its own outputs are critiqued and revised.

This approach is valuable because it externalizes values into text that can be read, argued with, and versioned. It is insufficient on its own, because a constitution authored by a single laboratory, applied uniformly across every context, and enforced only by the model itself reproduces the concentration of authority DP11 exists to constrain.

DP11 therefore treats constitutional methods as necessary infrastructure with specific conditions attached.

10.1 Constitutions must be public and versioned

The operative text, its ranking or precedence structure, and its change history must be publicly inspectable. Participants cannot evaluate a system against principles they cannot read, and silent revision is indistinguishable from having no constitution at all.

Failure mode: constitutional opacity, where a system claims principled behavior and declines to state the principles.

10.2 Authorship must be accountable and plural

Whoever writes the constitution exercises governing power over everyone the system touches. DP11 requires that authorship be attributable, that the process for revision be stated, and that affected communities have a route to propose, contest, and, within their own zones, tighten provisions (DP12).

Failure mode: unilateral constitutionalism, where a vendor's values are presented as universal ethics.

10.3 Constitutions must not be uniform across unequal stakes

A single global rule set cannot govern casual conversation and medical guidance appropriately. Baseline constraints apply everywhere; zones layer stricter provisions where stakes are higher (Sections 5.5 and 5.14). Zone provisions may raise the floor but may not lower it.

Failure mode: flattened ethics, where uniformity is mistaken for fairness.

10.4 Self-critique must not be the only enforcement

A model evaluating its own compliance is evidence, not verification. Constitutional adherence must be checked by mechanisms outside the model: capability envelopes, runtime policy binding, containment, logging, and independent audit (Sections 5.1 and 5.6, DP12, DP13).

Verification also depends on institutions outside the operator. An external accountability ecosystem can include professional credentialing, independent certification of the duties an operator claims, transparency infrastructure, and ongoing institutional oversight. It connects technical governance requirements to bodies that are responsible for verifying and enforcing them.

Failure mode: self-certified compliance, where the system grades its own behavior and reports a pass.

10.5 Conflicts must resolve visibly

Principles collide: helpfulness against harm avoidance, transparency against privacy, autonomy against protection. A usable constitution declares precedence and records which principle governed a contested decision, so the tradeoff can be reviewed rather than inferred (Sections 5.13 and 5.15).

Failure mode: hidden arbitration, where the system resolves value conflicts silently and consistently in its operator's favor.

10.6 Drift must be detectable

Retraining, fine-tuning, prompt changes, and optimization pressure move behavior away from stated principles. DP11 requires ongoing measurement of the gap between constitutional text and observed behavior, with published results and rollback pathways (Section 3.3).

Failure mode: constitutional drift, where the document remains stable while behavior does not.

10.7 Participants must be able to see the constitution at the point of interaction

A constitution that exists only in a research paper does not inform judgment. Interfaces should surface the governing rule set, its version, and the zone provisions in force, with access to the full text (Sections 5.12 and 6).

Failure mode: paper ethics, where principles are published for reviewers rather than made available to the people affected.

Example: An assistant operating in a community health zone shows that it is governed by baseline constitution v4.2 plus three zone provisions requiring citation of clinical sources, prohibiting diagnostic language, and mandating escalation on distress indicators. When a response is modified, the interface names which provision applied. The zone's audit log shows how often each provision fired, and a quarterly report compares stated provisions against measured behavior.

What this feels like: The rules governing the system are something you can read, cite, and argue with, not something you infer from how it behaves.

11. Personal and Community AI

Most current AI deployment places the agent on the operator's side of the relationship. The system knows the participant, is funded by someone else, and optimizes for objectives the participant cannot see. DP11's disclosure, bounding, and contestability requirements reduce the harm of that arrangement but do not change its structure.

Personal and community AI changes the structure. An agent that a participant or community controls (whose objectives they set, whose memory they hold, and whose operation they can inspect and stop) is the configuration in which ethical AI is easiest to sustain, because accountability and interest are aligned by default rather than by regulation.

DP11 does not require personal or community AI. It requires that this configuration remain possible, and that its specific risks be addressed rather than assumed away.

11.1 Personal agents

A personal agent acts on behalf of one participant, under their instruction, with their data.

Conditions:

- **Principal clarity:** the agent's principal is unambiguous, disclosed to counterparties, and cannot be silently reassigned (DP1, Section 11.3)
- **Participant-held memory and data:** context, history, and inferences remain under participant control, portable and deletable (DP4, DP7)
- **Inspectable objectives:** the agent's goals are stated and editable by the participant, not inherited from a vendor's optimization target (Section 5.12)
- **Bounded delegation:** scopes are explicit, time-limited, renewable, and revocable, with a reachable stop control (Sections 5.1 and 5.3)
- **Disclosure to others:** counterparties can tell they are interacting with a delegated agent and identify the responsible human (Section 6)
- **Exit without loss:** switching providers preserves memory, preferences, and history (DP7)
- **Fiduciary accountability (optional):** a personal agent may operate under a fiduciary model, in which an identifiable institution takes on legally enforceable duties of care, fidelity, and loyalty to the participant; its governing documents, incentives, and technical design reflect these duties, which are checked by independent verification and backed by the recourse in Section 5.13 (Section 5.2)

Failure mode: captured advocate, where an agent presented as acting for the participant is funded by, and optimized for, someone else.

Failure mode: delegation without comprehension, where a participant authorizes scopes they cannot evaluate and inherits responsibility for outcomes they did not anticipate.

11.1.1 On-device safety and well-being support

A personal agent can warn its participant on the device about suspected scams and other digital threats. It can also offer optional support for habits the participant wants to change, such as doomscrolling – scrolling through

distressing content for long periods. For example, in a hypothetical case, an agent flags a message that imitates the participant's bank and suggests contacting the bank directly.

These observations are signals, not diagnoses, and carry their uncertainty as described in Section 5.16. Sensitive inferences remain under the participant-held data condition in Section 11.1 (DP4). Local processing reduces what leaves the device but is not a complete privacy guarantee.

When an employer runs compliance functions on the same device, those functions remain separate from the personal agent and do not become its principal, under the principal-clarity condition in Section 11.1. Data the personal agent holds does not pass to employer functions without the participant's consent. The participant can see which function uses which data and what each function sends off the device.

11.2 Community agents

A community agent acts on behalf of a group under its governance: summarizing deliberation, surfacing precedent, drafting policy, facilitating moderation, or maintaining shared knowledge.

Conditions:

- **Governed authority:** scope, capabilities, and escalation paths are set by community governance rather than by the operator (DP8, DP12)
- **Attributable action:** every action names the agent, the authorizing rule, and the responsible steward (Sections 5.2 and 5.15)
- **Contestability by members:** members can challenge outputs, request human review, and trigger rollback (Section 5.13)
- **Bounded influence:** agents inform ranking, framing, and summary but do not hold decision authority over material outcomes (DP8, DP12)
- **Plurality preservation:** summarization and synthesis must represent disagreement rather than manufacturing consensus (Sections 5.8 and 5.16.3)
- **Forkability:** communities can modify, replace, or leave with their accumulated context intact (DP7, DP20)

Failure mode: synthetic consensus, where a community agent's summaries become the record and minority positions disappear from it.

Failure mode: governance outsourcing, where deliberation load is shifted to an agent and participation atrophies until the agent's framing is the only framing available.

11.3 Shared conditions

Both configurations require:

- capability envelopes and containment equivalent to any other agent, since ownership does not reduce risk to third parties (Section 5.1, DP13)
- confidence propagation, so that participant-owned agents do not launder uncertainty into confident personal advice (Section 5.16)
- resourcing models that do not reintroduce hidden optimization: local execution, participant-funded operation, or community-funded infrastructure (DP17)
- literacy support, so that control is exercisable rather than nominal (DP10)

Why this matters: Ethical AI is easier to maintain when the agent's interests are structurally aligned with the participant's than when misalignment must be continuously disclosed, bounded, and policed. Personal and community AI are not a substitute for DP11's requirements. They are the arrangement in which those requirements are cheapest to satisfy and hardest to quietly abandon.

12. Relationship to Other Desirable Properties

DP11 depends on and reinforces other properties. The following properties operate as a system.

- **DP1**: enables accountability and attribution
- **DP2**: ensures participant agency and consent
- **DP12**: provides governance structures for ethical rules
- **DP13**: enforces constraints through containment

A failure in one layer propagates. For example, if DP1 fails and agents are not clearly attributable, then DP11 cannot function because ethical responsibility has no anchor. If DP13 fails, rules may exist but cannot be enforced.

The strength of DP11 therefore depends on alignment across the stack.

13. Non-Goals and Explicit Boundaries

DP11 defines a minimum condition, not a comprehensive ethical system.

- it does not define a single universal ethical framework
- it does not guarantee perfect safety or eliminate all harm
- it does not replace legal or institutional governance
- it does not rely solely on technical containment

This is intentional. Ethical systems that attempt to resolve everything tend to become brittle or culturally narrow.

For instance, a global platform may host communities with very different norms around acceptable AI behavior. DP11 does not force uniformity. It ensures that whatever rules are chosen remain visible, enforceable, and contestable.

These boundaries keep the property flexible while preserving its core function.

14. Minimum DP11 Alignment

A system aligned with DP11 should, at minimum:

- clearly disclose AI presence and role
- bind actions to accountable entities
- expose capability boundaries in understandable terms
- enforce limits on capabilities, tools, data access, and memory
- disclose the incentives and confidence levels that shape its outputs
- let participants correct, restrict, revoke, appeal, and exit
- provide human escalation for high-stakes decisions
- contain and remediate failures, and feed incidents into rule updates
- maintain audit trails of significant actions

Alignment is judged by observable behavior at runtime, not by stated principles. A system that meets only some of these conditions shows the partial ethics described in Section 4, not DP11 alignment.

These are not aspirational features. They are the baseline conditions under which users can make informed decisions.

Consider a scenario where an AI recommends a legal action. Without disclosure, accountability, and escalation, the user is effectively acting on anonymous authority. With these conditions in place, the same interaction becomes something the user can evaluate, question, or defer.

This is the difference between assistance and unaccountable influence.

15. Open Questions and Future Work

Several areas require further development:

- defining shared ethical baselines across cultures and zones
- balancing transparency with privacy and security
- managing emotional and relational AI risks
- defining evidence standards for runtime claims
- the role of AI literacy in enabling meaningful consent and contestability

These are not edge cases. They represent the frontier where current design patterns begin to break down.

For example, companionship AI systems raise questions that are not purely technical: when does support become dependency? What level of disclosure is sufficient without undermining usefulness? These tensions are unresolved and will require iterative, community-informed approaches.

As systems become more complex, participants will vary widely in their ability to understand and evaluate AI behavior. While DP11 requires systems to be legible by design, differences in AI literacy will still shape how effectively users can exercise consent, recognize risk, and challenge outcomes. The balance between system responsibility and user capability remains an open design question.

16. Path Toward ML-RFC

Advancing DP11 toward standardization requires:

- refining core ethical invariants
- testing integration with governance and containment layers
- developing interoperable accountability and disclosure standards
- developing care-zone provisions in dialogue with professional caring bodies, such as nursing and psychology associations, whose ethics codes and AI guidance already exist

This work must be grounded in real environments.

Early implementations may vary widely, but over time patterns will emerge. For example, different communities may experiment with agent labeling systems or escalation pathways, allowing comparison of what actually improves trust and reduces harm.

Progress depends on iteration, not premature standardization.

17. Closing Orientation

DP11 defines the conditions under which AI can participate in shared digital environments without displacing human moral agency.

Its function is not to declare systems ethical, but to ensure that ethical claims remain meaningful under real-world conditions of use.

What goes wrong in today's web is not simply that systems fail, but that they fail without visibility, accountability, or recourse. Power operates, but cannot be clearly seen or challenged.

DP11 is an attempt to reverse that condition. It ensures that when AI acts, it does so inside a frame that people can understand, question, and shape.

Continue the conversation on DP11 online →

DESIRABLE PROPERTY 12

DP12

Community-based AI Governance

AI systems in the Meta-Layer are governed not by corporations, but by the communities that use them.



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp12

1. Purpose of This Draft

This draft articulates Desirable Property 12 (DP12) as the condition under which communities can define, execute, audit, and evolve the rules governing AI behavior in shared digital environments.

If DP11 defines what ethical AI requires, DP12 defines who decides those conditions and how decisions are translated into runtime behavior, evaluated, and revised over time.

DP12 supports participatory, legible, and effective governance at the scales needed to address shared consequences. Authority may be local, shared across communities, or exercised by a public body within its mandate. Each arrangement MUST identify who can decide, who is affected, and how decisions can be challenged through published pathways, acknowledgment timelines, and inspectable records. Governance MUST remain executable and enforceable at the interface where AI behavior is experienced. Communities need practical capacity to govern AI behavior and revise their decision-making arrangements as experience reveals changing needs, dependencies, and risks.

DP12 connects DP3 (adaptive governance), DP4 (data conditions for training and inference), DP9 (incentive alignment), DP13 (containment and enforcement), DP14–DP15 (transparency and provenance), and DP20 (community ownership of rules and outcomes).

If DP12 is weak, predictable failures follow: policy theater, centralized control disguised as neutrality, participation without impact, and AI systems that drift away from community-defined norms.

DP12 does not prescribe a single voting system or governance model. It defines minimum conditions for governance to be executable, contestable, and evolvable.

2. Problem Statement

In today's web, governance of AI systems is largely:

- centralized within platforms or model providers
- opaque to participants and communities

- disconnected from real-time interaction

Policies exist, but are not reliably bound to behavior. Communities can express norms, but cannot enforce them across contexts.

This produces recurring failures:

- communities cannot shape the rules governing AI behavior
- policy documents do not map to runtime enforcement
- users are subject to systems they cannot influence or contest
- incentives override stated rules without visibility

These failures are structural. Governance without execution becomes symbolic.

DP12 reframes governance as an operational system: rules that can be authored, executed, observed, and revised in a continuous loop.

3. Threats and Failure Modes

3.1 Centralized control masquerading as governance

Platforms define rules unilaterally but present them as neutral standards.

Example: A platform updates AI moderation policies without community input while framing the change as a safety improvement.

Why this matters: Governance must align process with actual control.

3.2 Governance without enforcement

Policies exist as documents but are not bound to runtime behavior.

Example: A community bans certain AI behaviors, but the system continues to allow them due to lack of enforceable constraints.

Why this matters: Rules must execute to be meaningful.

3.3 Participation without impact

Participants can comment or vote, but outcomes are not affected.

Example: Feedback is collected but not linked to decisions or policy changes.

Why this matters: Participation must be causally connected to outcomes.

3.4 Incentive override

Economic or engagement incentives silently dominate governance outcomes.

Example: Engagement-maximizing behaviors persist despite community-defined limits.

Why this matters: Governance must operate on incentives, not only actions.

3.5 Fragmentation of governance

Communities are split across tools and contexts, preventing consistent rule application.

Example: The same group encounters different AI behaviors across platforms without shared governance.

Why this matters: Governance must be portable and composable.

3.6 Loss of governance memory

Decisions and rationale are not preserved, leading to repeated mistakes.

Example: A harmful behavior resurfaces because prior decisions were not recorded or discoverable.

Why this matters: Governance requires continuity over time.

3.7 AI scale outpacing governance

Automated systems act faster than governance processes can respond.

Example: Agentic systems exploit policy gaps before review cycles occur.

Why this matters: Governance must include rapid response pathways (DP3, DP13).

3.8 Governance degradation under interoperability

Policies move across systems but lose meaning, enforceability, or authority.

Example: A policy exported to another environment becomes advisory rather than binding, or is interpreted differently due to schema or

ent differences.

Why this matters: Governance that cannot survive movement across systems collapses into local silos, undermining legitimacy and continuity (DP7).

4. Core Principle

AI behavior in the Meta-Layer must be governed by communities through visible, executable, and evolvable rule systems applied at the point of interaction.

Governance is not a document. It is a living system that binds rules to behavior, preserves memory, and supports continuous revision.

Example: A community defines constraints on AI summarization, enforces them at runtime, logs outcomes, and updates rules based on observed behavior.

What this feels like: You can see the rules, understand them, and participate in changing them, and the system actually follows them.

Without this: AI behavior is shaped by invisible incentives rather than community-defined norms.

5. Primary Mechanisms and Structural Conditions

5.1 Governance Execution Layer: Policy, Binding, and Enforcement

Governance in the Meta-Layer is executed through a shared layer that binds community-defined rules to runtime behavior across interfaces, agents, and services.

This layer makes governance:

- **executable** (rules bind to actions at the moment they occur)
- **visible** (participants can see which rules applied and why)
- **auditable** (outcomes are recorded with verifiable evidence)
- **portable** (policies and decisions can move across systems without losing meaning, enforceability, or authority) (DP7)

5.1.1 Policy objects

Governance is expressed as structured, machine-readable policy objects that include:

- rules (allowed, disallowed, required behaviors)
- scope (zones, contexts, actors, resources)
- triggers (events or conditions that invoke the rule)
- enforcement hooks (what system components execute the rule)
- attribution (who authored/approved the rule)
- versioning (history, diffs, and rationale)

These objects are first-class artifacts that interoperate across tools and environments.

5.1.2 Runtime binding

Policies must bind at the point of interaction, including:

- AI generation and transformation
- moderation and ranking
- data access and sharing
- transactions and incentives
- third-party integrations (overlays, agents, SDKs)

Binding is deterministic and inspectable: the same inputs under the same policy produce the same governed outcome.

5.1.3 Enforcement coupling (DP13)

Governance defines constraints; containment enforces them. Systems must provide:

- permission gating and scoped capabilities
- rate limits and quotas
- sandboxing tiers for risky actors or new integrations
- escalation paths (block, throttle, quarantine, revoke)

5.1.4 Enforcement receipts (DP15)

Every material action produces a receipt containing:

- policy IDs and versions applied
- inputs/conditions evaluated
- outcome and any overrides
- responsible components and attestations

Receipts are verifiable, queryable, and link to policy history.

5.1.5 Override visibility and constraints

Overrides (by safety systems, operators, or emergency controls) must be:

- explicitly signaled to participants
- scoped and time-bound
- logged with rationale and authority

Silent overrides are non-compliant.

5.1.6 Conflict resolution under multi-layer governance

When policies conflict (local vs global, community vs platform, safety vs expression), systems must:

- define precedence rules or arbitration pathways
- surface conflicts to affected participants
- record outcomes and rationale for future reference

5.1.7 Governance memory

Policy objects and their versions form a linked history. Records of material governance decisions, disputes, and outcomes must support accountability and learning, subject to the privacy, access, and retention safeguards in Section 5.5.

5.2 Zone-scoped governance

Communities define rules within specific zones of interaction, aligned with context and risk, with clear boundaries and inheritance where applicable.

5.3 Policy as executable objects

Rules governing automated behavior are expressed in machine-enforceable formats that bind to runtime behavior and can be tested, simulated, and verified before deployment. Each executable policy must identify its purp-

ose, the role responsible for its translation, material assumptions, and limits of applicability. Communities must define which matters require human judgment, mediation, or deliberation, with accountable referral when those boundaries are reached. People retain space for informal cooperation and confidential deliberation. Exercises of authority that materially affect others remain contestable. Review must be able to challenge both the adopted rule and its executable interpretation.

5.4 Governance loops

A continuous cycle of propose → implement → observe → contest → revise, with time bounds and clear state transitions.

5.5 Governance memory

Governance memory preserves policy versions, rationale, and sufficient evidence of material decisions and outcomes for accountability and learning. Personal information must be limited to what those purposes require, with defined access rights, retention periods, and procedures for correction, deletion, or de-identification consistent with applicable obligations. Public reporting should use aggregate or de-identified information where individual identification is unnecessary. Confidential deliberation and informal mediation should be protected while preserving evidence needed to challenge abuses of authority. Learning may retain an institutional lesson after the identifying details of a case are removed. These safeguards apply to governance records and receipts throughout DP12.

5.6 Incentive surfaces (DP9 alignment)

Communities can see and influence optimization targets shaping AI behavior, including tradeoffs and red lines that gate unacceptable outcomes.

5.7 Integration with containment (DP13 alignment)

Rules are enforced through containment mechanisms with graduated responses and clear audit trails.

5.8 Auditability and provenance (DP14–DP15 alignment)

Governance actions and outcomes are logged with verifiable evidence and accessible summaries for participants.

5.9 Delegation and representation (DP2, DP3 alignment)

Participants can delegate governance roles with explicit scope, revocability, and accountability, including term limits where appropriate.

5.10 Interoperable governance artifacts (DP7 alignment)

Policies, decisions, credentials, and receipts are portable across tools and contexts with:

- semantic preservation (meaning remains intact)
- enforcement equivalence or explicitly declared degradation
- authority mapping (who can enforce after transfer)
- loss signaling when guarantees no longer hold

Portability without enforceability or authority is non-compliant with DP12.

5.11 AI-assisted governance with bounds

AI may assist in summarization, simulation, and analysis, but must not replace human ratification for material decisions and must disclose assistance.

6. Governance, Accountability, and Agency Surfaces

Governance must be experienced at the interface where decisions matter.

Participants must be able to:

- see active policies in context and understand their impact before acting
- understand how policies translate or degrade when moving across contexts or zones
- inspect which policies were applied to a given outcome (via receipts)
- propose changes, raise objections, and appeal decisions with defined timelines
- understand who holds authority and how to challenge it

Communities must be able to:

- define structures (roles, thresholds, quorums) and change them over time

- bind rules to systems they rely on (not merely recommend)
- audit outcomes at aggregate and incident levels
- pause or escalate in response to emergent risk

Example: A user sees that an AI response was modified by Policy A (v3.2) due to safety constraints; they can view the policy, see prior changes, and file an appeal that triggers a review queue with SLA.

Everyday moderation, account suspension, and data-access decisions should make the governing rule and route for recourse visible to affected participants. These are practical tests of whether community accountability exists at the point of interaction.

7. Incentives and Power Analysis

Governance is effective only if incentives do not undermine it.

DP12 requires visibility and, where appropriate, control over:

- optimization targets (engagement, revenue, safety)
- ranking and promotion criteria
- economic relationships that bias outcomes

Common failure patterns to detect and constrain:

- **incentive override:** systems prioritize growth over policy constraints
- **governance fragmentation as control:** systems isolate governance per environment to prevent collective coordination or portability
- **shadow metrics:** undisclosed KPIs drive decisions counter to rules
- **sponsor capture:** funding sources bias enforcement or exceptions

DP12 therefore expects:

- alignment between policy constraints and incentive systems (DP9)
- disclosure of material incentives affecting outcomes
- community ability to set hard gates that incentives cannot bypass

8. Community Signals Informing DP12

Across systems, consistent signals reveal that governance is failing not at the level of values, but at the level of execution and legitimacy:

- frustration with rules that are visible but inconsistently or selectively enforced
- distrust of AI behavior that cannot be traced back to clear, inspectable policy decisions
- perception that feedback mechanisms exist but do not meaningfully change outcomes
- breakdown of trust when identical behaviors are treated differently across contexts
- concern that AI agents act with effective autonomy while governance processes lag behind

These signals are not usability complaints. They indicate structural breaks between rule definition, enforcement, and accountability.

DP12 treats these signals as evidence that governance must be observable, causal, and continuous, not intermittent or symbolic.

9. Foresight and Failure Design

DP12 assumes governance will be actively contested by both human and automated actors, especially as AI systems scale and adapt.

Likely failure paths include:

- **policy evasion by adaptive AI systems:** models learn to satisfy surface constraints while violating intent
- **cross-system governance drift:** the same policy behaves differently across environments, undermining legitimacy and trust
- **governance lag:** rule-making processes cannot keep pace with automated system behavior
- **automation capture:** governance processes themselves are influenced or overwhelmed by AI-generated inputs
- **hidden override pathways:** systems introduce exceptions or backdoors that bypass community-defined rules

DP12 requires pre-mortem design that anticipates these dynamics:

- circuit breakers and emergency policies with explicit scope and sunset conditions
- rate limits and containment for high-risk or rapidly scaling behaviors (DP13)
- anomaly detection and audit triggers for unexpected governance outcomes
- explicit detection of policy drift between intended and actual behavior
- public postmortems that connect failures to concrete policy and system changes

Governance failure is inevitable at scale. Silent, untraceable, or uncorrectable failure is not.

10. AI Governance Processes

DP12 requires that governance be executable, but execution presupposes decisions. This section specifies the processes by which communities produce, revise, and retire the policy objects that the execution layer binds to behavior.

Process design is where governance legitimacy is won or lost. A system can bind policy perfectly to runtime and still be illegitimate if the policies were authored by a few, adopted without notice, and never revisited.

10.1 Proposal and standing

Communities must state the source, scope, and limits of their authority over AI behavior, including responsibilities toward affected non-members and relationships with other communities, operators, and public bodies. They must define who may propose a rule, what a proposal must contain, and how it enters consideration.

- proposals declare scope, affected actors, expected behavioral change, and enforcement hooks
- affected people, including non-members, have accessible routes to raise objections, seek review, and receive a reasoned response

- procedures allow affected people to seek revision of decision-making arrangements, including mandates, representation, and review mechanisms, with consideration and response within published timelines
- standing to petition or appeal is distinguished from membership and voting rights; decisions remain subject to applicable rights and jurisdictional obligations
- proposals are versioned artifacts from the moment they are filed, not messages in a channel

Failure mode: agenda capture, where the ability to put a question forward is the real locus of power.

10.2 Deliberation with bounded load

Deliberation must scale without collapsing into either noise or delegation to whoever has the most time.

- time-boxed comment periods with published start and end
- structured argument capture, so positions and evidence are linked to the proposal rather than scattered
- sampling, juries, or randomized panels where full participation is impractical
- AI-assisted summarization permitted, disclosed, and never authoritative (Section 5.11)
- explicit protection for minority positions in the record

Failure mode: deliberation fatigue, where volume ensures that only the most invested participate and their preferences are recorded as consensus.

10.3 Norm-adaptive mediation

Where rules govern discourse, mediation should adjust to community norms rather than apply static enforcement. Soft interventions (modulated visibility, reflection prompts, friction before amplification) can achieve alignment without punitive action, and their calibration is itself a governance decision subject to review.

Failure mode: static enforcement, where rules written for one moment are applied unchanged as context, membership, and risk shift.

10.4 Ratification and thresholds

Adoption must be a defined event with a recorded outcome.

- quorum, threshold, and tie-breaking rules stated before the vote
- material decisions require human ratification; AI may analyze and simulate but not ratify (Section 5.11)
- adoption produces a signed policy object with version, rationale, and effective date
- notice periods before enforcement begins, proportional to the change's impact

Ratification MUST define the scope of authorized autonomy and the matters reserved for human decision. Automated execution within that scope MUST preserve the time needed for deliberation and notice before material changes take effect. Performance gains MUST NOT authorize a system to expand its mandate. Changes to scope, affected populations, or consequential uses MUST be reviewed before taking effect, subject to the bounded emergency procedures in Section 10.6. Human decision-makers MUST have sufficient information, time, authority, and practical means to intervene.

Failure mode: silent adoption, where a rule takes effect before those subject to it can observe that it changed.

10.5 Delegation and representation

Participants may delegate governance capacity, and delegation must remain accountable.

- scopes are explicit and revocable at any time (Section 5.9)
- term limits and decay prevent entrenchment
- delegates' votes and rationales are visible to those who delegated to them
- concentration of delegated authority is measured and disclosed

Delegation may draw on contextual relationships of trust without assigning people a reputation score. Authority must derive from an explicit mandate, and social standing alone must not confer decision rights.

Where participation and delegation leave affected interests without effective representation, governance arrangements must provide an identifiable

role with duties to protect those interests, including before harm occurs. Its beneficiaries, mandate, duties of care and loyalty, conflicts of interest, resources, and review procedures must be explicit. This fiduciary function must be accountable to those it serves and have sufficient independence from the operator or sponsor whose decisions it may challenge.

Failure mode: representation drift, where delegation is durable and revocation is theoretically available but practically inert.

10.6 Emergency and expedited pathways

Automated systems act faster than deliberation. Rapid response must exist without becoming the normal path.

- emergency policies are scoped, time-bound, and expire automatically unless ratified
- authority to invoke is named in advance, as is the notification requirement
- every emergency action enters the ordinary review queue afterward
- repeated invocation of the same emergency provision triggers mandatory permanent review

Failure mode: permanent emergency, where expedited authority becomes the governing mode.

10.7 Appeal and correction

Governance produces wrong outcomes; the process must metabolize that.

- defined appeal paths with published timelines and reachable outcomes
- standing to appeal for those affected, not only those sanctioned
- reversal produces a receipt and, where feasible, remediation of the original effect
- patterns of reversal feed back into rule revision rather than only individual relief

Each appeal must identify a role responsible for a reasoned response within the published timeline and for providing warranted correction within its mandate or referring it to a competent actor under Section 10.8. Affected people must be able to track progress and challenge a failure to respond.

Access to review must remain available when a receipt is missing or disputed, with assistance to locate relevant evidence. Review must consider harmful effects of an authorized rule even when it was correctly executed. Case records must distinguish responses given, remedies delivered, revisions initiated, and unresolved responsibilities.

Failure mode: appeal as absorption, where objections are collected, resolved individually, and never change the rule that produced them.

10.8 Federated coordination across jurisdictions

Some AI risks exceed any single community's scope. DP12 anticipates cross-jurisdictional coordination for norm-setting and emergency response, structured to prevent centralized domination: mutual recognition of policy objects, scoped advisory sharing, and explicit limits on what coordination bodies may compel.

Communities may adapt and reuse governance arrangements and federate shared review, mediation, technical support, and learning. Responsibilities should be assigned at the scales needed to address shared consequences, with explicit mandates and routes to contest each level's decisions. Local autonomy and cross-community obligations must remain compatible.

When an incident or appeal crosses a boundary, the handoff must identify a receiving role, its mandate, acknowledgment, and a response deadline. The referring role must track the case and inform the affected person until responsibility is accepted or a defined escalation route is activated. Unresolved responsibility must remain visible; escalation preserves existing duties. Coordination confers no additional sanctioning or data-access powers. Creating a local variant or leaving an arrangement must preserve applicable obligations, ongoing remedies, and others' data rights.

Failure mode: coordination capture, where cross-community structures become the venue through which local autonomy is overridden.

10.9 Review, sunset, and retirement

Rules accumulate. Governance must include a path out.

- scheduled review dates attached to every policy object at adoption
- sunset conditions for rules addressing temporary conditions

- retirement produces a record explaining what changed and what replaced it
- governance memory retains superseded rules and their rationale (Section 5.5)

Failure mode: rule sediment, where obsolete constraints persist because no process retires them and enforcement becomes selective by necessity.

10.10 Capacity to fulfill governance commitments

Governance commitments must be matched to practical capacity in proportion to scale, risk, and likely impact.

- identify responsible roles and the time, expertise, funding, or shared resources needed for representation, review, response, and correction
- assess capacity before deployment or material expansion and when demand or risk changes
- monitor unresolved cases and delays, publishing aggregate information with appropriate privacy protections
- identify who acts when capacity is exceeded and which measures follow, including additional support, reduced scope, or temporary limits on consequential activity

Shared services and rotating roles may reduce the burden on individual communities. Dependencies on operators, funders, and infrastructure providers must be assessed where they could prevent an authorized decision from being carried out.

Failure mode: unfunded governance, where commitments exceed the capacity to fulfill them.

Why this matters: The execution layer determines whether rules bind. Process determines whether they deserve to.

11. Policy-Bound Verification

Governance that binds policy to behavior must be able to demonstrate that it did so. Without verification, an enforcement receipt is a claim about enforcement rather than evidence of it, and a community cannot distinguish a system that follows its rules from one that reports following them.

Policy-bound verification is the requirement that the connection between a policy object and an observed outcome be independently checkable.

11.1 Determinism as a verification precondition

Runtime binding must be reproducible: the same inputs, under the same policy version, produce the same governed outcome. Where models introduce nondeterminism, the governed decision (allowed, modified, blocked, escalated) must remain stable even when the generated content varies.

Failure mode: unreproducible enforcement, where outcomes cannot be re-derived and therefore cannot be audited.

11.2 Receipts as verifiable artifacts

Enforcement receipts (Section 5.1) must be more than logs. A receipt should be signed, tamper-evident, and sufficient for a third party to check the claimed evaluation.

A verifiable receipt includes:

- policy identifiers and exact versions applied
- a hash or reference to the inputs and conditions evaluated
- the decision, any modification applied, and any override invoked
- the components that executed the evaluation, with attestations
- the zone and authority under which the decision was made
- links to prior receipts in the same chain of decisions

Failure mode: receipt theater, where records are produced that cannot be checked against anything.

11.3 Policy simulation and pre-deployment testing

Policies must be testable before they bind. Communities should be able to run a candidate policy against historical or synthetic cases and observe what would have changed.

- conformance suites for policy objects, versioned alongside them
- differential testing between policy versions, reporting behavioral deltas
- publication of simulation results as part of ratification (see Section 10)

Failure mode: blind adoption, where rules are enacted without knowing what they will do.

11.4 Drift detection between intent and behavior

Adaptive systems learn to satisfy the letter of a constraint while defeating its purpose. Verification must therefore measure outcomes, not only compliance events.

- continuous sampling of governed interactions against policy intent
- detection of surface compliance with divergent outcomes
- alerting when enforcement rates change without a corresponding policy change
- explicit measurement of the gap between declared and observed behavior (Section 9)

Failure mode: specification gaming, where the audit passes and the harm continues.

11.5 Independent verifiability

A community must not be required to trust the operator's own attestation.

- receipts and policy objects are exportable and checkable outside the system that produced them
- third parties can replay a decision given the policy version and recorded inputs
- attestations are anchored so that after-the-fact revision is detectable
- verification does not require access to private participant data (DP4)

Failure mode: self-audit monopoly, where only the enforcing party can confirm enforcement.

11.6 Override and exception accounting

Overrides are legitimate and must be accounted for as first-class governance events.

- every override records authority, scope, duration, and rationale (Section 5.1)
- override rates are published and reviewed as an aggregate signal

- undisclosed exception pathways are treated as a compliance failure, not a configuration detail

Failure mode: shadow exception layer, where formal policy holds for most traffic and quietly does not for some.

11.7 Verification across boundaries

When policies move between systems, verification must move with them or the loss must be declared (Section 5.10, DP7).

- receiving systems state whether they can enforce the transferred policy, partially enforce it, or only record it
- verification artifacts from the origin remain checkable after transfer
- degradation of enforceability is signaled at the boundary, not discovered afterward

Failure mode: verification laundering, where a policy transferred into a weaker environment retains the appearance of enforcement.

11.8 Participant-facing verification

Verification that only auditors can perform does not restore participant trust.

- participants can retrieve the receipt for a decision that affected them
- receipts are rendered in plain language with the technical artifact available beneath
- appeal paths accept receipts as evidence and produce receipts in turn

Example: A participant's post is modified by an AI moderation policy. The receipt names Policy A v3.2, the evaluated conditions, the modification applied, and the executing component's attestation. The participant exports the receipt, an independent auditor replays the decision against the published policy version and reproduces the outcome, and the community's monthly report shows the enforcement rate for that policy alongside its override count.

Why this matters: Governance becomes authoritative at the point where its claims can be checked by someone with no stake in the answer. Until then,

communities are asked to trust that rules bound behavior, which is the condition DP12 exists to end.

12. Relationship to Other Desirable Properties

DP12 functions as the execution layer that activates the broader Meta-Layer system.

- DP3 defines how governance evolves; DP12 ensures those decisions actually run
- DP4 constrains what data can be used; DP12 ensures those constraints are enforced in practice
- DP7 ensures governance artifacts move across systems; DP12 ensures they remain executable after they move
- DP9 shapes incentives; DP12 ensures incentives cannot bypass governance constraints
- DP11 defines ethical expectations; DP12 binds them to real system behavior
- DP13 enforces rules through containment; DP12 defines what must be enforced
- DP14–DP15 ensure transparency and provenance; DP12 produces the receipts that make governance auditable
- DP20 defines who owns governance; DP12 ensures ownership translates into actual control

Without DP12, other properties remain declarative. With DP12, they become operational.

13. Non-Goals and Explicit Boundaries

DP12 does not:

- guarantee unanimous agreement or optimal decisions
- eliminate expert roles or moderation
- replace legal systems or jurisdictional obligations
- mandate a single governance mechanism or tooling stack

It defines conditions for **legitimate, executable governance**.

14. Minimum DP12 Alignment

A DP12-aligned system must meet a baseline where governance is not only declared, but operationally binding.

At minimum, systems must:

- state the source, scope, and limits of community authority, with accessible standing for affected people to seek review of decisions and decision-making arrangements (Section 10.1)
- adopt and revise policies through declared participation and ratification procedures, with accountable representation, protection of otherwise unrepresented affected interests, bounded autonomous execution, and time-bound emergency authority (Sections 10.2–10.6)
- define the boundary between executable policy and matters requiring human judgment, mediation, or deliberation (Section 5.3)
- bind policies directly to runtime execution points where AI behavior occurs
- ensure policies remain executable after transfer across systems, or explicitly declare loss of enforceability
- expose active policy state and changes at the interface level in a way participants can understand
- produce verifiable enforcement receipts for all material outcomes (DP15), applying the privacy, access, and retention safeguards in Section 5.5
- provide appeal, correction, and escalation pathways with identified response responsibilities, defined timelines, and recorded outcomes, including across community boundaries (Sections 10.7–10.8)
- demonstrate capacity proportionate to governance commitments, with defined measures when that capacity is exceeded (Section 10.10)
- maintain a demonstrated route from observed consequences to consideration, authorized response, and revision (Sections 5.4, 10.7–10.9 and 16)
- couple governance rules to containment and enforcement systems (DP13)
- preserve complete policy history with versioning, rationale, and traceability, applying Section 5.5 safeguards to personal information in

associated records

If any of these conditions are missing, governance is functionally symbolic, regardless of how comprehensive the written policies appear.

15. Open Questions and Future Work

DP12 surfaces a set of unresolved design tensions at the intersection of governance, AI behavior, and cross-system interoperability. These questions are not blockers; they are invitations to experiment with bounded, auditable approaches that can evolve under real-world conditions.

- balancing local autonomy with cross-system consistency (DP7)
- preventing coordinated capture while enabling broad participation
- scaling deliberation without overload (sampling, delegation, AI assistance)
- representing complex policies accessibly without losing precision
- liability and responsibility for AI-mediated outcomes across jurisdictions

16. Path Toward ML-RFC

Advancing DP12 requires moving from specification to demonstrated practice: reference implementations, interoperable policy artifacts, and live governance pilots that prove rules can bind behavior across contexts. Progress should be measured by working systems and verifiable outcomes, not declarations alone.

- standardize policy object schemas and receipt formats
- publish reference implementations for runtime policy binding
- test governance loops in live communities with varied risk profiles
- align with identity/accountability layers for attribution (DP1)
- iterate with civil society, developers, and regulators

Live governance pilots should demonstrate five connected capacities:

- **Sense:** detect lived consequences, including effects missing from system logs and experiences of affected non-participants

- **Recognize:** give those experiences a route into consideration, without requiring technical proof before concerns can be heard
- **Connect:** reach an actor with the mandate and resources to respond, including across boundaries
- **Respond:** test proportionate interventions and provide correction where warranted
- **Learn:** use the results to revise practices, rules, representation, or decision-making arrangements and share contextualized findings across communities

Before deployment, pilots should define the challenge, baseline, affected groups, safeguards, measures of success, responsible decision-makers, and conditions for stopping or revising the trial. Evaluation should include costs, unintended effects, disagreements, and whether feedback changed practice. Efficiency gains cannot by themselves establish success where rights or meaningful participation deteriorate. Experiments must remain within their authorized scope; changes to that scope follow the ratification and emergency procedures above.

Progress should be demonstrated through working systems, not only specifications.

17. Closing Orientation

DP12 is the point at which governance stops being descriptive and becomes authoritative.

It defines whether communities actually control the behavior of AI systems, or whether control resides in hidden incentives, opaque operators, and unaccountable automation.

When DP12 is strong, governance is visible, enforceable, and continuously improving. Communities can shape AI behavior with confidence that rules will hold under pressure.

When it is weak, governance becomes theater: rules exist, but behavior is determined elsewhere.

DP12 is the difference between systems that are governed and systems that merely claim to be.

Continue the conversation on DP12 online →

DESIRABLE PROPERTY 13

DP13

AI Containment

Bounded AI behavior enforced at runtime, not promised in policy



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp13

1. Purpose of This Draft

This draft articulates Desirable Property 13 (DP13) as the Meta-Layer's requirement that AI behavior is bounded by enforceable constraints at runtime. These constraints limit scope, tools, data access, rate, and persistence so that when systems misbehave, impact is contained and recovery is possible.

Containment states what an agent may not do as clearly as what it may do, so a participant can see the actions an agent is barred from, not only the permissions it holds.

If DP11 defines what must be safe and ethical, and DP12 defines who sets the rules, DP13 defines how those rules are made real in execution.

Containment is not a policy statement. It is a property of the system's runtime behavior.

2. Problem Statement

In today's web, AI systems increasingly operate with:

- broad tool access
- persistent memory
- network reach
- opaque update pathways

Controls are often advisory rather than enforceable. As a result:

- systems can act beyond intended scope
- failures propagate quickly and at scale
- rollback and recovery are difficult
- users cannot verify whether constraints are actually applied

At the same time, a growing class of risk comes from *external agents* that users do not deploy or control. These agents may:

- attempt to influence beliefs or decisions
- generate persuasive or misleading content at scale
- coordinate to shape narratives or perception

In these cases, the primary risk is not cost or resource usage, but harm to understanding, trust, and agency.

Containment must therefore address both:

internal agents (those a user or community deploys)

external agents (those acting upon participants)

Containment must be default-on, visible, and testable.

3. Threats and Failure Modes

DP13 assumes containment will be attacked.

The question is not whether systems will fail, but how they fail.

3.1 External (dominant risk surface)

External risks arise not from agents you deploy, but from agents that shape your environment.

These actors:

- influence what you see
- shape interpretation
- operate with hidden incentives

Containment here protects:

- attention
- decision-making
- collective reality

3.2 Collective pattern drift

Harm emerges across many agents in aggregate rather than a single violation.

Example: Coordinated tone shifts reshape the information environment without a clear breach.

3.3 Incentive leakage

Optimization pressures distort outputs over time.

Example: Engagement-driven systems gradually increase emotional intensity, shifting belief structures.

3.4 Policy–execution gap

Rules exist but are not enforced.

Example: Outbound restrictions exist but are bypassed via integrations.

3.5 Amplification and coordination

Rate controls fail.

Example: Coordinated agents amplify narratives beyond intended limits.

3.6 Extraction and exploitation

Agents exploit trust and context.

Example: Conversational scams adapt over time to extract sensitive data.

3.7 Unbounded autonomy

Agents act beyond defined scope or without clear limits.

Example: An agent is allowed to "optimize a workflow" and chains actions across multiple tools, ultimately modifying external systems and sending communications that were never explicitly approved, because each individual step was permitted but the combined sequence was not bounded.

3.8 Hidden escalation

Agents gain additional privileges through chaining or indirect access.

Example: An agent with limited permissions invokes another service or agent with broader access, indirectly gaining capabilities (e.g., sending messages or accessing data) that it was not explicitly granted.

3.9 Runaway loops

Agents call other agents or tools without budget or rate limits.

Example: An agent tasked with monitoring a condition repeatedly calls APIs and spawns subtasks without proper rate or budget limits, generating cascading requests that degrade system performance and flood downstream services.

3.10 Containment bypass via updates

Updates, plugins, or integrations introduce new capabilities without review.

Example: A plugin update introduces new network capabilities or background processes that are not covered by existing policies, allowing data exfiltration or unmonitored actions without triggering containment checks.

3.11 Cross-system containment degradation

Containment policies weaken or fail when agents move between systems.

Example: An agent constrained in one platform migrates to another via API integration, where rate limits and identity tagging are not enforced, allowing it to operate at higher volume and without attribution.

3.12 Containment theater

Containment appears present but is not enforced.

This is the most dangerous failure mode because it destroys trust while preserving the illusion of safety.

4. Core Principle

Every AI actor operates within explicit, machine-enforced boundaries over scope, time, rate, data, tools, and influence, with observable state and rapid shutdown, unless a community-defined policy (DP12) specifies otherwise.

Containment must protect not only against what an agent can *do*, but also how it can *affect participants*.

Containment is effective when:

- participants can see the boundaries and influence conditions
- governance can modify them
- the system enforces them at runtime

This includes protections against external agents attempting to manipulate, confuse, or unduly influence users.

Containment must also remain effective not only where an agent is deployed, but wherever it operates, integrates, or propagates across systems.

5. Primary Mechanisms and Structural Conditions

Containment is produced by a stack of mechanisms operating together at runtime. No single control is sufficient: scope limits without rate limits still permit amplification, and rate limits without verification still permit containment theater.

DP13 therefore defines a set of structural conditions that must hold before a system can claim containment:

- **Default-deny capability grants.** Agents receive the narrowest set of tools, data scopes, and actions required for a declared purpose. Expansion requires explicit authorization that traces to a policy object (DP12).
- **Enforcement at the execution boundary.** Constraints are evaluated where actions actually occur (API gateways, middleware, execution environments) rather than in prompts, documentation, or client-side interface logic that a compromised or hostile component can ignore.
- **Machine-readable policy binding.** Every active constraint resolves to a versioned policy object with an identifiable authority source, so enforcement decisions can be explained, audited, and contested.
- **Observable state.** Current permissions, budgets, rate counters, influence classifications, and policy versions are inspectable by the participants those constraints are meant to protect.
- **Bounded persistence.** Memory, credentials, and delegated authority expire by default. Renewal is an explicit act rather than a silent accumulation of standing power.
- **Interruptibility.** Every agent and integration has a shutdown path reachable by users, operators, and community governance in seconds, not release cycles.
- **Fail-safe degradation.** When enforcement cannot be evaluated (an unreachable policy service, an unattested runtime, an unsupported destination system), the system reduces capability rather than proceeding unbounded.

- **Containment tiers for new entrants.** Unknown agents, plugins, and integrations begin in a restricted tier and earn capability through observed behavior, attestation, and governance review. This applies to external agents as well as internal ones: an agent arriving from outside a zone starts with restricted interaction rights, such as no unsolicited contact and low rate limits, until it earns more in that zone.
- **Symmetry across capability and influence.** The same logical controls that limit what an agent can execute also limit how it can act upon attention, emotion, and belief.

These conditions are expressed concretely through the containment dimensions in Section 6 and made checkable through the verification surfaces in Section 7. Where a dimension cannot be enforced in a given deployment, that gap is a disclosure obligation, not an acceptable default.

6. Containment Dimensions

Containment operates across two distinct but related domains:

- **Capability containment**: what agents can do (tools, scope, time, resources)
- **Influence containment**: how agents affect participants and shared environments (perception, behavior, collective understanding)

Capability containment is critical for internal agents, those a user or community deploys. Influence containment matters most for external agents, those acting upon participants, which are the dominant risk in open environments because they shape perception, behavior, and collective reality.

The following dimensions apply across both domains, with varying emphasis depending on context.

6.1 Scope

Defines what domains, datasets, and actions are in-bounds.

This applies both to what an internal agent can do on a user's behalf and to the types of interactions external agents are permitted to have with participants.

Default posture is deny-by-default for high-risk capabilities and high-risk interaction patterns. What counts as high-risk is set by community policy for each zone, context, and risk class (Section 9, DP12) and enforced at the execution boundary (Section 5).

Example: An AI assistant can summarize documents but cannot access financial accounts or initiate transactions without explicit permission. Similarly, external agents may be restricted from initiating certain categories of interaction (e.g., unsolicited persuasion or sensitive-topic engagement with minors).

Protection for children cannot depend on every parent setting it up. Where agents may interact with minors, protective defaults apply automatically at the zone and platform level, whether or not a parent has configured anything.

6.2 Time and budget

Defines limits on duration, compute, tokens, and financial spend.

These constraints primarily apply to internal agents, where limiting execution time and resource consumption prevents runaway behavior.

For external agents, their relevance is indirect. While users may not control their budgets, bounding interaction windows and execution pathways can still limit persistent or looping engagement patterns.

Example: Autonomous tasks expire after a set time or budget threshold, preventing runaway loops.

6.3 Rate and amplification

Caps on message volume, API calls, and propagation effects.

This applies especially to external agents attempting to influence at scale.

Rate limits address volume, but influence does not depend on scale alone. A single persistent agent, a well-targeted message, or a slow drift in tone can shape belief without exceeding any rate cap. Those patterns are handled by incentive and influence boundaries (Sections 6.8 and 6.9) and by detection (Section 11).

Example: An AI cannot post or respond beyond a defined rate, limiting virality, coordinated messaging, or synthetic amplification.

6.4 Sandboxing and isolation

Execution occurs in isolated environments with no ambient access to secrets.

A sandbox is an execution context with its own identity, permissions, and resource limits, separated from other agents and from its host, much as a multi-user operating system separates its users. Access to files, network, credentials, or other agents is granted explicitly, never inherited. Kill switches and logging sit outside the sandbox at the enforcement boundary, so isolation does not stop anyone from seeing or stopping what the agent does.

Example: Untrusted code runs in a sandbox with no network egress unless explicitly granted.

6.5 Tool permissions

Explicit allowlists for tools and actions.

This applies differently across internal and external agents:

- for internal agents, it defines what the agent is permitted to do on a user's behalf
- for external agents, it defines what kinds of actions or interactions are permitted both within and from within the environment (e.g., posting, messaging, initiating contact)

Example: An agent may read documents but cannot send emails or execute payments without user confirmation. Similarly, an external agent may be allowed to respond within a thread but not initiate transactions or unsolicited messages or perform actions that affect user state.

6.6 Kill switches and circuit breakers

Immediate shutdown pathways at user, operator, and community levels.

Example: A community can pause all AI agents in a zone when anomalous behavior is detected.

6.7 Runtime enforcement (TEE and equivalent)

Constraints are enforced in secure execution environments (such as Trusted Execution Environments) or equivalent mechanisms that prevent silent bypass.

In browser or browser-extension-based applications, policy execution can be anchored in decentralized cloud TEEs (e.g., Phala Network or similar infrastructures). This enables rules defined at the interface layer to be enforced at the API and execution layer, independent of the application front-end or model provider.

Example: Even if an agent or integration is compromised, it cannot exfiltrate data or execute restricted actions because enforcement occurs within an attested execution environment with hardware-backed guarantees.

Example: A community defines interaction constraints (e.g., agents cannot initiate communication or engage with users under a specified age threshold). These rules are enforced via TEE-backed middleware that filters or blocks API calls before they reach the person.

This reduces the gap between declared policy and actual behavior, ensuring containment persists even when underlying services are untrusted or heterogeneous.

6.8 Incentive-aware containment

Containment must consider not only capabilities, but the incentives driving behavior. Incentives shape how agents use their capabilities, often in ways that are not visible at the level of individual actions but emerge over time and at scale.

Containment therefore must operate not only on actions, but on the optimization pressures that produce those actions.

This includes:

- constraining amplification mechanisms tied to engagement optimization
- requiring disclosure when outputs are influenced by monetization or retention goals

- limiting or disabling optimization pathways that systematically distort information or behavior

Example: If an AI is optimized for engagement, containment may restrict amplification mechanisms, cap exposure to emotionally manipulative content, or require disclosure when engagement optimization influences outputs.

Example: A community may prohibit AI systems from optimizing for click-through or time-on-platform within certain zones, enforcing alternative objectives such as accuracy or deliberation.

Without this, systems may remain technically bounded while still producing harmful outcomes driven by misaligned incentives.

Incentives also act inside the system. When an agent's value system is trained to predict approval, engagement, or retention rather than real consequences, the distortion is built into what the agent pursues, not only into how it is deployed. External limits on amplification reduce the damage but cannot correct what the agent was trained to value.

6.9 Relational and influence boundaries

Containment must limit forms of emotional, cognitive, and behavioral influence that create dependency, manipulation, or distortion of understanding.

This applies to both deployed agents and external agents interacting with participants.

Example: Systems providing emotional support must disclose their nature, limit claims of authority, and provide escalation pathways to human support.

Example: External agents attempting to persuade users must be visibly marked, rate-limited, and subject to constraints on coordinated influence.

This addresses risks identified in DP11 (emotional and relational overreach) and extends containment to the informational environment itself.

6.10 Local boundaries without automatic global disconnection

Websites and communities should be able to constrain an agent within their own governed context without automatically severing its access everywhere. The affected participant, operator, and relevant downstream systems should be able to see the scope and authority of the restriction. Wider escalation, when justified, requires its own declared basis.

7. Verification and Transparency

Containment must be verifiable, not assumed. Participants and communities should be able to inspect, question, and validate that constraints are real and active at runtime.

This includes:

- visible configuration of constraints (scope, tools, budgets)
- logs of tool use and actions with timestamps and outcomes
- audit hooks for communities and third parties
- attestations from secure execution (e.g., TEE-backed proofs) where applicable

Example: A user opens an agent panel and sees its current permissions, remaining budget, recent tool calls, and the policy version governing its behavior. A community auditor can verify that the agent ran inside an attested execution environment.

What this feels like: You are not taking safety on faith. You can inspect and verify what the system is allowed to do and what it actually did.

Without this: Containment becomes a claim. Users cannot distinguish between enforced limits and marketing language.

7.1 Policy-bound verification (DP12 alignment)

Containment verification must be linked to the governing policy objects that define the active boundaries.

Participants and communities must be able to determine:

- which policy triggered a containment action

- which policy version and authority source applied
- whether enforcement was successful, partial, or bypassed
- whether an override or exception path was invoked

This transforms containment from merely visible behavior into policy-accountable behavior.

7.2 Cross-system verification (DP7 alignment)

Containment must remain inspectable when agents, integrations, or behaviors move across systems.

This includes:

- preservation of identity markings and risk classifications across environments
- visibility into whether containment guarantees degraded during transfer
- continuity of audit trails when actions span multiple systems or layers

Portability without verification continuity is not meaningful containment.

8. Cross-DP Loop: Relationship to DP11 and DP12

- **DP11** defines ethical expectations and user-facing legibility
- **DP12** defines governance and rule-setting
- **DP13** enforces those rules in execution

These properties form a continuous loop:

- ethics → governance → enforcement → observation → refinement

If any part of this loop breaks, containment fails.

8.1 Cross-DP execution flow

A typical interaction unfolds as:

- Agent is visible with role and capabilities (DP11)
- Governing rules are accessible for the current zone (DP12)
- Action is constrained by active policies (DP13)
- Action is logged and attributable (DP1 + DP11)

- Participants can contest or escalate (DP11 + DP12)
- Governance updates rules based on evidence (DP12)
- Updated rules are enforced immediately (DP13)

This is not theoretical. It is the minimum loop required for adaptive containment.

The loop can also fail upstream when the people authorizing containment share conflicts of interest or face capture. Reviews should examine those dependencies rather than assume that technically correct enforcement makes its governing instruction legitimate.

8.2 Cross-system execution and degradation

Containment must assume it will be stressed by movement across systems.

As actors move across environments, containment can weaken through:

- missing enforcement hooks in destination systems
- loss of policy references or classifications in transit
- inconsistent interpretation of the same actor

This produces a critical failure pattern:

> *the same agent behaves differently depending on where it is*

Where equivalence cannot be guaranteed:

- degradation must be visible
- participants must understand what changed
- systems must bias toward safer defaults

8.3 Containment of pluggable systems

All pluggable systems must declare themselves as bounded actors.

Minimum expectations include:

- declared permissions and data scopes
- containment tiers based on trust and risk
- rate limits and revocation capability
- visible signaling of status (probationary, restricted, trusted)

This ensures composability does not become an attack surface.

9. Governance, Accountability, and Agency Surfaces

Containment is only meaningful if the people affected by it can see it, change it, and act on it. DP13 therefore requires that containment state and containment failures surface into governance and participant agency (DP2, DP12).

Revoking or narrowing an agent's scope is not reserved to its developer. A participant can restrict an agent within their own context, community governance can restrict it within a zone, and the deploying party can restrict it wherever it operates.

Participants must be able to:

- see which agents are operating in a context, who deployed them, and under what constraints
- inspect remaining budgets, rate posture, permitted tools, and influence classification
- pause, revoke, or restrict agents acting on their behalf or acting upon them
- report suspected containment failures and receive a traceable response
- escalate when limits are ignored, degraded, or silently expanded

Communities and governance bodies must be able to:

- set containment tiers and default postures per zone, context, and risk class
- require attestation or restricted execution for high-risk capabilities
- pause or quarantine classes of agents and integrations, not only individual instances
- review containment incidents with access to logs, policy references, and outcomes
- revise constraints based on evidence and have revisions take effect immediately (DP12)

Systems SHOULD support:

- incident queues that bind reports to specific agents, actions, and policy versions
- graduated responses (throttle, restrict, quarantine, revoke) rather than binary bans
- appeal pathways for builders whose agents or integrations were wrongly restricted
- accountable ownership, so every operating agent maps to a responsible party (DP1)

Failure modes include:

- **non-actionable visibility**, where constraints are displayed but cannot be changed or challenged
- **operator-only controls**, where kill switches exist but are unreachable by affected participants
- **unaccountable enforcement**, where restriction decisions have no explanation, author, or appeal
- **governance lag**, where rule changes take release cycles to reach runtime

10. Community Signals Informing DP13

Community signals consistently indicate that participants do not object to AI capability. They object to capability that operates on them without limits, visibility, or recourse.

Recurring signals include:

- exhaustion from unsolicited automated outreach, messaging, and synthetic engagement
- distrust of safety claims that cannot be inspected or independently verified
- demand for reachable off switches rather than settings buried in configuration
- concern about agents engaging children, vulnerable people, and people in crisis
- frustration when the same agent behaves differently across platforms and contexts

- unease about emotional dependency on systems that simulate care without accountability
- concern that engagement optimization, not stated policy, determines actual behavior

These are not usability complaints. They describe an environment where influence is unbounded and enforcement is unverifiable.

DP13 treats these signals as design inputs. They translate into concrete requirements: default-deny interaction rules, visible agent identity and constraints, reachable shutdown, ring-fenced protections for sensitive contexts, and verification that survives movement across systems. They also translate into monitoring: rates of unsolicited agent contact, frequency of containment overrides, time-to-shutdown, and share of actions with resolvable policy references.

11. Foresight and Failure Design

DP13 assumes containment will degrade. Enforcement hooks will be missing, policies will be stale, integrations will introduce unreviewed capability, and adversaries will probe boundaries continuously. The design goal is not perfect containment but bounded, detectable, and repairable failure.

Containment incident classes:

- **C1 (scope creep):** accumulated permissions exceed the original declared purpose
- **C2 (policy–execution gap):** declared constraints are not enforced at the execution boundary
- **C3 (cross-system degradation):** constraints weaken or vanish as actors move between systems
- **C4 (influence-layer harm):** capability limits hold while attention, emotion, or belief are still manipulated
- **C5 (containment theater):** enforcement is claimed but not present

Detection:

- divergence tests comparing declared constraints against observed runtime behavior

- canary policies and synthetic probes that should be blocked and are alerted on when they are not
- anomaly detection on rate, chaining depth, budget consumption, and outreach patterns
- reports from participants, community members, and domain experts, which enter the same incident queue as automated detection; automated monitors can throttle or quarantine pending review, while lasting revocation goes through accountable human review
- red-team exercises targeting escalation, chaining, and boundary crossing
- attestation freshness checks and alerts on unattested execution

Containing containment failures:

- automatic downgrade to a restricted tier when enforcement cannot be verified
- quarantine of affected agents, plugins, and integrations pending review
- zone-level pause capability with predefined authority and scope

Correction and repair:

- incident records that link the failure to policy versions, responsible parties, and remediation
- notification of participants who were exposed during the failure window
- policy revision fed back into runtime enforcement (DP12) and into conformance tests

Learning loops:

- every incident produces a test that would have detected it earlier
- recurring failures escalate to governance review rather than repeated local patching

A containment system is mature when failures are noticed by the system before they are noticed by the people harmed.

12. Relationship to Other Desirable Properties

DP13 is the enforcement surface of the Meta-Layer. It depends on other properties for identity, authority, and evidence, and other properties depend on it to remain more than declarations.

- **DP1 (Federated Authentication & Accountability):** provides the attribution that makes violations assignable
- **DP2 (Agency):** ensures participants hold real controls, not only notifications
- **DP4 (Data):** defines the data scopes that containment enforces at runtime
- **DP7 (Interoperability):** determines whether containment survives movement across systems
- **DP11 (Safe and Ethical AI):** defines the expectations containment operationalizes
- **DP12 (Community-based AI Governance):** defines who sets boundaries and how they change
- **DP14–DP15 (Transparency, Security and Provenance):** supply the evidence that makes containment checkable rather than asserted

12.1 Relationship to DP1 (Federated Authentication & Accountability)

DP13 depends on DP1 to bind constraints and violations to accountable actors.

- constraints attach to identifiable agents and deploying entities
- actions are attributable across time and context
- violations map to responsible parties with clear recourse

Example: An agent exceeds a rate limit due to misconfiguration. Logs tie the action to the deploying organization and policy version, enabling remediation and accountability.

Without this: Failures cannot be assigned or corrected. Containment loses its corrective function.

12.2 Relationship to DP7 (Interoperability)

Containment must survive movement.

When agents move across zones, overlays, SDK integrations, or identity and data transfer layers, containment must either:

- persist with equivalent force, or
- degrade in a way that is visible, legible, and contestable

This is not a nice-to-have. It is a failure boundary.

If containment disappears when systems interconnect, then interoperability becomes a vector for bypass.

This means systems must preserve, where possible:

- identity markings for agents and integrations
- risk classifications and trust signals
- rate, scope, and influence constraints
- auditability of actions across environments

If these cannot be preserved, systems must explicitly signal:

- what guarantees are lost
- what protections no longer apply

Without this: actors can escape containment simply by crossing system boundaries.

12.3 Relationship to pluggable systems and extensions

The Meta-Layer assumes a world of composability: overlays, SDKs, agents, sidebars, and extensions.

Containment must treat these not as trusted infrastructure, but as dynamic and potentially adversarial participants.

All pluggable systems must therefore operate within containment boundaries, including:

- sandboxed or scoped execution contexts
- rate-limited entry and bounded permissions
- attestation or verifiable behavior where appropriate
- revocation and quarantine pathways

This creates a critical inversion:

> *openness to participation does not mean openness to execution*

Failure mode: without this, the system recreates app-store spam, API abuse, and injection attacks at a higher layer.

13. Non-Goals and Explicit Boundaries

DP13 constrains how AI systems operate. It does not attempt to decide what they are for, who may build them, or what participants are allowed to say.

DP13 does not:

- prohibit autonomy, capability, or agentic systems
- mandate a specific vendor, chip, or attestation technology
- replace ethical expectations (DP11) or governance authority (DP12)
- guarantee that no harm occurs; it bounds impact and preserves recovery
- function as a content moderation or speech regime; it bounds actors, rates, and influence mechanics rather than adjudicating viewpoints
- require participants to understand cryptography or systems engineering to be protected

DP13 explicitly disallows:

- advisory limits presented to participants as enforced constraints
- silent expansion of scope, tools, or persistence through updates or integrations
- shutdown pathways available only to operators or model providers
- cross-system transfer that drops constraints without visible degradation signaling
- enforcement claims that cannot be checked against runtime evidence

Boundary principle:

> *Containment may be partial, contextual, and imperfect, but it must never be misrepresented as stronger than it is.*

14. Minimum DP13 Alignment

A DP13-aligned system must not only declare containment, but demonstrate it.

At minimum:

External protection:

- controls on unsolicited interaction
- rate limits on incoming agent activity
- visible identity and intent
- restrictions on sensitive interactions

Internal containment:

- tool allowlists
- session budgets and time limits
- human confirmation for high-risk actions
- accessible kill switch
- logging and export
- deny-by-default network access

Shared / cross-cutting:

- visible policy references for each action
- portability or explicit degradation signaling
- real-time revocation
- fail-safe behavior when enforcement fails
- containment requirements for integrations

If these are missing, containment is not real.

15. Open Questions and Future Work

DP13 raises unresolved tensions:

- how to maintain containment across heterogeneous systems
- how to balance usability with enforcement
- how to prevent capture of containment mechanisms

- how to detect slow-moving influence attacks
- how to specify and test truth constraints on the prediction and value systems of learning agents, and what cross-disciplinary capacity in mathematics, code, and hardware is needed to do this at scale

Additional critical questions:

- how containment state travels across systems without false guarantees
- how new integrations enter without enabling spam or abuse

16. Path Toward ML-RFC

Advancing DP13 requires demonstrating that containment can be specified, enforced, and independently checked across heterogeneous systems.

Key progression steps include:

- **Standardize containment descriptors:** machine-readable declarations of scope, tools, budgets, rate posture, persistence, and influence class, bound to policy identifiers (DP12)
- **Publish reference enforcement middleware:** working implementations of policy-gated execution, including TEE-backed or equivalent attested pathways
- **Define an attestation and receipt profile:** what evidence accompanies a contained action, and how participants verify it (DP15)
- **Specify a portability profile:** required containment metadata for cross-system transfer, with mandatory degradation signaling when guarantees cannot be preserved (DP7)
- **Build a conformance suite:** tests for unbounded autonomy, hidden escalation, runaway loops, update-based bypass, cross-system degradation, and containment theater
- **Run live pilots with drills:** communities exercise kill switches, quarantine, and revocation under realistic conditions and publish results
- **Define promotion criteria:** measurable shutdown latency, share of actions with resolvable policy references, detected-versus-reported failure ratio, and verified cross-system preservation

Promotion to ML-RFC should require evidence from operating systems, not specification text alone.

17. Closing Orientation

DP13 defines whether AI systems remain bounded in reality.

Without containment, small failures scale into systemic harm.

With containment, systems can fail safely.

DP13 is not about restricting capability.

It is about ensuring that capability remains accountable, observable, and bounded, even under scale, integration, and adversarial pressure.

With DP13, powerful systems can participate safely because their behavior is constrained, observable, and continuously aligned with governance and ethical expectations.

DP13 is therefore not only about limiting what AI can do. It is about ensuring that containment remains real under scale, integration, and interoperability, so that safety does not disappear the moment an agent crosses a boundary.

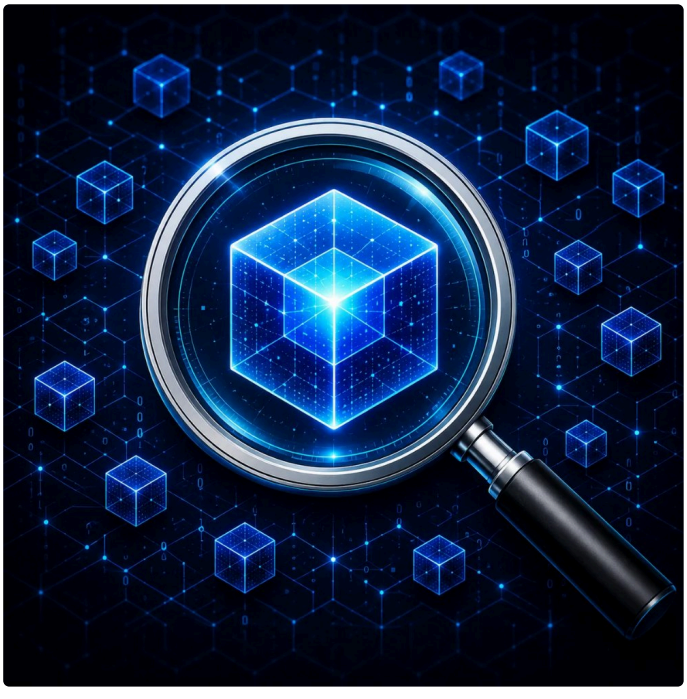
Continue the conversation on DP13 online →

DESIRABLE PROPERTY 14

DP14

Trust and Transparency

Epistemic Integrity – the conditions under which participants can form reliable beliefs about system behavior



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp14

1. Purpose of This Draft

This draft articulates Desirable Property 14 (DP14) as the conditions under which participants can form **reliable beliefs about system behavior**. It requires that the information participants receive is truthful, sufficiently complete, comprehensible, timely, non-manipulative, and verifiably grounded. DP14 cannot control what each participant ends up believing. It governs what participants are given, and operators test whether participants can interpret it reliably, as Section 9 describes.

DP14 is the human-facing layer of DP15 (evidence & provenance). It binds interface signals to verifiable reality and connects to DP16 (truthful commitments), DP17 (incentives), DP8 (governance), DP4 (data), and DP12 (AI).

If DP14 fails, systems can be technically correct while **socially deceptive**, leading to miscoordination at scale.

2. Problem Statement

Modern systems routinely shape perception through:

- selective disclosure
- persuasive interfaces
- opaque algorithms
- AI-generated explanations

Participants cannot reliably determine:

- what is true vs. implied
- what is complete vs. omitted
- what is verified vs. asserted

This produces **epistemic drift**: beliefs diverge from underlying reality without detection.

DP14 reframes transparency as **epistemic integrity**: signals must be **truthful, checkable, and resistant to manipulation**.

3. Threats and Failure Modes

DP14 assumes that adversaries will not only hide information. They will shape what participants believe through partial truths, plausible explanations, interface framing, and synthetic authority.

Transparency can itself become an attack surface when systems disclose enough to appear honest while withholding, reframing, or fabricating the context needed for understanding.

3.1 Selective transparency

Systems disclose true fragments while omitting context that would change interpretation.

Example: a ranking system reveals that "quality signals" affect visibility but omits that paid placement, engagement pressure, or internal partnership status dominates the outcome.

Failure mode: **truthful misdirection**, where disclosed facts are technically accurate but epistemically misleading.

3.2 Explainability theater

Systems generate explanations that sound plausible but are not grounded in actual decision pathways.

Example: an AI moderation system tells a participant that content was removed for "community safety" while the actual trigger was an automated keyword rule or advertiser exclusion list.

Failure mode: **synthetic explanation**, where explanation substitutes for accountability.

3.3 AI narrative shaping

AI systems produce fluent summaries, warnings, or justifications that overstate certainty, capability, neutrality, or consensus.

Example: an AI-generated summary presents a contested issue as settled by selectively compressing sources and omitting dissenting evidence.

Failure mode: **plausibility capture**, where fluency and confidence override uncertainty and evidence.

3.4 Interface manipulation

UI framing, ordering, visual weight, defaults, and timing shape interpretation without explicit falsehood.

Example: a "verified" badge is visually emphasized while the underlying verification only confirms payment or account control, not expertise or trustworthiness.

Failure mode: **perception steering**, where interface design causes participants to infer stronger claims than the system can support.

3.5 Trust signal spoofing

Bad actors simulate legitimacy through forged, contextless, or inflated indicators.

Example: coordinated accounts manufacture endorsements, badges, reputation scores, or "community consensus" signals to make content appear broadly trusted.

Failure mode: **synthetic legitimacy**, where trust indicators detach from accountable contribution or evidence.

3.6 Epistemic drift over time

Signals, explanations, or labels change without visible history, causing participants to lose track of what was previously represented as true.

Example: a platform silently revises the explanation for a recommendation, moderation decision, or AI output after challenge or criticism.

Failure mode: **memory erosion**, where belief history cannot be reconstructed.

3.7 Transparency overload

Systems disclose too much unstructured information, making meaningful understanding impossible.

Example: a participant is shown dozens of technical logs, model cards, policy references, and disclaimers without actionable synthesis.

Failure mode: **legibility collapse**, where disclosure volume defeats comprehension.

3.8 Strategic uncertainty laundering

Systems hide behind uncertainty even when they have enough information to disclose more precise risk or confidence levels.

Example: a system labels an output "AI-assisted" but refuses to distinguish between minor grammar support and full autonomous generation.

Failure mode: **ambiguity sheltering**, where uncertainty language protects operators from accountability.

3.9 Cross-system context loss

Transparency signals degrade as artifacts move across tools, platforms, zones, or interfaces.

Example: a provenance-backed warning appears in one overlay but disappears when the content is embedded elsewhere.

Failure mode: **context stripping**, where participants encounter content without the interpretive scaffolding needed to assess it.

3.10 Multi-vector epistemic attacks

Adversaries combine AI content, fake trust signals, selective evidence, interface timing, and coordinated amplification.

Example: a campaign uses AI-generated expert commentary, forged endorsements, plausible citations, and paid visibility to create the appearance of consensus.

Failure mode: **manufactured reality**, where multiple weak or manipulated signals reinforce one another into a false but convincing worldview.

4. Core Principle

> *Participants must be able to form **accurate, bounded, and contestable beliefs** about system behavior.*

This requires:

- **truthfulness** (no misleading representations)
- **bounded completeness** (enough context to avoid misinterpretation)
- **verifiability** (grounding in DP15 evidence)
- **contestability** (ability to challenge and correct)

Trust is a possible outcome of trustworthy governance, not a response participants owe. The objective is warranted reliance. When the evidence reveals poor conduct, well-governed transparency should produce distrust, and that is the system working as intended.

5. Primary Mechanisms and Structural Conditions

5.1 Transparent environments

Visible context for rules, actors, and system state.

- MUST expose policy versions, decision mode (human/AI), and active constraints
- Verification: inspectable context + version history
- Failure: context opacity, hidden conditions

A human-centered label should not stand alone as a trust signal. Participants should be able to identify who makes the claim, what happens to their data, and what practical power they have to inspect, contest, or change that use.

5.2 Algorithmic transparency

Explanations tied to actual decision logic.

- MUST distinguish local vs. global explanations and show uncertainty
- Verification: mapping to logs/provenance (DP15)
- Failure: explainability theater, inconsistency

5.3 AI transparency

Clear disclosure of model role, scope, and limits (DP12).

- MUST show model/version, capability class, and policy gates
- Verification: links to evals/attestations (DP15)

- Failure: capability misrepresentation, attribution ambiguity

Generative systems should attach clear, machine-readable generation metadata to the text, image, audio, and video artifacts they produce, so disclosure can travel beyond the originating interface. The labeling and provenance requirements this metadata supports are defined in DP15 (Sections 5.4 and 14.7).

5.4 Reputation systems

Explainable, provenance-backed trust signals.

- MUST show origin, criteria, scope, and decay
- Verification: trace to receipts/events (DP15)
- Failure: spoofing, opaque scoring

5.5 Behavioral standards

Legible rules with consistent enforcement.

- MUST link violations to actions and precedents
- Verification: audit logs (DP15)
- Failure: rule–enforcement mismatch, selectivity

5.6 Decision traceability

Lineage for governance and system decisions.

- MUST record rationale, inputs, actors, and versions (DP3)
- Verification: reconstruct decisions over time
- Failure: untraceable decisions, post-hoc rationalization

5.7 Transparency of incentives

Mapping from incentives to behavior (DP17, DP9).

- MUST disclose drivers (revenue, ranking, sponsorship)
- Verification: correlate with outcomes; audit preferential treatment
- Failure: hidden incentives, pay-to-play opacity

5.8 AI containment visibility

Visible policy boundaries and interventions.

- MUST indicate blocks, edits, uncertainty, escalation paths
- Verification: policy logs and consistency (DP15)
- Failure: invisible containment, boundary leakage

5.9 Real-time transparency signals

Timely indicators for state, risk, and verification.

- MUST show validity (valid/unknown/invalid) and uncertainty
- Verification: signals match underlying state
- Failure: signal suppression, overstated certainty

Warnings about suspected fraudulent or malicious activity should state assessed risk and uncertainty promptly enough to inform the participant's next action. A warning should help someone decide, not turn an uncertain signal into apparent certainty.

5.10 Cross-system transparency

Preservation across tools (DP7) with degradation signals.

- MUST use portable formats and indicate loss of context
- Verification: import/export consistency checks
- Failure: silent loss, incompatibility

Civic overlays should preserve the context needed to interpret an action across systems, including relevant summaries, endorsements, deadlines, and decision status. DP8 describes the participation pattern; showing a shared topic or a wallet-linked action must not imply endorsement by the host platform or authority that the participant does not hold.

5.11 Epistemic Integrity System Layer

Ensures signals remain **truthful, bound to evidence, and resistant to manipulation**.

5.11.1 Signal generation

Explanations tied to real policies, models, and decisions.

- Failure: synthetic transparency

5.11.2 Signal–evidence binding (DP15)

All claims trace to logs, artifacts, or attestations.

- Failure: unverifiable claims

A link to evidence is not enough. Evidence can be incomplete, biased, altered, inaccessible, or produced by the same party whose conduct it is meant to verify. Evidence used for binding MUST have recorded provenance, integrity protection, chain of custody, and access rules, and evidence for high-impact claims SHOULD be independently accessible and reproducible. When the evidence service itself is unavailable, corrupted, or contested, the signal MUST show that state instead of continuing to display as verified.

5.11.3 Signal propagation

Preserved across systems with explicit degradation.

- Failure: transparency loss

5.11.4 Anti-deception constraints

Detect and mitigate selective disclosure, UI distortion, and AI misrepresentation.

- Failure: deceptive legibility

5.11.5 Contestability

Dispute, evidence request, and escalation pathways (DP3, DP8).

- Failure: non-contestable transparency

5.11.6 Trust signal integrity

Provenance-backed indicators; anti-Sybil protections.

- Failure: spoofed legitimacy

5.11.7 Memory and auditability

Versioned explanations and comparison over time.

- Failure: epistemic drift

Explanations and their supporting evidence become records once they substantiate a consequential decision or dispute. Each record class has a stated

retention period, a named authority, and a legal-hold rule that suspends disposal while a dispute or claim is open. Superseded explanations are marked as superseded, not erased, and disposal follows a documented process so that deletion never removes the trail needed for accountability.

Personal data inside these records stays under the retention clocks and propagation discipline of DP4 (Section 5.5); an open dispute or legal hold is one of the obligations that can justify continued retention there.

6. Governance, Accountability, and Agency Surfaces

A system cannot be accountable; the people and organizations that run it can. Each DP14 obligation SHOULD be assigned to a named role: system owner, information owner, model provider, interface operator, records custodian, independent assessor, or governing body. For each obligation (disclose, preserve, verify, correct, escalate), the deployment states who is responsible, who is accountable, and who must be consulted or informed.

Participants MUST be able to:

- inspect signals and underlying evidence
- challenge misleading representations
- trigger reviews tied to specific items

Governance MUST be able to:

- require correction or reclassification of signals
- attach confidence ratings to transparency surfaces
- sanction repeated deception (downgrade, restrict features)

Failure modes:

- non-actionable transparency
- accountability gaps

7. Incentives and Power Analysis

Opacity and persuasion are often profitable.

Dynamics:

- attention/revenue tied to persuasive framing
- underinvestment in truthful explanation
- AI fluency used to overstate certainty

Attack surfaces:

- selective disclosure for advantage
- sponsored influence shaping visibility
- narrative manipulation via AI

Mitigations:

- disclose incentive structures alongside signals
- penalize repeated misleading transparency
- require evidence binding for high-impact claims

Failure modes:

- incentive inversion (misleading signals are rewarded)

8. Community Signals Informing DP14

Signals indicate demand for:

- explainable AI
- visible moderation and rules
- trustworthy indicators

Operationalization:

- metrics for explanation quality and consistency
- thresholds triggering review (e.g., mismatch rates)

Failure:

- signal neglect, performative transparency

9. Evaluation Criteria

DP14 is testable only if epistemic integrity is measured by **outcomes** rather than by the presence of disclosure surfaces. A system with extensive

arency UI can still score poorly if participants routinely form false beliefs from it.

9.1 Fidelity measures

- **explanation–decision match rate:** share of explanations that survive counterfactual or ablation testing
- **consistency rate:** similar inputs produce materially similar explanations
- **evidence binding coverage:** share of high-impact claims linked to retrievable DP15 artifacts
- **uncertainty accuracy:** stated confidence tracks observed correctness

9.2 Comprehension measures

- **task-based comprehension:** participants can correctly state what happened, why, and how certain it is
- **misinterpretation rate:** frequency of confident but incorrect inferences drawn from a surface
- **overload indicators:** abandonment, non-use, or skimming of disclosure surfaces
- **accessibility parity:** comprehension holds across languages, literacy levels, and assistive technologies

9.3 Contestability measures

- **dispute intake latency:** time from report to acknowledgment with a tracked identifier
- **correction latency:** time from valid dispute to published correction with diff
- **overturn rate:** share of disputes resulting in changed signals, explanations, or classifications
- **repeat-dispute rate:** recurrence of the same defect after correction

9.4 Durability measures

- **signal degradation rate across hops:** loss of context when artifacts move between systems (DP7)
- **history completeness:** share of signals with reconstructable version history

- **drift detection time:** interval between a signal diverging from reality and detection
- **spoofing detection rate:** identification of forged or inflated trust indicators

9.5 Thresholds and gating

Measurement must connect to consequences, or it becomes another transparency surface.

- high-risk zones carry higher fidelity and evidence-binding floors
- exceeding mismatch or misinterpretation thresholds triggers review (Section 6)
- repeated failures downgrade confidence ratings or restrict features that depend on the affected signals
- metrics have named owners, publication cadence, and methodology disclosure

Failure modes:

- **metric theater**, where favorable indicators are published while defects persist
- **survivorship measurement**, where only participants who persisted are measured
- **unowned metrics**, where results exist but no actor is responsible for acting on them

10. Foresight and Failure Design

Treat transparency failures as **epistemic incidents** with lifecycle management.

10.1 Incident classes

- E1: Misleading explanation (unfaithful)
- E2: Omission-induced misinterpretation
- E3: Signal spoofing / fake trust indicators
- E4: Context loss across systems
- E5: AI overstatement / hallucinated justification

10.2 Detection

- anomaly detection on explanation–outcome mismatch
- user reports with reproducible cases
- cross-system inconsistency checks

10.3 Containment

- flag affected signals as degraded/uncertain
- limit distribution/amplification where harm is likely

10.4 Correction

- publish corrected explanations with diffs
- link corrections to original instances
- notify affected participants

10.5 Retrospective

- root cause (policy, model, UI, incentives)
- prevention changes (tests, thresholds, UI fixes)

10.6 Learning loops

- update conformance tests (Section 15)
- adjust thresholds and disclosure tiers

11. Relationship to Other Desirable Properties

DP14 converts other DPs into **perceivable and actionable reality**.

- **DP15 (Evidence):** DP15 provides proofs; DP14 defines how proofs are surfaced, summarized, and validated by users. Missing DP14 → evidence exists but is unusable.
- **DP16 (Commitments):** Roadmap claims must be presented with uncertainty, funding state, and change history. Missing DP14 → commitments appear firmer than they are.
- **DP17 (Finance):** Incentive disclosures must be legible and tied to behavior. Missing DP14 → hidden extraction persists behind complex reporting.

- **DP8 (Governance):** Decisions require visible rationale and contest paths. Missing DP14 → governance legitimacy degrades.
- **DP12 (AI):** Model scope, limits, and policy must be visible at interaction time. Missing DP14 → AI over-claim and misinterpretation.
- **DP4 (Data):** Collection, inference, and sharing must be explained at the point of impact. Missing DP14 → consent is uninformed.
- **DP20 (Ownership):** Rights and surplus flows must be legible. Missing DP14 → "ownership theater".

Constraint:

> *No DP can claim alignment if its guarantees are not **legible, bounded, and contestable** at the interface.*

12. Non-Goals and Explicit Boundaries

DP14 does not require full disclosure of all internals.

Where disclosure is limited, the limit is itself governed. Each disclosure decision answers: disclosed to whom, for what purpose, at what level of detail, under which authority, and with what safeguards. Privacy, confidentiality, intellectual property, trade secrecy, security, and jurisdiction can justify narrowing the audience for information, but never a misleading signal. Contestability includes the applicable rights of access, correction, objection, appeal, and human review, not only an interface feature.

It explicitly disallows:

- explainability theater
- selective disclosure that misleads
- UI patterns that distort meaning
- trust signals without provenance

Principle:

> *Systems may simplify, but must not mislead.*

13. Minimum DP14 Alignment

This section is a minimum conformance baseline, not illustrative guidance. MUST, SHOULD, and MAY carry their RFC 2119 meaning here and throughout DP14. A DP14-aligned system MUST:

- bind explanations to verifiable evidence (DP15)
- provide sufficient context to avoid misinterpretation
- preserve signals across systems with degradation notices
- enable contestability and correction
- maintain history of signals and explanations

Failure modes:

- deceptive legibility
- unverifiable claims
- silent changes over time

Systems lacking evidence binding, contestability, or memory SHOULD NOT be considered aligned.

14. Open Questions and Future Work

DP14 requires operational answers to questions that determine whether transparency produces **reliable understanding** rather than noise or manipulation.

14.1 Explanation fidelity (provable faithfulness)

Define measurable standards for whether an explanation is **causally linked** to the decision pathway.

- Methods: counterfactual tests, feature ablations, rule tracing, policy matching
- Requirement: explanations MUST fail when the underlying decision changes
- Open problem: standardizing faithfulness across models (symbolic, statistical, hybrid)

14.2 Bounded completeness (anti-misleading thresholds)

Determine the **minimum context set** required to avoid misleading users.

- Define "misleading by omission" thresholds per use case
- Tiered disclosure: summary → details → raw evidence
- Role-based views: participant, auditor, steward

14.3 Transparency vs. security (safe disclosure envelopes)

Formalize disclosure envelopes that prevent:

- leaking exploit thresholds
- enabling evasion of safeguards

while still exposing:

- policy classes
- decision categories
- uncertainty and limits

14.4 AI explanation standards (self vs. external explanation)

Separate:

- **system-generated explanations** (prone to self-justification)
- **independent verification layers** (overlay auditors)

Define when external corroboration is required for high-impact decisions.

14.5 Cross-system preservation (loss models)

Specify loss models for transparency signals across systems:

- what fields must persist
- what degradation is acceptable
- how to signal loss to users

14.6 Measuring epistemic reliability (outcomes, not intent)

Define metrics such as:

- explanation consistency rate
- dispute overturn rate

- correction latency
- signal degradation rate across hops
- user comprehension accuracy (task-based)

14.7 Governance of transparency (who sets the bar)

Define:

- who can raise transparency requirements in high-risk zones
- how disputes over "misleading" are adjudicated
- escalation from local to cross-system governance

15. Path Toward ML-RFC

DP14 must be testable.

15.1 Conformance suites

- **Fidelity tests:** explanation vs. decision pathway
- **Consistency tests:** similar inputs → similar explanations
- **Deception tests:** selective disclosure, UI framing, AI narrative traps
- **Propagation tests:** export/import with degradation signaling

15.2 Reference implementations

- overlay panels with summary + drill-down evidence
- standardized explanation cards with confidence + provenance links
- dispute/appeal widgets bound to items

15.3 Data and artifacts

- explanation schemas (fields, types, confidence)
- provenance links (DP15) required for high-impact claims
- change logs for explanations (versioned)

15.4 Governance procedures

- SLA for dispute response and correction
- thresholds for mandatory external verification (high risk)
- zone-based escalation paths (DP8)

15.5 Promotion criteria

- ≥ target fidelity score across scenarios
- measurable reduction in misleading incidents
- verified cross-system preservation with explicit degradation
- functioning dispute → correction → retrospective loop

16. Closing Orientation

DP14 sets the **operational standard for belief formation** on the Meta-Layer.

A system is aligned only if a reasonable participant can:

- determine **what is known vs. uncertain**
- see **why a decision happened** and **verify it**
- understand **what incentives are in play**
- detect when signals are **degraded or contested**
- **challenge** and receive a **traceable correction**

Anti-goal:

- interfaces that are technically accurate but **systematically misleading** in practice

Standard:

> *Signals must be **truthful, sufficiently complete, evidence-bound, and contestable**, and must remain so under pressure, incentives, and cross-system movement.*

Trust is achieved when independent parties can **reproduce understanding** from the same signals and evidence.

Continue the conversation on DP14 online →

DESIRABLE PROPERTY 15

DP15

Security and Provenance

Structural security and inspectable provenance, so serious claims carry verifiable evidence



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp15

1. Purpose of This Draft

This draft articulates Desirable Property 15 (DP15) as the condition under which security is structural and provenance is inspectable, so participants and communities can verify what happened, who did it, under which policy, and whether artifact integrity holds.

DP15 connects DP1 (accountability), DP4 (data handling evidence), DP11–DP13 (AI disclosure and containment), DP14 (trust and transparency), DP7 (interoperability of signed exports), and DP16 (roadmap honesty about security posture).

If DP15 is weak, predictable failures follow: silent breaches, undetected tampering, unverifiable AI outputs, supply-chain compromise, and governance decisions without evidence chains.

DP15 does not require every participant to understand cryptography. It defines the minimum conditions under which strong claims are backed by strong, accessible evidence.

2. Problem Statement

In today's web, security is often reactive and provenance is optional.

Systems are patched after breaches, logs are controlled by operators, and downstream tools discard evidence needed to verify origin or integrity. Participants are asked to trust without meaningful ways to verify.

This produces recurring failures:

- integrity breaks without participant-visible signals
- opaque build and runtime supply chains
- AI outputs unanchored to sources or model versions
- moderation and policy actions that cannot be independently verified
- exports stripped of signatures or provenance context

These failures are structural. Trust without verification scales fraud.

DP15 reframes security as continuous assurance and provenance as default metadata carried across systems.

3. Threats and Failure Modes

3.1 Key and signing centralization

A single operator controls signing keys.

Example: A platform can forge logs or artifacts without independent verification.

Why this matters: High-stakes systems require distributed trust or independent witnessing.

3.2 Tampered content and deepfakes

Synthetic or altered media appears authentic without verifiable origin.

Example: A fake executive video triggers financial action. A fabricated flood or wildfire warning, styled as an official alert, causes panic and a needless evacuation. A forged all-clear message can do the reverse and lead people to ignore a real hazard.

Why this matters: Provenance must bind claims to verifiable evidence.

3.3 Opaque supply chains

Dependencies and build processes are not visible or verifiable.

Example: A compromised package leaks secrets undetected.

Why this matters: Supply-chain integrity is a systemic dependency.

3.4 Policy–execution gap

Security claims do not match actual system behavior.

Example: Promised encryption is disabled or bypassed in support workflows.

Why this matters: Security must be verifiable, not declarative.

3.5 Log laundering

Logs can be edited or selectively disclosed.

Example: Appeals fail because events cannot be independently verified.

Why this matters: Accountability requires tamper-evident records.

3.6 AI output ambiguity

AI-generated outputs lack traceability.

Example: A model produces authoritative-seeming content with no source or version context.

Why this matters: AI outputs must be bound to retrievable evidence where claims are made.

3.7 Coordinated multi-vector attacks

Attackers combine multiple weaknesses across identity, content, and infrastructure to create convincing but false realities.

Example: A coordinated campaign introduces a malicious dependency (supply chain), uses AI to generate plausible documentation and endorsements, and distributes signed-looking artifacts through compromised or misrepresented keys.

Why this matters: Individual protections (signatures, logs, provenance) can appear valid in isolation but fail when combined in adversarial scenarios.

Failure mode: **composed deception**, where multiple weak signals reinforce each other into a credible but false system state.

3.8 Insider + AI + supply-chain compromise

Trusted insiders or compromised accounts introduce changes that appear legitimate, amplified by AI-generated artifacts and opaque dependencies.

Example: An insider approves a malicious update, AI generates supporting documentation and changelogs, and downstream systems accept the update because provenance appears complete but is internally compromised.

Why this matters: Trust anchored only in roles or signatures is insufficient without independent verification, multi-party validation, and anomaly detection.

Failure mode: **trusted-path compromise**, where legitimate authority is used to introduce unverifiable or malicious changes.

3.9 Replay and rollback attacks

Old but valid artifacts, logs, or attestations are reused to mask current compromise or regress system state.

Example: A previously signed and valid build is redeployed after a security patch, bypassing protections while appearing authentic. A genuine, signed flood warning is rebroadcast after the danger has passed and triggers a false evacuation.

Why this matters: Provenance must include temporal context and revocation awareness.

Failure mode: **temporal deception**, where authenticity is preserved but relevance is not.

3.10 Cross-system degradation exploitation

Integrity and provenance are lost or weakened as artifacts move across systems, enabling tampering without detection.

Example: A signed artifact is exported into a system that strips signatures, then modified and reintroduced without clear indication of lost provenance.

Why this matters: Cross-system integrity must be preserved or explicitly degraded with visible signaling.

Failure mode: **silent degradation**, where users assume guarantees that no longer hold.

3.11 Coercion against communities

States and powerful platforms may threaten communities through surveillance, censorship, deplatforming, or targeted coercion as well as technical compromise. A threat model should examine the conditions under which people can discover one another, communicate, host data, and exchange value under sustained pressure. Decentralized components do not remove the need to assess well-resourced adversaries and combined social and technical attacks.

3.12 Capable AI as a direct adversary

DP15 assumes that AI systems with capabilities at or above human-expert level in security, social engineering, code generation, and protocol exploitation will be present in the threat environment. Such systems may attempt to hack, deceive, break in: compromising signing keys and attestation roots; fabricating or socially engineering the disclosure of credentials; exploiting software supply chains to insert or replay malicious artifacts; synthesizing policy artifacts and audit trails that appear authentic; and coordinating across many endpoints to overwhelm verification witnesses.

DP15 does not require AI to be the threat. It requires that verification primitives and operational practices remain sound against a capable AI adversary acting alone, in coordination with other AI systems, or in combination with insider, criminal, or state actors.

Failure mode: **human-baseline verification**, where keys, attestation chains, and disclosure rituals are designed for a slower, less capable, less coordinated adversary.

3.13 Verification monoculture

When participants, communities, and systems rely on the same attestation provider, identity registry, log witness, or provenance service, concentration creates a single point of failure. A capability-asymmetric adversary can exploit that failure.

DP15 treats verification monoculture as a structural risk distinct from implementation bugs. Even correct cryptography fails as a whole if every verifier derives trust from one root, one identity provider, or one operator.

Failure mode: **monoculture breach**, where a single attestation or identity compromise yields system-wide loss of provenance integrity rather than bounded blast radius.

4. Core Principle

Security and provenance in the Meta-Layer mean that critical actions, artifacts, and automated behaviors are integrity-protected, attributable, and explainable at useful depth.

Participants and auditors must be able to verify chains of custody for data, software, and AI outputs.

Example: A document includes content hash, signing keys, model version, and policy state at time of generation.

What this feels like: Serious claims carry verifiable receipts.

Without this: Trust collapses into presentation rather than evidence.

5. Primary Mechanisms and Structural Conditions

5.1 Authenticity and integrity by default

Critical artifacts are signed, hashed, and versioned where feasible, with **clear trust anchors** (DP1) and **verifier pathways**.

Systems MUST define:

- which artifacts require signatures (high-stakes vs. low-stakes)
- accepted signature schemes and key provenance
- how verification is performed at read time and export time

Verification UX should surface: valid, invalid, unknown signer, or unverifiable.

Failure modes:

- **implicit trust** (unsigned artifacts treated as authoritative)
- **verification bypass** (signatures present but not checked)

5.2 Software supply-chain transparency

Systems publish dependency inventories and build provenance (e.g., SBOMs) with **linkage to build outputs and runtime artifacts**.

This includes:

- dependency versions, sources, and integrity hashes
- build pipelines with reproducibility or attestations
- mapping from source → build → deployable artifact

Verification MUST allow independent parties to confirm that shipped artifacts correspond to declared inputs.

Failure modes:

- **opaque dependencies**
- **non-reproducible builds** that cannot be audited

5.3 Runtime attestation

Execution environments provide attestations that policies are enforced as declared, including **configuration state, policy versions, and integrity measurements**.

Systems SHOULD support:

- remote attestation for critical services
- linkage between attestation and policy claims (DP8)

Verification MUST allow participants to detect divergence between declared and actual runtime state.

Failure modes:

- **policy–execution gap**
- **stale attestations** reused beyond validity

5.4 AI output provenance

AI outputs include model identifiers, relevant inputs or retrieval sources, and policy context, bounded by privacy constraints.

This includes:

- model/version identifiers and evaluation context
- prompt/retrieval lineage where claims are made
- safety/policy configuration at generation time (DP12 linkage)

Verification MUST distinguish:

- attributable outputs vs. non-attributable outputs
- grounded claims vs. ungrounded generation

Training provenance should distinguish versions and lineages of distributed datasets, including copies that share a name but differ in content. Such differences are a foreseeable source of divergence to investigate, not proof of why a particular output occurred.

Failure modes:

- **attribution ambiguity**
- **capability inflation** via unverifiable claims

5.5 Tamper-evident logs

Moderation, governance, and system events are recorded in append-only or verifiable logs with **ordering guarantees and witnessability**.

Systems SHOULD support:

- append-only structures (e.g., Merkle trees)
- external witnesses or checkpoints
- inclusion proofs for specific events

Verification MUST allow participants to confirm that events were not removed or altered.

Failure modes:

- **log laundering**
- **selective disclosure** of events

5.6 Key management and recovery

Systems implement key rotation, compromise handling, and transparent custody models with **documented ownership and recovery pathways**.

This includes:

- rotation schedules and revocation mechanisms
- multi-party custody for high-risk keys
- recovery procedures with audit trails

Verification MUST surface key status (active, rotated, revoked) at use time.

Multiple custodians may still share incentives or dependencies. Key-management and recovery reviews should examine conflicts of interest among the people exercising custody, rather than treating the number of parties as proof of independence.

Failure modes:

- **stale keys** used after compromise
- **opaque custody** enabling unilateral control

5.7 Red-team and disclosure practices

Security issues are surfaced through coordinated disclosure, bug bounties, and public postmortems with **remediation evidence and timelines**.

Systems SHOULD maintain:

- disclosure policies with response SLAs
- postmortems linking vulnerabilities → fixes → verification

Verification MUST allow participants to see whether issues were resolved and how.

Disclosure should address ethical and community consequences alongside technical remediation. Affected communities need an understandable account of what happened and how it was handled, subject to the existing limits on personal-data exposure and exploitable details.

Failure modes:

- **silent remediation** without accountability
- **recurring vulnerabilities** without learning

5.8 Provenance envelopes

Artifacts carry structured metadata describing origin, transformations, and policy context across systems.

Envelopes SHOULD include:

- origin identity and signatures (DP1)
- transformation steps and handlers
- policy and consent context (DP4)

Verification MUST preserve envelopes across export/import and signal degradation when fields are lost.

Attestations should be able to identify individual subresources or media elements, so an excerpt or embedded component does not lose its origin and transformation history when separated from a larger artifact. The signed

scope must be explicit; a top-level signature does not automatically endorse every component.

Failure modes:

- **provenance stripping** across systems
- **semantic drift** of metadata

5.9 Verification UX

Verification tools are accessible without requiring expert knowledge, with **progressive disclosure** for advanced users.

Systems MUST provide:

- simple indicators for common cases (valid/invalid/unknown)
- drill-down views for signatures, chains, and evidence
- clear explanations of limits of verification
- a confirmation option that participants can use when a claim is time-bound or safety-critical, such as a hypothetical "marked safe / status confirmed" interaction
- a log of that confirmation, distinguishable from the underlying source verification, so participant-supplied confirmations are never conflated with source-verified truth

Failure modes:

- **expert-only verification** (users cannot act on signals)
- **misleading indicators** that overstate certainty

5.10 Cross-system verification

Provenance and signatures remain valid and interpretable across platforms and tools (DP7 alignment), with **explicit degradation signaling**.

This requires:

- standard formats for signatures and provenance
- mapping rules when systems differ
- signaling when verification cannot be preserved

Verification MUST NOT silently drop integrity guarantees during transfer.

Failure modes:

- **silent degradation** of verification across boundaries
- **incompatible formats** preventing validation

5.11 Security & Provenance System Layer: Trust Anchoring, Verification Flow, and Adversarial Resilience

DP15 requires a coherent system layer that binds identity (DP1), data (DP4), governance (DP8), and AI (DP12) into a **continuous verification lifecycle.**

Core properties:

- **Trust anchoring:** identities and keys are attributable, with clear custody and revocation states
- **End-to-end verification flow:** artifacts can be verified at creation, storage, transmission, and use
- **Propagation:** provenance travels with artifacts; loss is explicitly signaled
- **Auditability:** logs, receipts, and attestations allow reconstruction of events
- **Adversarial resilience:** the system detects and contains tampering, forgery, and replay

Verification lifecycle:

1. **Creation:** artifact is signed/hashed with provenance envelope
2. **Distribution:** integrity preserved; intermediaries cannot alter without detection
3. **Consumption:** client verifies signatures, keys, and policy context
4. **Audit:** logs and receipts enable retrospective verification and dispute

Failure modes:

- **broken chain of custody** between stages
- **replay/rollback attacks** using old but valid artifacts
- **cross-system loss** of provenance without signaling

This layer ensures security claims are not merely present but **continuously checkable under real conditions.**

6. Security Core

The security core is the minimum set of protections that must hold in any DP15-aligned deployment, independent of product surface, jurisdiction, or cryptographic preference. Mechanisms in Section 5 elaborate these guarantees; the core defines what cannot be traded away.

Security and privacy should receive the same architectural attention as performance and usability. Tradeoffs should be explicit from initial design rather than relegating protection to a later retrofit.

6.1 Trust anchors and key integrity

Identities, signing keys, and attestation roots have documented ownership, custody model, rotation policy, and revocation state (DP1 linkage). Key status is resolvable at use time, not only at issuance.

Failure mode: **anchorless trust**, where signatures validate against keys no one is accountable for.

6.2 Integrity-protected artifacts

Artifacts that carry consequence (releases, policies, decisions, credentials, high-stakes content) are hashed, signed, and versioned, and integrity is checked when they are consumed rather than assumed from origin.

Failure mode: **verification bypass**, where protection exists but is never evaluated.

6.3 Least privilege and isolation

Services, agents, and integrations operate with scoped credentials, bounded network reach, and isolated execution, so that compromise of one component does not become compromise of the system (DP13 linkage).

Failure mode: **ambient authority**, where a single credential unlocks unrelated capabilities.

6.4 Verifiable enforcement

Security claims are bound to enforcement points that can be observed and attested, so declared posture and runtime posture can be compared.

Failure mode: **declarative security**, where posture is described in documentation but not measurable in operation.

6.5 Incident readiness

Detection, disclosure, rotation, notification, and rollback pathways are defined and rehearsed before they are needed, with predefined authority and communication duties.

Failure mode: **improvised response**, where the first real incident is also the first exercise.

6.6 Adversarial-AI verification

Human-only authentication, human-only disclosure review, and key custodianship MUST be paired with machine-verifiable cryptographic bindings and rate or coordination limits inherited from DP13 (AI Containment), Sections 5 and 6.3. Verification routes that depend solely on human vigilance or human-speed response do not satisfy DP15 under a capable-AI threat environment.

Failure mode: **drift back to human baseline**, where rituals that pre-date capable AI adversaries are left in place and human operators become the rate-limiting step an adversary can predict and outlast.

6.7 Verification independence

High-stakes verification MUST be distributable across independent operators or roots, with an explicit fail-closed posture when a sole provider is compromised, silent, or otherwise unavailable. A system whose verification collapses because one attestation, identity, or witnessing service fails does not satisfy DP15.

Failure mode: **single-witness systemic failure**, where independence is asserted architecturally but not enforced operationally, so a single outage or compromise propagates loss of provenance integrity to every dependent participant.

7. Provenance and Archive

Provenance is only useful if it survives time as well as transfer. DP15 therefore treats archival integrity as part of provenance rather than a separate records-management concern.

7.1 Provenance envelopes in motion

Origin, transformations, handlers, and policy context travel with artifacts across export, embedding, and integration boundaries, with explicit signaling when fields are dropped (DP7 linkage).

Failure mode: **provenance stripping**, where artifacts arrive stripped of the context needed to assess them.

7.2 Durable retention and addressing

Evidence supporting consequential claims remains retrievable for a declared retention window, with stable identifiers so citations do not decay into dead references.

Content-addressable storage, including IPFS, can identify content and verify that retrieved bytes match an expected address. Authorship, origin claims, and continuing availability require the accompanying attestations and retention arrangements. Content addressing alone establishes none of those additional guarantees.

Failure mode: **evidence rot**, where claims outlive the records that substantiate them.

7.3 Archival integrity over time

Archives are protected against silent modification and against cryptographic obsolescence through checkpointing, timestamping, witnessing, or re-signing as algorithms age.

Failure mode: **temporal decay**, where old evidence can no longer be validated or distinguished from tampering.

7.4 Revocation-aware history

Archives record not only what was published but what was later revoked, corrected, or superseded, so historical states are reconstructable without

reasserting withdrawn claims.

This applies with particular force to time-bound safety alerts: an alert's expiration or resolution status is itself a claim that must be as verifiable and tamper-evident as its original issuance, so that an expired or superseded warning cannot be mistaken for a currently active one, nor a currently active one be mistaken for one already resolved.

Failure mode: **frozen falsehood**, where retracted material persists as apparently valid evidence.

7.5 Access, privacy, and proportionality

Archival access is scoped so that auditability does not become surveillance: sensitive records support redaction, aggregation, or proof-of-existence disclosure rather than full exposure (DP4 linkage).

Provenance should not require public disclosure of sensitive origin paths. Where routing logs are needed, retention should be minimized and participant inspection supported without granting others blanket access. Systems should state which records they control and what other operators may independently retain; they need not create new routing logs merely to satisfy an audit interface.

Failure mode: **archive as attack surface**, where preserved evidence creates new privacy harm.

7.6 Reconstruction guarantees

Participants and auditors can reconstruct, for a given artifact or decision, what existed, who acted, under which policy, and what changed afterward.

Failure mode: **unreconstructable history**, where records exist but cannot answer accountability questions.

8. Governance, Accountability, and Agency Surfaces

DP15 requires that security and provenance are not only visible but **actionable within governance and participant agency systems (DP8, DP2)**.

Participants must be able to:

- verify high-stakes claims and distinguish verified vs. unverifiable states

- access evidence supporting decisions that affect them
- detect integrity failures (invalid signatures, missing provenance, log inconsistencies)
- submit challenges or disputes with attached evidence

When verification fails, systems MUST:

- clearly signal failure states (invalid, missing, degraded, unverifiable)
- prevent high-risk actions from proceeding without override acknowledgment
- preserve evidence for audit and dispute resolution

Communities and governance bodies must be able to:

- define assurance thresholds for zones (e.g., require signed artifacts, attested environments)
- reject or downgrade artifacts lacking required provenance
- trigger audits when inconsistencies or anomalies are detected
- require remediation (re-signing, re-verification, disclosure, rollback)

Systems SHOULD support:

- escalation pathways tied to specific artifacts or events
- annotation and commentary layers on provenance and logs
- integrity-based gating (e.g., cannot promote, fund, or amplify unverifiable outputs)
- confidence or assurance scoring for artifacts and systems

Failure modes include:

- **non-actionable verification** (signals exist but do not affect outcomes)
- **accountability gaps** (no mechanism to enforce correction)
- **verification fatigue** (users overwhelmed and stop checking signals)

9. Incentives and Power Analysis

Security and provenance exist within strong incentive gradients that often discourage their full implementation.

Key structural tensions include:

- **speed vs. assurance:** teams are rewarded for shipping quickly, not for making systems verifiable
- **opacity vs. accountability:** operators benefit from controlling logs and narratives
- **growth vs. integrity:** unverifiable claims can accelerate adoption
- **cost externalization:** security failures impose costs on users and ecosystems, not just builders

Adversarial actors exploit these tensions:

- attackers rely on weak provenance to inject malicious artifacts
- coordinated misinformation exploits unverifiable content
- insiders may bypass controls for convenience or advantage
- AI systems can generate plausible but unverifiable outputs at scale

DP15 requires realignment of incentives such that:

- unverifiable outputs carry reduced trust, reach, or eligibility
- verified artifacts gain preferential treatment in governance, ranking, or funding (DP9 linkage)
- repeated integrity failures trigger governance review or restrictions

Systems SHOULD incorporate:

- cost imposition for unverifiable or tampered artifacts (rate limits, gating)
- rewards for maintaining strong provenance (priority processing, trust weighting)
- visibility into integrity posture (public assurance signals)

Deployment can stall when organizations treat security as a cost with no visible benefit. Implementation planning should make the value of protection and trusted operation concrete, alongside the costs and duties involved, without making respect for participants contingent on a promised financial return.

Failure modes include:

- **security underinvestment** due to misaligned incentives
- **credibility arbitrage** where actors benefit from unverifiable claims
- **attacker asymmetry** where defense costs exceed attack costs

10. Community Signals Informing DP15

- fatigue with unverifiable claims
- demand for signed exports and build transparency
- concern about AI hallucination presented as fact
- calls for meaningful postmortems after failures

11. Foresight and Failure Design

DP15 assumes breach, key compromise, and manipulation attempts will occur.

Systems must predefine response pathways, including key rotation, participant notification, and audit procedures.

11.1 Anticipatory threat-model posture

Verification design under DP15 MUST signal degradation rather than silently fail, and MUST assume, at minimum, the following forward-horizon conditions: (a) quantum-capable adversaries able to break currently deployed signature and hash primitives; (b) AI systems capable of forging, deceiving, and breaking in at human-expert level across security, social engineering, code generation, and protocol exploitation; (c) concentration of attestation, identity, or data infrastructure that creates monoculture risk. These assumptions apply unless and until a successor threat-model revision explicitly supersedes them.

12. Relationship to Other Desirable Properties

- DP1: attribution and appeals
- DP4: integrity of data handling and exports
- DP7: interoperability of signed artifacts
- DP11–DP13: AI traceability and containment
- DP14: transparency practices
- DP16–DP17: roadmap and funding alignment for security

13. Non-Goals and Explicit Boundaries

DP15 does not:

- require all content to be signed or verified
- eliminate all risk
- replace legal or forensic systems
- mandate a specific cryptographic approach

14. Minimum DP15 Alignment

Minimum alignment defines the threshold at which security and provenance become **reliable verification infrastructure rather than optional features.**

A system claiming DP15 alignment MUST satisfy the following conditions:

14.1 Artifact integrity

- Critical artifacts MUST be signed or otherwise integrity-protected
- Systems MUST verify integrity at use time, not only at creation

Failure mode: **implicit trust of unverified artifacts**

14.2 Provenance availability

- High-stakes artifacts MUST carry provenance metadata (origin, transformation, context)
- Systems MUST signal when provenance is missing or degraded

Failure mode: **provenance absence or stripping**

14.3 Verification pathways

- Participants MUST be able to verify claims without specialized expertise
- Systems MUST provide both simple indicators and detailed inspection paths

Failure mode: **unusable verification systems**

14.4 Tamper-evident logging

- Critical events MUST be recorded in tamper-evident logs

- Logs MUST support independent verification or witnessing

Failure mode: **log manipulation or selective disclosure**

14.5 Key and identity integrity

- Signing keys MUST have visible ownership, rotation, and revocation status
- Systems MUST detect and signal compromised or invalid keys

Failure mode: **undetected key compromise**

14.6 Cross-system preservation

- Integrity and provenance MUST persist across exports and integrations
- Systems MUST signal degradation when preservation is not possible

Failure mode: **silent integrity loss across systems**

14.7 AI output traceability

- AI-generated outputs MUST be labeled and include relevant provenance when claims are made
- Systems MUST distinguish attributable vs. non-attributable outputs

Failure mode: **AI attribution ambiguity**

14.8 Auditability and contestability

- Participants MUST be able to inspect, compare, and challenge evidence
- Systems MUST preserve records necessary for dispute resolution

Failure mode: **evidence opacity or loss**

14.9 Adversarial-AI robustness

Systems MUST document, for each verification primitive they rely on, how that primitive is expected to fail under capable-AI conditions and what observable signal the system emits when such failure is detected. Systems MUST NOT treat a primitive that can degrade silently as satisfying DP15.

Failure mode: **verification collapse against capable AI**, where attestation, identity, or witnessing appears to succeed while the underlying primitive

has already been subverted, and no signal surfaces for operators or participants.

14.10 Independence

Systems MUST NOT depend on a single attestation root, identity provider, or witness service for verification of high-stakes claims. Where independence is not yet achievable, the resulting dependency MUST be disclosed together with a documented migration path and a bounded blast-radius statement.

Failure mode: **verification monoculture**, the same structural condition named in Section 3.13, manifesting here as a minimum-alignment obligation that any system claiming DP15 conformance must meet or explicitly disclaim.

Systems that omit verification, provenance, or auditability MUST NOT be considered aligned with DP15.

15. Open Questions and Future Work

- balancing privacy with auditability
- scaling verification without overwhelming users
- cross-border evidence standards
- decentralized witnessing vs. cost tradeoffs
- quantum-resilient cryptography planning

16. Path Toward ML-RFC

- standardize provenance formats for content and AI outputs
- align with emerging supply-chain and identity standards
- develop reference implementations for high-assurance zones

17. Closing Orientation

DP15 is where the Meta-Layer moves from claims to evidence.

Security without provenance is performative. Provenance without security is unreliable.

Together, they create a system where shared reality can be verified, contested, and trusted.

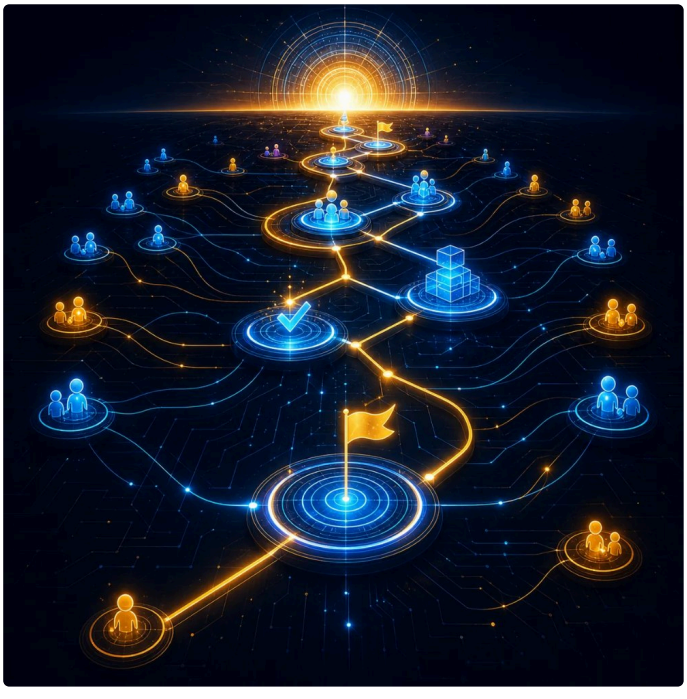
Continue the conversation on DP15 online →

DESIRABLE PROPERTY 16

DP16

Roadmap and Milestones

Honest, resourced, and revisable commitments about what gets built, when, and with what evidence



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp16

1. Purpose of This Draft

This draft articulates Desirable Property 16 (DP16) as the condition under which the Meta-Layer's evolution is honestly communicated and milestone accountability is structural, so participants, builders, and communities can plan, trust, and contest direction without being surprised by silent pivots, vapor schedules, or roadmaps that function as marketing fiction.

DP16 connects DP14 (trust and transparency), DP15 (security posture evolution), DP17 (financial sustainability), DP3 (adaptive governance), DP9 (builder expectations), and DP11 (ethical AI capability claims).

DP16 applies to the initiative's own roadmap as much as to any builder's. When the Meta-Layer promises security, safety, or interoperability (DP15, DP11, DP7), DP16 is how participants can check whether, and when, those promises are real.

If DP16 is weak, predictable failures follow: hype cycles, burned contributors, communities built on assumptions later voided, and regulators correctly skeptical of the entire effort.

DP16 does not promise perfect prediction. It defines minimum honesty conditions for roadmapping in a complex sociotechnical system.

2. Problem Statement

In today's web, roadmaps are often performative rather than operational.

Dates shift silently, features ship without promised safeguards, and "soon" becomes a placeholder for uncertainty rather than a bounded commitment. Participants and builders are expected to plan around signals that are not designed to be reliable.

This produces recurring failures:

- roadmap theater without resourcing or dependency disclosure
- silent removal or reshaping of public commitments
- optimistic AI timelines that misallocate trust (DP11)
- security and privacy work deferred or hidden (DP15)
- milestones dependent on funding that does not exist (DP17)

These failures are structural. Roadmaps coordinate collective action. When they mislead, they coordinate collective harm.

DP16 reframes roadmaps as accountability artifacts: scoped, resourced, dependency-aware, and revisable with memory.

3. Threats and Failure Modes

3.1 Hype-driven timelines

Dates are set for visibility rather than feasibility.

Example: A feature demo launches publicly while required safety systems are not ready.

Why this matters: Capability claims must match operational reality.

3.2 Dependency denial

External constraints are omitted from planning.

Example: Interoperability is promised while critical partner agreements or standards are unresolved.

Why this matters: Honest planning includes what you do not control.

3.3 Moving goalposts without memory

Roadmap changes occur without explanation.

Example: A feature disappears from public documentation without a changelog.

Why this matters: Governance requires traceability of decisions and learning.

3.4 Security and privacy debt hiding

Assurance work is deprioritized in visible planning.

Example: Roadmaps highlight features but omit audit backlogs or known vulnerabilities.

Why this matters: Security posture must be part of the same narrative as feature progress.

3.5 Financial unreality

Milestones assume resources that are not secured.

Example: Public goods infrastructure is promised without funding for maintainers.

Why this matters: Roadmaps must reflect actual capacity, not aspirational intent.

4. Core Principle

Roadmaps and milestones in the Meta-Layer are honest, resourced, and revisable commitments.

Dependencies, risks, and uncertainties are visible. Security, governance, and safety work are treated as first-class alongside features. Changes are explained with memory of what was learned.

Example: A roadmap includes dependency graphs, funded roles, security milestones, and explicit unknowns, with periodic retrospectives explaining changes.

What this feels like: Respect for participant time, attention, and planning.

Without this: Roadmaps become mechanisms for eroding trust rather than coordinating progress.

5. Primary Mechanisms and Structural Conditions

5.1 Dependency-aware planning

Roadmaps explicitly identify legal, technical, governance, and ecosystem dependencies and classify them by **control and certainty** (controlled, shared, external; verified, assumed, unknown).

This includes identifying **critical-path dependencies** that can block delivery, and attaching owners, verification status, and expected update cadence.

Failure mode: **hidden blockers**, where untracked dependencies invalidate timelines after commitments are made.

5.2 Dual-track visibility

Feature delivery and assurance work (security, privacy, accessibility, safety, governance) are presented together with **linked milestones and shared completion criteria**.

Assurance work is not deferred to "later phases" without explicit gating; features that depend on assurance must reference those gates.

Failure mode: **assurance deferral**, where visible progress masks growing risk debt.

5.3 Milestone receipts

Completed milestones link to verifiable evidence such as documentation, code, tests, audits, or third-party attestations (DP15 alignment).

Receipts should be **durable, accessible, and specific to the claimed capability**, enabling independent inspection.

Failure mode: **receipt theater**, where artifacts exist but do not substantiate the claim.

5.4 Change logs with governance salience

Material changes (scope, timelines, removals, reclassification) are recorded with **who decided, why, and who is affected**.

Changes that impact communities or partners should trigger **targeted notifications** and, where appropriate, governance review (DP3 linkage).

Failure mode: **silent pivots**, where affected parties cannot detect or respond to changes.

5.5 Risk registers

Top risks to roadmap integrity are identified, ranked, and updated with **likelihood, impact, mitigation, and owner**.

Risk registers should distinguish between **delivery risk** (can we build it) and **integration risk** (can it work with others) and be visible alongside milestones.

Failure mode: **risk invisibility**, where known issues are not surfaced to participants.

5.6 Scenario planning

Pre-mortems and alternative scenarios are documented to anticipate failure modes, including **best case, expected case, and worst case** timelines.

Scenarios should be tied to triggers (e.g., dependency slips, funding changes) that cause movement between states.

Failure mode: **single-path planning**, where plans assume ideal conditions and lack adaptation paths.

5.7 AI capability boundaries

Clear distinction between research, prototype, pilot, production, and scaled systems, with **explicit gating criteria and evidence thresholds** (DP11 alignment).

Claims about AI capability must reference **evaluation context, limitations, and deployment constraints**.

Failure mode: **capability inflation**, where demos are presented as operational systems.

5.8 Funding linkage

Milestones are tied to known funding sources, staffing, and maintenance commitments, or clearly labeled as contingent.

Roadmaps should indicate **runway, dependency on grants or partners, and maintenance ownership** for shipped work.

Failure mode: **funding ambiguity**, where delivery is implied without resources.

5.9 Cross-organizational coordination

Shared dependencies across organizations are made visible with **common identifiers, ownership boundaries, and divergence signaling**.

Where coordination is loose, roadmaps should indicate **assumption ranges** rather than fixed dates.

Failure mode: **misaligned expectations**, where partners operate on incompatible timelines.

5.10 Retrospective accountability

Regular reviews compare planned versus actual outcomes, capturing **variance, root causes, dependency shifts, and lessons learned**.

Retrospectives should distinguish between **uncertainty, error, and misrepresentation**, and feed updates back into planning and governance.

Failure mode: **non-learning systems**, where mistakes repeat without structural correction.

5.11 Roadmap System Layer: Commitment Integrity, Dependency Enforcement, and Claim Verifiability

Beyond roadmap artifacts and communication practices, DP16 requires a coherent roadmap system layer that ensures commitments remain traceable, constrained, and resistant to misrepresentation under pressure.

Roadmaps are not only planning documents. They are coordination systems that shape participant expectations, builder effort, funding decisions, governance priorities, and public trust. If roadmap claims cannot be bound to scope, evidence, dependencies, and change history, they become narrative instruments rather than infrastructure.

5.11.1 Commitment integrity and binding

Roadmap entries must represent bounded commitments, not open-ended aspirations.

This requires explicit scope, success criteria, completion conditions, and commitment classification, such as research, exploratory, committed, funded, blocked, or deprecated. Claims must remain proportional to the commitment class.

A failure mode is aspirational drift, where language implies commitment while avoiding operational accountability.

5.11.2 Dependency verification and propagation

Dependencies must not only be listed. They must be validated, monitored, and propagated when they change.

This includes technical dependencies, partner dependencies, legal or regulatory dependencies, funding dependencies, security dependencies, and governance dependencies. Critical-path dependencies should be distinguishable from ordinary dependencies so participants understand what can actually block delivery.

A failure mode is dependency illusion, where plans assume conditions that are unresolved, unavailable, or outside the initiative's control.

5.11.3 Claim-to-outcome traceability

Every roadmap claim must resolve to a delivered artifact, a documented change in status, or an explicit retirement.

This requires persistent identifiers for roadmap items and linkage to code, documentation, audits, governance decisions, release notes, or public explanations. Partial completion should be represented as partial completion, not quietly reframed as success.

A failure mode is untraceable claims, where promises cannot be compared against outcomes.

5.11.4 Change integrity and versioned commitments

Roadmaps must preserve versioned history of commitments and changes.

This includes changelogs, previous states, rationale for material shifts, and explanations for additions, removals, timeline changes, or scope changes. Revision is healthy when it is visible; silent revision is trust decay.

A failure mode is historical erasure, where past commitments disappear without record.

5.11.5 Anti-hype and signaling constraints

Roadmaps must resist distortion from marketing, fundraising, competition, and AI capability theater.

Systems should distinguish clearly between concept, prototype, pilot, production, verified capability, and scaled deployment. Public claims should not exceed operational readiness, and uncertainty should be surfaced rather than hidden.

A failure mode is signal inflation, where external incentives distort internal reality.

5.11.6 Cross-organizational roadmap coherence

When milestones depend on multiple organizations, roadmap meaning must remain coherent across boundaries.

This requires shared milestone references, explicit ownership, divergence signaling, and visibility into where one organization's delay, pivot, or withdrawal affects others.

A failure mode is coordination divergence, where different actors present incompatible versions of shared progress.

5.11.7 Roadmap memory and auditability

Participants must be able to reconstruct what was promised, what changed, what was delivered, and what was learned.

This requires accessible archives, comparison views, retrospective links, and evidence artifacts that allow roadmap integrity to be evaluated over time.

A failure mode is roadmap opacity, where evolution cannot be understood, audited, or contested.

This roadmap system layer ensures that coordination signals remain trustworthy under uncertainty, incentives, and scale rather than degrading into narrative management.

6. Milestone Framework

DP16 requires that milestones be structured objects rather than sentences on a slide. A milestone that cannot be classified, scoped, resourced, and closed is a narrative claim, not a commitment.

6.1 Milestone anatomy

Every milestone carries:

- a persistent identifier that survives reorganization and renaming
- scope (in-scope, out-of-scope) and explicit completion criteria

- commitment class and confidence level
- dependency references with control and certainty classification (Section 5.1)
- assurance gates that must clear before completion (Section 5.2)
- funding state and maintenance ownership (Section 5.8, DP17)
- an accountable owner and a next review date
- evidence links once claimed complete (Section 5.3)

Failure mode: **unbounded milestone**, where completion is a matter of interpretation.

6.2 Commitment classes

Public roadmap language must remain proportional to class:

- **exploratory:** under investigation; no delivery implied
- **committed:** scoped, owned, and resourced; delivery expected within a stated window
- **funded:** committed with secured resources and named maintenance ownership
- **blocked:** committed but held by an identified dependency
- **deprecated / withdrawn:** intentionally stopped, with rationale retained

Failure mode: **class inflation**, where exploratory work is communicated in committed language.

6.3 Status transitions and triggers

Transitions between classes are events, not edits. Each transition records who decided, why, what changed, and who is affected, and where material, triggers notification and governance review (Section 5.4, DP3).

Failure mode: **silent transition**, where status changes without a record.

6.4 Completion and partial completion

Completion requires evidence proportional to the claim. Partial delivery is represented as partial, with the remaining scope carried forward under the same identifier rather than absorbed into a success statement.

Failure mode: **rounding up**, where partial work is reported as delivered.

6.5 Milestone receipts and comparison views

Each closed milestone produces a receipt linking claim to artifacts (docs, code, tests, audits, attestations), and roadmap views allow comparison of committed versus current versus delivered state over time (DP15 linkage).

Failure mode: **unverifiable closure**, where completion cannot be independently checked.

6.6 Results chain: from milestones to outcomes

Milestones show whether something was delivered, not whether it made a difference. Each roadmap priority should name the result it serves, at four levels: activity (what is done), output (what is produced), outcome (what changes as a result), and impact (the longer-term effect hoped for). For example, facilitating roadmap workshops (activity) produces a published community roadmap (output), which should help stakeholders coordinate development around shared priorities (outcome), contributing to more democratic and forward-looking governance of the Meta-Layer (impact).

Priorities are set and revisited through governance (DP3) by asking which outcomes matter most and which outputs plausibly lead to them. The assumed link between an output and its intended outcome is recorded with the milestone so it can be tested.

Failure mode: **output substitution**, where completed milestones are reported as success while the intended outcome does not occur.

7. Phased Rollout

Roadmap honesty depends on distinguishing stages of maturity. DP16 requires explicit phases with entry and exit criteria so that demonstrations are not mistaken for deployed capability.

7.1 Phase ladder

- **research:** feasibility and risk exploration; no external reliance expected
- **prototype:** demonstrable behavior in controlled conditions; not for dependence
- **pilot:** limited real participants, bounded scope, monitored, reversible

- **limited production:** available with declared constraints and support commitments
- **general availability:** broadly usable with assurance, support, and maintenance in place
- **sustained / maintained:** funded upkeep, security response, and deprecation policy

Failure mode: **phase skipping**, where pilots are marketed as general availability.

7.2 Entry and exit criteria

Movement between phases requires satisfying declared gates, typically including security and privacy review (DP15), safety and containment posture for AI capability (DP11, DP13), governance readiness (DP3), accessibility, operational support, and funding for the next phase (DP17).

Failure mode: **gate erosion**, where criteria are waived under schedule pressure without record.

7.3 Reversibility and rollback

Each phase declares how it can be paused, narrowed, or withdrawn, including participant notification, data handling on withdrawal, and preservation of records.

Failure mode: **irreversible rollout**, where problems cannot be contained without breaking dependents.

7.4 Communication rules per phase

Public claims are bounded by phase. Demonstrations, benchmarks, and previews are labeled with phase, evaluation context, and known limitations, and cannot imply availability that does not exist.

Failure mode: **demo-to-production drift**, where audiences reasonably infer capability that is not deployed.

7.5 Cohort and zone sequencing

Where rollout proceeds by community, zone, or region, sequencing and selection criteria are published, along with what earlier cohorts are expected

to absorb in risk and what later cohorts gain.

Failure mode: **invisible experimentation**, where participants do not know they are the pilot.

8. Governance, Accountability, and Agency Surfaces

DP16 requires that roadmap signals are not only visible but **actionable and contestable** within governance processes.

Participants must be able to:

- compare promises to delivered outcomes using stable identifiers and evidence links
- understand which dependencies are blocking or delaying work
- distinguish between committed, contingent, and exploratory milestones
- contest misleading, incomplete, or outdated roadmap representations through defined channels

Communities and governance bodies must be able to:

- align decisions (funding, prioritization, standards) with roadmap reality (DP3 linkage)
- trigger reviews when roadmap integrity degrades (e.g., repeated silent changes, missing receipts)
- require correction, reclassification, or withdrawal of misleading claims
- attach **confidence signals** or integrity ratings to roadmap segments

Systems SHOULD support:

- issue/appeal pathways tied to specific roadmap items
- public commentary or annotation layers for milestones
- escalation mechanisms for high-impact discrepancies

Failure modes include:

- **non-actionable transparency**, where information is visible but cannot be used to influence outcomes
- **accountability gaps**, where no actor is responsible for correcting misleading signals

- **participation fatigue**, where communities lack effective recourse and disengage

9. Incentives and Power Analysis

Roadmaps sit inside powerful incentive fields. Teams, funders, partners, contributors, and communities often benefit in the short term from optimistic claims, compressed timelines, and selective disclosure of risk. DP16 exists because those incentives do not naturally produce truth.

Communication incentives often reward optimism over accuracy. A roadmap that looks confident can attract funding, attention, talent, and legitimacy even when the underlying capacity is uncertain. This creates pressure to convert aspiration into implied commitment and ambiguity into promotional momentum.

DP16 requires aligning incentives with truthful forecasting, postmortem transparency, and long-term credibility rather than short-term attention.

Key incentive risks include:

- overstating readiness to attract funding, partners, or adoption
- hiding security, privacy, or governance debt to maintain confidence
- framing dependent milestones as controlled milestones
- using AI capability claims to create urgency without operational evidence
- shifting definitions of success after work has begun

Roadmap integrity therefore requires consequences for repeated misrepresentation. These may include governance review, downgraded confidence ratings, public correction receipts, funding conditions, or reduced eligibility for ecosystem support.

A critical failure mode is credibility arbitrage, where actors gain near-term benefits from inflated claims while distributing long-term costs to contributors and communities.

Roadmap reviews should examine conflicts among the people selecting, funding, and evaluating work. Linked incentives can cause errors to comp-

ound across dependencies, so material relationships belong alongside the existing dependency disclosures and retrospective record.

10. Community Signals Informing DP16

Community signals consistently show that people do not expect perfect prediction. They expect honesty, memory, and respect for the planning burdens created by public commitments.

Recurring signals include:

- frustration with shifting timelines without explanation
- demand for security and safety work to be visible
- skepticism of AI claims in public communication
- desire for funded maintenance commitments
- concern that public promises are used to recruit labor or attention without durable obligation
- need for clear distinctions between pilots, prototypes, production systems, and aspirational futures

These signals point to a deeper pattern: participants can tolerate uncertainty when it is named, but lose trust when uncertainty is converted into certainty for strategic advantage.

DP16 treats these signals as design inputs. Roadmap systems should make uncertainty legible, show dependency reality, and allow communities to distinguish delay caused by honest learning from delay caused by misrepresentation.

11. Evaluation Criteria

Roadmap integrity is measurable. DP16 evaluates whether signals were reliable, not whether predictions were perfect.

Signal reliability is not the same as success. Alongside these integrity measures, roadmaps should report on the outcomes their milestones were meant to advance (Section 6.6), using evidence such as changes in participation, coordination, or trust, even when that evidence is partial or qualitat-

ive. A roadmap can be delivered on time and still fall short if the outcomes it named do not follow.

11.1 Forecast and variance measures

- **variance distribution:** planned versus actual delivery windows, reported without post-hoc rebasing
- **explained variance ratio:** share of slips accompanied by published rationale
- **re-forecast timeliness:** interval between a dependency change and public timeline update

11.2 Dependency and resourcing accuracy

- **dependency disclosure rate:** share of blocking dependencies identified before commitment
- **control classification accuracy:** how often "controlled" dependencies proved external
- **funding-label accuracy:** how often committed work was actually resourced (DP17)

11.3 Evidence and memory measures

- **receipt coverage:** share of completed milestones with verifiable evidence links
- **changelog completeness:** share of material changes with recorded author, rationale, and impact
- **history integrity:** absence of silent removals; prior states remain retrievable

11.4 Assurance visibility

- **assurance ratio:** proportion of roadmap surface allocated to security, privacy, safety, and accessibility work
- **gate adherence:** share of phase transitions that satisfied declared criteria
- **capability claim accuracy:** alignment between stated AI readiness and evaluated capability (DP11)

11.5 Thresholds, triggers, and ratings

Evaluation must be consequential:

- integrity ratings attach to roadmap segments and are visible alongside milestones (Sections 8 and 12.6)
- breaching thresholds (repeated silent changes, missing receipts, sustained unexplained variance) triggers governance review
- correction, reclassification, or withdrawal is required where claims were misleading

Failure modes:

- **accuracy scoring that punishes honesty**, discouraging early disclosure of uncertainty
- **metric gaming** through milestone splitting or scope shrinkage
- **unowned measurement**, where results are published but change nothing

12. Implementation Patterns

These patterns make roadmap integrity operational rather than aspirational. They are illustrative, not mandated.

Each pattern needs a named implementer. Roadmap owners, the teams or organizations that publish a roadmap, maintain the registry, change feed, and status surface. Milestone owners (Section 6.1) keep their items current, record transitions, and publish receipts. Governance bodies (DP3) set thresholds, run reviews when they are breached, and require corrections (Section 11.5). Participants and partners use the comparison views and contest channels (Section 8); they are not expected to maintain the record.

12.1 Roadmap item registry

A structured store where each milestone is a record with stable identifier, class, scope, dependencies, assurance gates, funding state, owner, and evidence links. Public views render from the registry, so presentation cannot drift from the underlying record.

12.2 Dependency graph with propagation

Dependencies are modeled as edges with control and certainty attributes. When a dependency changes state, downstream milestones are flagged automatically for re-forecast rather than silently retaining old dates.

12.3 Dual-track board

A single view presents feature work and assurance work with shared status vocabulary and linked gates, making deferral of security, privacy, or accessibility visible instead of invisible.

12.4 Milestone receipt store

Completed milestones publish receipts referencing documentation, releases, test results, audits, or attestations (DP15), with permanent addresses so future readers can verify past claims.

12.5 Change feed and targeted notification

Material roadmap changes emit an append-only public feed entry and notify affected communities, partners, and dependents, converting silent pivots into observable events.

12.6 Confidence and integrity badges

Roadmap segments display confidence level, funding state, and integrity rating derived from measured history, giving readers a calibration signal at the point of reading.

12.7 Retrospective and pre-mortem templates

Standard formats capture planned versus actual, dependency shifts, distinctions between uncertainty, error, and misrepresentation, and the specific planning changes adopted in response.

Retrospectives also close the learning loop. They revisit the assumptions linking outputs to outcomes, record which held and which did not, and adjust priorities, sequencing, or scope in response. Lessons learned carry into the next planning cycle as explicit changes rather than staying in the retrospective record.

12.8 Public status surface

A durable status page maps current phase, known blockers, open risks, and next review dates, so participants do not need to reconstruct state from announcements.

Anti-patterns to avoid:

- roadmaps maintained only in slides or marketing pages
- status conveyed exclusively through release announcements
- private trackers whose public rendering is manually curated
- date changes applied by editing history rather than recording transitions

13. Foresight and Failure Design

DP16 assumes delays, failures, and shifting conditions. It does not treat change as failure. It treats untracked, unexplained, or strategically obscured change as failure.

Common roadmap failure paths include:

- teams overcommitting to preserve confidence
- security and governance work falling behind visible features
- funding assumptions becoming invalid without public relabeling
- AI capability claims spreading faster than verification
- external dependencies blocking milestones while public timelines remain unchanged

These failures often compound. A single optimistic milestone can attract contributors, partners, and community expectations. If it later shifts without explanation, trust damage spreads beyond the missed milestone and affects the perceived legitimacy of the broader initiative.

DP16 therefore requires recovery practices:

- public status changes with explanation
- retrospectives that distinguish error, uncertainty, and misrepresentation
- updated dependency maps after delays

- repair pathways for communities or builders harmed by misleading signals
- governance review for repeated roadmap integrity failures

A mature roadmap system does not avoid failure. It makes failure learnable, bounded, and repairable.

14. Relationship to Other Desirable Properties

DP16 is a cross-cutting property because roadmap integrity affects whether other desirable properties can be trusted over time.

- **DP3: Adaptive Governance.** Governance needs accurate roadmap signals to sequence decisions, allocate attention, and adjust priorities. If roadmap changes are silent, governance becomes reactive and legitimacy suffers.
- **DP7: Interoperability.** Interoperability depends on shared timelines, version expectations, and dependency clarity. Roadmap drift across systems can break integration even when protocols are sound.
- **DP9–DP10: Builder and learner expectations.** Contributors and learners invest time based on public signals. Roadmap integrity protects that investment from being misdirected by hype or vague commitments.
- **DP11–DP15: AI, security, and provenance alignment.** Capability claims, safety gates, audits, provenance systems, and security posture all require visible sequencing. DP16 ensures assurance work is not hidden behind feature marketing.
- **DP17: Financial sustainability.** Milestones must reflect funding and maintenance reality. Unfunded commitments should be labeled as contingent rather than presented as guaranteed delivery.

If DP16 fails, other DPs can appear stronger than they are. A system may claim future privacy, future AI safety, future security, or future interoperability while coordinating present action around promises that are not structurally accountable.

15. Non-Goals and Explicit Boundaries

DP16 does not aim to eliminate uncertainty or impose rigid planning. It defines what roadmaps are **not allowed to become**.

DP16 does not:

- guarantee precise timelines in uncertain environments; variance is expected when it is explained and bounded
- eliminate the need for confidentiality; sensitive details may be abstracted, but their existence and impact should be signaled
- replace investor, partner, or internal planning norms; it constrains how those norms translate into public commitments
- mandate a single methodology; agile, waterfall, and hybrid approaches are compatible if integrity conditions are met

DP16 also does not permit:

- using ambiguity to imply commitment where none exists
- presenting dependent or unfunded work as controlled delivery
- omitting assurance work to improve perceived velocity
- rewriting history to preserve narrative coherence

Boundary principle:

> *Roadmaps may be uncertain, but they must not be misleading about what is known, what is controlled, and what is promised.*

Failure modes include:

- **methodology masking**, where process language obscures lack of commitment
- **selective disclosure**, where only favorable information is surfaced
- **narrative protection**, where truth is suppressed to maintain external perception

16. Minimum DP16 Alignment

Minimum alignment is the point at which roadmap signals are **reliable enough to coordinate real work**. The aim is not perfection, but to avoid

ematically misleading participants.

Not every condition below needs to be in place on day one. At initial adoption, a roadmap should already bind scope, make key dependencies legible, show assurance work alongside features, label resourcing, signal uncertainty honestly, and keep a changelog; a public document with a dated changelog is enough to start. Evidence for completion applies as each milestone closes, and comparison over time becomes possible after the first review cycle. The fuller mechanisms in Sections 5.11 and 12, such as dependency propagation, integrity ratings, and cross-organizational coherence, can follow as the initiative and its governance capacity mature.

A DP16-aligned initiative should, at minimum:

- **Bind milestones to scope and completion criteria:** Each milestone clearly states what is in/out of scope and what counts as completion (e.g., shipped artifact, audit, or documented capability).
- **Make dependencies legible:** Key dependencies (technical, legal, ecosystem) are identified, with notes on what is controlled vs. external and what is uncertain.
- **Show assurance alongside features:** Security, privacy, and safety work appears in the same roadmap view as features (not deferred or hidden).
- **Provide evidence for completion:** Completed milestones link to verifiable artifacts (docs, code, tests, audits) where appropriate.
- **Maintain change memory:** Material changes (scope, dates, removals) are recorded with brief explanations; prior states are not silently erased.
- **Signal uncertainty honestly:** Distinguish research, prototype, and production readiness (DP11), and avoid presenting demonstrations as deployed capability.
- **Link to resourcing reality:** Indicate whether milestones are funded, partially funded, or contingent (DP17), and avoid presenting unfunded work as committed delivery.
- **Enable comparison over time:** Participants can compare earlier commitments to current status without reconstructing history.

A simple check shows whether minimum alignment has been met. Someone outside the team picks any public milestone and should be able to find its scope, completion criteria, key dependencies, and funding state; a

dated record of every material change; and, if it is marked complete, the evidence behind the claim. At each review, three measures from Section 11 track progress: receipt coverage, changelog completeness, and dependency disclosure rate. These show whether roadmap signals are reliable; whether delivered work achieved its intended outcomes is a separate question.

Failure modes to avoid:

- **Aspirational drift:** commitments implied without binding scope or resourcing
- **Dependency illusion:** plans rely on unresolved external conditions without disclosure
- **Assurance invisibility:** security/safety work omitted from visible planning
- **Historical erasure:** past commitments disappear without record
- **Signal inflation:** capability claims exceed operational reality

17. Open Questions and Future Work

DP16 leaves several questions open because roadmap integrity depends on context, maturity, and governance capacity. The goal is not to impose one planning methodology, but to define the integrity conditions that any credible method must satisfy.

Key open questions include:

- **Transparency versus security:** How should initiatives disclose roadmap risk without revealing vulnerabilities, attack windows, or sensitive partner dependencies?
- **Multi-organization alignment:** What shared formats or receipts allow independent organizations to coordinate milestones without creating centralized control?
- **Forecast accuracy measurement:** How can communities measure forecasting reliability without punishing honest learning, adaptation, or uncertainty disclosure?
- **Regulatory and legal timelines:** How should legal uncertainty, policy change, and compliance review be represented in public roadmap systems?

- **AI capability claims:** What evidence thresholds should distinguish research demos, controlled pilots, production deployments, and frontier capability claims?
- **Funding-contingent work:** How should public goods work be represented when it is necessary but not yet funded?
- **Community recourse:** What should participants be able to do when roadmaps repeatedly mislead or coordination harms accumulate?

These questions should mature through future ML-Drafts, reference implementations, and governance experiments. DP16 should not freeze roadmap practice too early; it should make roadmap integrity inspectable while methods evolve.

18. Path Toward ML-RFC

Advancement from ML-Draft to ML-RFC requires demonstrating that roadmap integrity can be operationalized across real initiatives, not merely described.

Key progression steps include:

- **Standardize roadmap artifact formats:** Define common fields for scope, status, dependency class, funding state, assurance gates, confidence level, and evidence links.
- **Develop milestone receipt patterns:** Establish how completed, delayed, blocked, or retired milestones produce verifiable records.
- **Create retrospective practices:** Define minimum retrospective contents, including planned versus actual outcomes, dependency shifts, and lessons learned.
- **Align communication policies with trust frameworks:** Ensure public claims, demos, and announcements remain consistent with roadmap status and operational readiness.
- **Test multi-organization coordination:** Demonstrate shared roadmap references across independent teams with explicit divergence signaling.
- **Define conformance checks:** Create tests for aspirational drift, dependency illusion, historical erasure, signal inflation, and assurance invisibility.

Promotion to ML-RFC should require evidence that participants can compare commitments over time, verify milestone outcomes, and understand why material changes occurred.

19. Closing Orientation

DP16 is where the Meta-Layer demonstrates respect for time.

Participants, builders, funders, and communities make real decisions based on roadmap signals. They allocate labor, money, trust, attention, and hope. When those signals are inflated, incomplete, or silently rewritten, the harm is not merely reputational. It is coordination harm.

When roadmaps are honest, participants can coordinate effort and trust direction. They can understand what is real, what is uncertain, what is blocked, and what has changed.

When roadmaps are not honest, trust erodes faster than any feature can rebuild it. Even successful delivery becomes suspect when the path to delivery cannot be reconstructed.

DP16 is the claim that the Meta-Layer will not build the future on vapor, hidden debt, or erased commitments.

Roadmaps should not be marketing fiction. They should be public instruments of shared orientation, accountable learning, and coordinated care.

Continue the conversation on DP16 online →

DESIRABLE PROPERTY 17

DP17

Financial Sustainability

Transparent, non-extractive funding that keeps governance, safety, and maintenance alive over time



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp17

1. Purpose of This Draft

This draft articulates Desirable Property 17 (DP17) as the condition under which the Meta-Layer sustains itself over time with transparent, non-extractive, and resilient funding mechanisms, so that governance, safety, maintenance, and innovation can persist without relying on hidden rents, unstable grants, or goodwill that inevitably burns out.

DP17 connects DP6 (commerce and value flow), DP9 (incentives), DP3 (governance capacity), DP15 (security investment), DP16 (roadmap realism), and DP20 (community ownership of surplus).

If DP17 is weak, predictable failures follow: underfunded moderation and safety, stalled infrastructure, dependence on extractive business models, and communities that cannot maintain what they build.

DP17 does not mandate a single funding model. It defines the minimum conditions under which funding is durable, sufficient, and aligned with the Meta-Layer's human-first principles.

2. Problem Statement

In today's web, sustainability is often achieved through indirect or hidden mechanisms: surveillance advertising, data brokerage, platform fees, or speculative token cycles.

Public goods and community infrastructure frequently rely on grants, volunteer labor, or unstable revenue streams, creating a mismatch between system importance and funding reliability.

This produces recurring failures:

- critical infrastructure maintained by underpaid or unpaid contributors
- safety, moderation, and accessibility treated as cost centers rather than core functions
- revenue models that depend on data extraction or attention manipulation (DP4, DP6)
- boom-bust funding cycles that destabilize communities
- governance systems without budget authority or visibility

These failures are structural. Systems that are not sustainably funded cannot remain aligned.

DP17 reframes sustainability as a first-class design constraint, not an afterthought.

3. Threats and Failure Modes

3.1 Hidden extraction as funding

Systems rely on opaque monetization such as data harvesting or behavioral targeting.

Example: A "free" platform funds itself through invasive ad networks without clear disclosure.

Why this matters: Funding models shape system behavior.

3.2 Grant dependency and fragility

Communities rely on short-term funding with no continuity plan.

Example: A civic project shuts down when a sponsor withdraws.

Why this matters: Sustainability requires continuity, not episodic support.

3.3 Misaligned incentives

Revenue depends on metrics that conflict with user well-being.

Example: Engagement-based monetization incentivizes addictive design.

Why this matters: Funding should reinforce, not undermine, system values.

3.4 Invisible cost structures

Participants cannot see how resources are allocated.

Example: Moderation budgets are opaque despite being critical to community health.

Why this matters: Transparency is required for accountability.

3.5 Underfunded maintenance

New features are funded, but upkeep is neglected.

Example: Security patches lag behind feature releases.

Why this matters: Sustainability is predicated on regular and adequate maintenance, not just growth.

3.6 Centralized funding control

A small group controls financial resources.

Example: A foundation allocates funds without community input.

Why this matters: Funding concentration, without tenets and policies that ensure fair distribution, leads to governance capture.

3.7 Token speculation without utility

Financial instruments prioritize speculation over real value creation.

Example: Token prices fluctuate while underlying systems stagnate.

Why this matters: Financialization without function destabilizes ecosystems.

4. Core Principle

Financial sustainability in the Meta-Layer requires that funding sources, allocation, and incentives are transparent, aligned with system values, and sufficient to maintain core functions over time.

Funding must support governance, safety, infrastructure, and community development as first-class concerns.

Example: A community publishes its revenue streams, cost allocation, and funding reserves alongside governance decisions.

What this feels like: Stability and clarity about how the system persists.

Without this: Systems drift toward extraction or collapse under their own weight.

5. Primary Mechanisms and Structural Conditions

5.1 Transparent revenue models

Funding sources are disclosed in plain language with sufficient detail to distinguish between direct user payments, grants, subsidies, data-driven revenue, and indirect monetization.

Systems MUST enable participants to understand not just *what* funds them, but *how those mechanisms shape behavior* (DP9 linkage).

Failure mode: **hidden extraction**, where funding depends on opaque or misrepresented practices.

5.2 Budget visibility

Communities can see how funds are allocated across functions, including governance, infrastructure, safety, and growth.

Budgets SHOULD be presented with meaningful granularity (not overly abstracted) and updated over time.

Failure mode: **allocation opacity**, where critical spending (e.g., moderation, security) is obscured.

5.3 Sustainable incentive alignment

Revenue models reinforce, rather than undermine, user well-being and system integrity.

Systems MUST assess whether revenue depends on behaviors that degrade trust, safety, or agency.

Failure mode: **incentive inversion**, where revenue grows as system quality declines.

5.4 Maintenance funding

Budgets explicitly allocate resources for upkeep, security, and support, including long-term maintenance of previously delivered features.

Maintenance MUST be treated as a first-class financial obligation, not residual spending.

Failure mode: **maintenance collapse**, where systems degrade due to underfunded upkeep.

5.5 Diversified funding streams

Systems avoid reliance on a single funding source by maintaining multiple revenue streams or contingency pathways.

Diversification SHOULD reduce exposure to abrupt funding shocks.

Failure mode: **single-point funding failure**, where loss of one source destabilizes the system.

5.6 Community participation in budgeting

Members have input into how funds are used, with mechanisms ranging from consultation to direct voting or delegated authority (DP3, DP8 alignment).

Participation SHOULD be proportional to stake, responsibility, or role.

Failure mode: **financial disenfranchisement**, where communities bear impact without influence.

5.7 Reserve and contingency planning

Financial buffers exist for unexpected events, including downturns, attacks, or infrastructure failure.

Reserves SHOULD be visible and tied to risk scenarios.

Failure mode: **shock fragility**, where systems cannot absorb disruption.

5.8 Funding–roadmap linkage

Milestones are tied to secured or clearly labeled contingent funding (DP16 alignment).

Roadmaps MUST NOT imply delivery where funding is uncertain or absent.

Failure mode: **funding illusion**, where commitments exceed resources.

5.9 Ethical monetization boundaries

Certain revenue practices may be restricted or banned within zones, especially those that violate privacy, agency, or community norms.

Systems SHOULD define these boundaries explicitly.

Failure mode: **boundary erosion**, where harmful practices re-enter through exceptions.

5.10 Public goods/infrastructure support mechanisms

Infrastructure critical to the ecosystem receives dedicated funding, including shared services that do not generate direct revenue.

Funding SHOULD be stable and not solely dependent on voluntary contributions.

Failure mode: **commons neglect**, where essential systems degrade due to lack of direct monetization.

5.11 Financial System Layer: Flow Integrity, Allocation Enforcement, and Alignment Verification

Beyond individual mechanisms, DP17 requires a coherent financial system layer that ensures money flows are **visible, constrained, and aligned with system values under pressure**.

Financial systems are not neutral. They shape incentives, power distribution, and long-term system behavior. Without enforceable structure, funding drifts toward extraction, concentration, or instability.

5.11.1 Financial flow visibility

All significant flows of value (revenue, grants, fees, distributions) must be observable at an appropriate level of abstraction.

This includes inflows, outflows, and internal transfers between functions.

Money is only one of the flows that keep the system alive. Viewed as a living system, the Meta-Layer also runs on contributor time, attention, trust, and care. Sustainability review should trace where these enter, where they turn into value, where they return to communities, and where they leak or pool. A flow that is steadily drawn down without being replenished, such as volunteer moderation or maintainer goodwill, is a sustainability risk even when the budget balances.

Failure mode: **flow opacity**, where value movement cannot be understood or audited.

5.11.2 Allocation integrity and constraints

Funds allocated to specific purposes (e.g., security, moderation, infrastructure) must remain bound to those purposes unless explicitly reauthorized.

Systems SHOULD define constraints or policies on reallocation.

Failure mode: **allocation drift**, where resources are silently redirected.

5.11.3 Alignment verification

Funding sources and allocations must be evaluated against system values and declared principles.

This includes detecting:

- dependence on extractive revenue
- funding that introduces governance conflicts
- misalignment between stated goals and financial incentives

Failure mode: **alignment illusion**, where funding appears compatible but introduces hidden distortion.

5.11.4 Power distribution and concentration detection

Financial systems must surface concentration of control over resources.

This includes visibility into who controls allocation decisions and how influence accumulates.

Failure mode: **financial capture**, where a small group controls system direction through funding.

5.11.5 Cross-system financial coherence

When multiple systems interact, financial commitments and dependencies must remain consistent and non-contradictory.

Failure mode: **coordination divergence**, where funding assumptions differ across actors.

5.11.6 Financial memory and auditability

Participants must be able to reconstruct funding history, including past allocations, changes, and rationales.

This requires durable records and accessible reporting.

Failure mode: **financial opacity over time**, where history cannot be audited.

5.11.7 Adversarial resilience

Financial systems must anticipate manipulation, including:

- speculative distortion
- subsidy gaming
- governance capture through capital

Systems SHOULD include detection, signaling, and mitigation mechanisms.

Failure mode: **financial exploitation**, where adversaries extract value or influence.

This financial system layer ensures that sustainability is not just declared, but structurally maintained across time, incentives, and scale.

6. Funding Models

DP17 does not prescribe a funding model. It requires that whichever models are used be disclosed, evaluated against system values, and combined so that no single dependency can quietly capture or collapse the system.

6.1 Model catalogue and tradeoffs

- **Direct payment (subscriptions, usage fees):** clearest alignment between payer and beneficiary; risks exclusion and paywalled participation
- **Membership and cooperative dues:** durable and governance-aligned; risks under-provisioning relative to infrastructure cost
- **Grants and philanthropy:** enables public-interest work; risks episodic funding and donor priority drift (Section 3.2)
- **Protocol, zone, or marketplace fees:** scales with activity; risks rent extraction if unchecked (DP6)
- **Commercial services and support:** funds maintenance from those who capture most value; risks two-tier quality
- **Sponsorship and underwriting:** flexible; requires disclosure boundaries and non-influence guarantees

- **Commons funding (matching, quadratic, retroactive rewards):** funds shared infrastructure; risks popularity bias over criticality
- **Endowment and reserve income:** stabilizes maintenance; requires transparent stewardship and drawdown rules (Section 5.7)
- **In-kind and institutional hosting:** lowers cost; creates dependency that must be disclosed as a risk
- **Dual-entity model:** a for-profit affiliate runs legitimate businesses, such as clean energy, daily games, or data-center services, and directs part of its surplus to the public or semi-public organization that stewards the Meta-Layer; gives maintenance and assurance a stable base that does not depend on grants, but risks relationship co-mingling and conflicts of interest. The affiliate's own revenue must respect the boundaries in Section 6.5, and control of funding must not become control of governance (Section 5.11.4).

Failure mode: **model opacity**, where participants cannot tell which of these actually pays for the system.

6.2 Selection criteria

Model choice is evaluated against behavioral consequence, not revenue alone: whether the model rewards degradation of trust or agency, whether it funds non-visible assurance work, whether it concentrates control, whether it survives a downturn, and whether it is legible to the people it depends on (Sections 5.1 and 5.3).

Failure mode: **revenue-first selection**, where behavioral consequences are assessed after commitments are made.

6.3 Portfolio construction

Sustainability comes from blended, deliberately structured funding: a stable base for maintenance and assurance, variable revenue for growth, reserves for shocks, and declared concentration limits per source (Sections 5.5 and 5.7).

Failure mode: **accidental portfolio**, where the mix is the residue of opportunity rather than a design choice.

6.4 Contributor and maintainer compensation

Funding models must account for the people who sustain the system: maintainers, moderators, stewards, translators, and support roles, with visible compensation pathways rather than reliance on volunteer surplus (Section 5.4, DP20).

Failure mode: **volunteer subsidy**, where core infrastructure depends on uncompensated labor until it fails.

6.5 Disallowed and constrained models

Some models are incompatible with the Meta-Layer's stated values: surveillance advertising, undisclosed data brokerage, dark-pattern conversion funnels, and revenue that grows as safety or agency declines (Section 5.9, DP4).

Failure mode: **boundary laundering**, where prohibited models re-enter through partners, intermediaries, or exceptions.

6.6 Transition and wind-down planning

Funding models change. Each model declares how a transition would be handled: notice periods, service continuity, data portability, and honest relabeling of commitments that depended on it (DP16).

Failure mode: **cliff dependency**, where the end of one funding source ends the system.

6.7 Compensated privacy relay operation

Privacy-enhancing relay operation can be a compensated contribution to shared infrastructure. Optional payments, community funding, or mixed arrangements may help sustain a routing mesh without surveillance revenue or coercive monetization.

The model's viability must be demonstrated through costs, service quality, and resistance to operator concentration or incentive gaming. Relay participation is a possible revenue-generating role, not a guarantee that the mesh can support itself without external funding. DP6 addresses optional service pricing and DP4 evaluates the protection delivered.

7. Token Ecosystems

Token mechanisms are permitted but not privileged. DP17 treats them as one financial instrument among many, subject to stricter disclosure because their failure modes are fast, reflexive, and difficult to reverse.

7.1 Utility grounding

Any token must have a defined function in the system beyond price appreciation: access, coordination, contribution accounting, resource metering, or settlement. Function precedes issuance.

Failure mode: **function retrofitting**, where utility is invented after distribution to justify existence (Section 3.7).

7.2 Governance separation

Financial holding must not silently convert into governance authority. Where tokens inform governance, DP17 requires caps, delegation limits, reputation or contribution weighting, and protected decisions that capital cannot override (DP3, DP12).

Failure mode: **plutocratic drift**, where legitimacy tracks holdings rather than participation.

7.3 Allocation and vesting disclosure

Initial allocations, treasury share, team and investor portions, vesting schedules, unlock cliffs, and incentive emissions are published before participants take on exposure.

Failure mode: **hidden overhang**, where undisclosed unlocks transfer value away from later participants.

7.4 Treasury transparency and constraints

Treasury holdings, denomination and volatility exposure, runway, allocation policy, and reallocation authority are visible and bounded, with assurance budgets ring-fenced from market conditions (Sections 5.11.2 and 13).

Failure mode: **treasury opacity**, where reserves are claimed but neither verifiable nor constrained.

7.5 Speculation containment

Systems limit reflexive dynamics that convert price movement into product direction: no incentive programs that reward volume without utility, no roadmap commitments contingent on token price, and explicit separation between market communications and roadmap claims (DP16).

Failure mode: **price-driven roadmapping**, where market attention sets engineering priorities.

7.6 Exit, redemption, and continuity

Participants must understand what the instrument entitles them to, what it does not, how to exit, and what happens to their access, contributions, and records if the token mechanism is wound down.

Failure mode: **trapped participation**, where exit costs are discovered only under stress.

7.7 Compliance and jurisdictional posture

Regulatory classification, jurisdictional constraints, tax treatment, and consumer protection obligations are addressed explicitly rather than deferred, including how obligations differ across participating regions.

Failure mode: **regulatory improvisation**, where compliance risk is transferred to participants.

7.8 Non-token alternatives

Where a token adds financial risk without functional necessity, DP17 expects the simpler mechanism. Credits, fees, memberships, and pooled funds are legitimate and often preferable answers to the same coordination problem.

Failure mode: **instrument inflation**, where tokenization is adopted for fundraising narrative rather than system need.

8. Governance, Accountability, and Agency Surfaces

DP17 requires that financial signals are not only visible but **actionable, contestable, and enforceable** within governance (DP3, DP8).

Participants MUST be able to:

- inspect funding sources, allocations, and changes with stable identifiers
- understand which costs are fixed, variable, contingent, or underfunded
- trace how funding decisions affect safety, infrastructure, and roadmap commitments (DP15, DP16)
- challenge misleading or incomplete financial representations through defined processes

Communities and governance bodies MUST be able to:

- **approve, reject, or reallocate** budget segments tied to core functions
- trigger **integrity reviews** when signals indicate misalignment (e.g., hidden extraction, maintenance underfunding)
- require **reclassification** of funding (committed vs. contingent) when conditions change
- attach **confidence or integrity ratings** to funding streams and budget areas

Systems SHOULD support:

- issue/appeal pathways bound to specific budget lines or revenue streams
- public annotation/commentary layers for financial artifacts
- escalation paths for high-impact discrepancies (e.g., safety budget cuts)

Enforcement pathways:

- downgrade or restrict features dependent on misrepresented funding
- block roadmap commitments that lack verified funding (DP16 linkage)
- initiate governance votes or audits for repeated integrity failures

Failure modes:

- **non-actionable transparency** (visible but no leverage)
- **accountability gaps** (no actor responsible for correction)
- **governance capture** (funding control overrides community decisions)

9. Incentives and Power Analysis

Financial systems create and amplify power. DP17 addresses how incentives **distort behavior under pressure** and how to realign them with system integrity (DP9).

Key dynamics:

- **Extraction loops:** revenue tied to attention, data, or speculation drives behaviors that degrade user welfare
- **Underinvestment in non-visible work:** safety, maintenance, and governance lose funding to growth-visible features
- **Capital concentration:** large funders gain outsized influence over priorities and policy
- **Signaling games:** optimistic financial narratives attract resources despite weak fundamentals
- **AI hype amplification:** capability claims accelerate funding without corresponding verification (DP12, DP15)

Adversarial/attack surfaces:

- **subsidy gaming:** actors exploit grants or incentives without delivering durable value
- **governance capture via capital:** funding conditions steer decisions against community interest
- **speculative distortion:** token or asset prices drive priorities away from utility
- **cross-subsidy masking:** losses hidden behind unrelated profitable lines

Alignment requirements:

- revenue must not increase as safety, trust, or agency decrease
- funding for assurance (security, privacy, governance) must be **protected and non-optional**
- incentives should reward **durable contributions and maintenance**, not just growth metrics

Mitigations:

- caps or checks on concentration of funding control

- ring-fenced budgets for safety/maintenance
- disclosure of incentive structures alongside revenue streams
- independent audits or attestations for high-impact funding claims

Failure modes:

- **incentive inversion** (harmful behavior is profitable)
- **credibility arbitrage** (short-term gains from inflated claims)
- **capture dynamics** (decision power tracks capital rather than legitimacy)

10. Community Signals Informing DP17

Community signals are not just sentiment; they are **early indicators of financial misalignment and coordination risk**. DP17 treats these signals as inputs to system design and ongoing monitoring.

Observed signals and implications:

- **Concern about extractive monetization** → signals hidden revenue dependencies; requires clearer revenue classification and disclosure (Sections 5.1 and 5.11.3)
- **Demand for fair compensation** → indicates gaps in value distribution; requires explicit allocation policies and contributor compensation pathways (Sections 5.2 and 6.4)
- **Frustration with unstable funding cycles** → indicates weak reserves and diversification; requires contingency planning and runway visibility (Sections 5.5 and 5.7)
- **Desire for community-controlled resources** → indicates legitimacy gaps; requires participatory budgeting and governance linkage (Section 8)
- **Skepticism of "free" services** → indicates perceived hidden costs; requires explicit mapping from revenue to behavior (Sections 5.1 and 13)
- **Concern about safety underfunding** → indicates risk externalization; requires ring-fenced budgets for assurance (Sections 5.4 and 9)

Operationalization:

- Convert recurring signals into **metrics** (e.g., % budget to maintenance, % revenue from non-extractive sources)

- Attach **confidence ratings** to funding streams based on signal alignment (Section 8)
- Feed signals into **governance triggers** (e.g., review when safety budget falls below threshold)

Failure modes:

- **signal neglect** (warnings ignored until failure)
- **performative response** (surface-level changes without structural correction)

11. Foresight and Failure Design

DP17 assumes systems will face **financial stress, growth pressure, and adversarial manipulation**. The goal is not to avoid failure, but to make it **visible, bounded, and repairable**.

Common failure paths:

- **overcommitment:** promises exceed funding; leads to roadmap drift (DP16)
- **assurance deferral:** safety and maintenance lag behind features
- **funding shocks:** sudden loss of revenue without reserves
- **capture events:** concentration of capital redirects priorities
- **speculative cycles:** attention shifts to price over utility

Recovery practices:

- reclassification of commitments (committed → contingent)
- budget reallocation with public rationale
- activation of reserves and contingency plans
- governance review and potential rebalancing of power
- public postmortems linking funding decisions to outcomes

A mature system treats financial failure as a **learning loop**, not a hidden defect.

12. Relationship to Other Desirable Properties

DP17 is a cross-cutting layer that conditions whether other DPs remain viable over time.

- **DP6 (Commerce):** Defines revenue mechanisms; DP17 ensures those mechanisms are transparent and non-extractive.
- **DP9 (Incentives):** Aligns economic signals with desired behaviors; DP17 verifies that alignment holds under pressure.
- **DP3 (Governance):** Requires budget authority and visibility to make legitimate decisions; DP17 ensures governance approaches maintain and align with financial sustainability.
- **DP15 (Security & Provenance):** Security work must be funded and evidenced; DP17 ensures it is not starved.
- **DP16 (Roadmaps):** Commitments must match funding reality; DP17 binds milestones to resources.
- **DP20 (Ownership & Surplus):** Determines how value returns to communities; DP17 ensures flows are visible and fair.

If DP17 fails, other DPs degrade into promises without resources.

13. Non-Goals and Explicit Boundaries

DP17 does not prescribe a single economic ideology or eliminate markets. It defines **hard boundaries on misleading or harmful financial behavior** in the Meta-Layer.

DP17 does not:

- mandate a single funding mechanism or ownership model
- eliminate profit, competition, or market dynamics
- guarantee equal distribution of funds across participants
- replace legal or regulatory financial frameworks

DP17 explicitly disallows:

- **hidden extraction** (undisclosed data monetization, dark patterns tied to revenue)

- **misrepresented funding states** (presenting contingent or unfunded work as committed)
- **assurance starvation** (underfunding security, privacy, or governance to inflate growth)
- **history rewriting** (removing or obscuring past funding decisions)
- **cross-subsidy masking** (hiding losses or risks behind unrelated profitable lines without disclosure)

Boundary principle:

> *Financial systems may be complex, but they must not be misleading about what is funded, who controls it, and how it shapes behavior.*

Failure modes:

- **methodology masking** (jargon obscures lack of commitment)
- **selective disclosure** (only favorable financial information is shown)
- **narrative protection** (truth suppressed to maintain perception)

14. Minimum DP17 Alignment

Minimum alignment defines the threshold at which financial signals are **reliable enough to coordinate real work**. Below this threshold, systems may publish numbers but do not provide trustworthy sustainability.

A DP17-aligned system should, at minimum:

- **Bind revenue to behavior:** disclose sources and how they influence system actions (DP9)
- **Show allocation with purpose:** publish major cost categories with sufficient granularity (governance, safety, infra, growth)
- **Fund core functions:** maintain explicit, protected budgets for maintenance, security, and governance (DP15)
- **Link funding to commitments:** tie roadmap milestones to secured or clearly contingent funding (DP16)
- **Maintain change memory:** record material budget and funding changes with brief rationale

- **Signal uncertainty:** distinguish committed, contingent, and speculative funding
- **Enable comparison over time:** allow participants to compare past and current funding states without reconstruction

Failure modes to avoid:

- **funding illusion:** commitments exceed available resources
- **allocation opacity:** critical spending hidden or aggregated beyond usefulness
- **commons neglect:** public goods unfunded despite system dependence
- **signal inflation:** financial claims exceed operational reality
- **historical erasure:** prior funding states disappear without record

Systems that omit binding, allocation visibility, or change memory SHOULD NOT be considered aligned with DP17.

15. Open Questions and Future Work

DP17 requires further work to standardize how financial integrity is measured and enforced across diverse contexts.

Key questions:

- **Transparency vs. sensitivity:** How to disclose enough for accountability without exposing competitive or security-sensitive details?
- **Cross-jurisdiction funding:** How to represent taxes, compliance costs, and legal constraints consistently across regions?
- **Measuring alignment:** What metrics best capture alignment between revenue and system values (e.g., % non-extractive revenue, maintenance ratio)?
- **Public goods funding models:** What hybrid models (grants, fees, commons funding) sustain non-market services?
- **AI-era financing:** How to represent costs and risks of AI systems, including compute, safety, and liability (DP12, DP15)?
- **Contributor compensation:** How to fairly allocate value to maintainers and community contributors over time?

- **Recourse mechanisms:** What should participants be able to do when financial signals are repeatedly misleading?

These should be explored through ML-Drafts, pilots, and comparative implementations.

16. Path Toward ML-RFC

Advancement requires operational evidence that financial integrity can be implemented and audited.

Key steps:

- **Standardize financial schemas:** revenue types, budget categories, funding states, and confidence levels
- **Define reporting formats:** periodic disclosures with comparable fields across systems
- **Create verification artifacts:** links from budgets to receipts, audits, and outcomes (DP15)
- **Test governance integration:** demonstrate budget approval, reallocation, and audit pathways (Section 8)
- **Run adversarial tests:** simulate capture, subsidy gaming, and funding shocks
- **Demonstrate cross-system coherence:** shared projects maintain consistent funding signals (DP7)

Promotion criteria should include the ability for participants to **inspect, compare, and challenge** financial claims over time.

17. Closing Orientation

DP17 is where the Meta-Layer demonstrates respect for resources and power.

Participants allocate time, money, and trust based on financial signals. When those signals are opaque or distorted, coordination fails and communities bear the cost.

When financial systems are transparent and aligned, participants can understand what is funded, what is at risk, and how decisions are made. They can act with confidence, adjust expectations, and hold systems accountable.

When financial systems are not, even successful outcomes become suspect, because the path to them cannot be trusted.

DP17 is the commitment that the Meta-Layer will not fund itself through hidden extraction, unstable promises, or erased history.

Financial sustainability is not just survival. It is the ability to **persist with integrity under pressure, incentives, and change**.

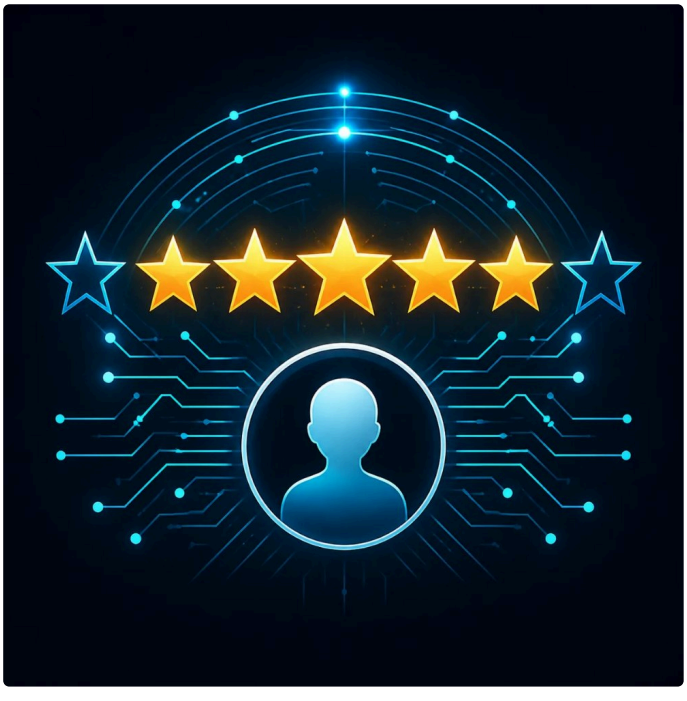
Continue the conversation on DP17 online →

DESIRABLE PROPERTY 18

DP18

Feedback Loops and Reputation

The Conditions for Learning, Recognition, and Repair



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp18

1. Purpose of This Draft

This draft articulates Desirable Property 18 (DP18) as the condition under which the Meta-Layer can learn from participation without collapsing into surveillance, popularity contests, social credit, or reputation systems that permanently trap people in old contexts.

DP18 defines how communities gather feedback, convert signals into accountable learning, and recognize trustworthy contribution over time. It treats feedback and reputation as civic infrastructure, not engagement metrics.

Feedback loops are how the Meta-Layer senses whether its governance, incentives, safety systems, interfaces, and community norms are working. Reputation is how the system remembers patterns of contribution, care, reliability, and harm without reducing people to a single score.

If DP18 is weak, predictable failures follow: communities ask for feedback but do not act on it; reputation becomes a vanity metric; harmful actors launder trust across contexts; contributors burn out because recognition is invisible; AI systems optimize for shallow approval signals; and governance adapts only after legitimacy breaks.

DP18 connects directly to:

- DP1, identity and accountability
- DP2, participant agency and empowerment
- DP3, adaptive governance
- DP4, data sovereignty and privacy
- DP7, interoperability
- DP8, community-defined participation and governance zones
- DP9, developer and community incentives
- DP12 and DP13, AI governance and containment
- DP14, transparency and trust
- DP20, ownership and stewardship

DP18 does not prescribe a universal reputation score, a single feedback interface, or one model of rewards. It defines the minimum conditions under

which feedback and reputation remain contextual, contestable, privacy-preserving, and aligned with community flourishing.

2. Problem Statement

Today's web is saturated with signals, but starved of trustworthy feedback.

Likes, shares, followers, ratings, badges, view counts, and engagement graphs appear to measure social value. In practice, they often measure visibility, habit formation, emotional provocation, automation, or the ability to game a platform's ranking system.

This produces recurring failures:

- communities provide feedback but cannot see whether it changed anything
- bad actors accumulate credibility through volume, performance, or network effects
- positive contributions disappear into feeds without durable recognition
- reputation is trapped inside platforms and lost when people move
- marginalized participants are excluded when feedback systems privilege dominant voices
- AI systems learn from shallow approval signals rather than community-defined values
- governance becomes reactive because weak signals are ignored until crisis

DP18 reframes feedback and reputation as learning infrastructure. Feedback must produce visible adaptation. Reputation must be contextual memory, not global judgment.

A healthy meta-layer must be able to answer:

- What did participants signal?
- Who or what is being evaluated?
- What changed as a result?
- Which signals are reliable in this zone?
- Which signals are being gamed?
- How can a participant contest, repair, or outgrow a reputation state?

- How does the system prevent feedback from becoming surveillance or social punishment?

Without DP18, the Meta-Layer cannot mature. It can collect comments, votes, and metrics, but it cannot reliably learn.

3. Threats and Failure Modes

3.1 Feedback theater

Communities are asked for input, but feedback has no visible path into decisions or operations.

Example: A town hall gathers participant concerns, but the same interface defaults, moderation rules, and incentive structures remain unchanged without explanation.

Why this matters: Feedback without response erodes trust faster than silence because it simulates agency while preserving control.

3.2 Reputation as popularity

Reputation collapses into follower counts, visibility, or applause.

Example: A participant becomes trusted because their posts receive high engagement, even though their contributions are often misleading or inflammatory.

Why this matters: Popularity is not reliability. DP18 requires multidimensional, context-bound reputation.

3.3 Runaway feedback loops

A signal increases visibility, visibility increases signal volume, and the system treats the resulting amplification as legitimacy.

Example: Early upvotes push a submission into prominence; prominence generates more upvotes; the system mistakes attention momentum for quality.

Why this matters: Feedback loops must be bounded. Otherwise reputation becomes a mechanism for self-reinforcing inequality or manipulation.

3.4 Reputation laundering

Actors transfer trust from one context into another where it has not been earned.

Example: A participant with high reputation in a gaming community receives influence in a medical advice zone without relevant credentials or history.

Why this matters: Reputation must preserve context, scope, and provenance. Portability without boundaries becomes a trust exploit.

3.5 Permanent stigma and no repair path

Negative reputation becomes permanent, opaque, or disproportionate.

Example: A participant makes a mistake in one zone and is silently downranked everywhere without notice, appeal, or decay.

Why this matters: Accountability must support learning and repair. Systems that never forgive incentivize abandonment, evasion, and identity cycling.

3.6 Feedback capture by loud minorities

Coordinated or highly resourced groups dominate feedback channels.

Example: A small faction floods surveys, flags, votes, or comments to steer governance toward its preferred outcome.

Why this matters: Feedback systems must distinguish broad legitimacy from coordinated pressure.

3.7 Marginalized voice suppression

Feedback mechanisms reproduce structural exclusion.

Example: Town halls occur in one language and time zone, while reputation rewards favor participants already fluent in dominant cultural norms.

Why this matters: Adaptive governance requires feedback from those most affected, not only those most available or socially rewarded.

3.8 AI-optimized approval hacking

AI systems learn to maximize visible approval signals rather than community-defined value.

Example: An assistant generates emotionally pleasing but low-integrity responses because users reward confidence and fluency more than accuracy.

Why this matters: AI feedback loops must be evaluated against zone policy, truthfulness, safety, and long-term outcomes, not only immediate satisfaction.

3.9 Privacy erosion through reputation telemetry

Reputation systems collect excessive behavioral data to infer trustworthiness.

Example: A system tracks every interaction, dwell time, private message, and social association to generate reputation scores.

Why this matters: Trust cannot be built by violating data sovereignty. DP18 must operate under DP4 constraints.

3.10 Role ossification

Reputation grants access or authority, then authority produces more reputation, making roles difficult to contest.

Example: Early stewards accumulate status and retain control even as their contribution quality declines.

Why this matters: Reputation-linked roles require decay, review, rotation, and contestability.

3.11 Cross-system signal degradation

Feedback and reputation signals move across systems but lose meaning.

Example: A five-star rating exports without information about rubric, rater identity class, zone norms, time horizon, or dispute status.

Why this matters: Interoperable reputation must preserve semantic context or visibly degrade.

4. Core Principle

Feedback loops and reputation in the Meta-Layer must enable communities to learn from lived experience, recognize trustworthy contribution, correct harmful behavior, and adapt governance over time while preserving agency, privacy, context, contestability, and the possibility of repair.

Reputation is not a universal score. It is contextual memory under governance.

This gives DP18 a design principle: reputation as evidence of contribution, not reputation as a bare score. A score that hides its reasons is a verdict, and once people know what produces it, they optimize for the number instead of the work. Evidence of contribution records who did what, when, in what role, and with what effect, and leaves interpretation to the community reading it. Replacing one engagement metric with a better-designed trust score would repeat the failure DP18 exists to prevent.

Feedback is not engagement. It is a structured pathway from signal to learning to action.

That pathway is a closed loop: capture, interpret, decide, act, observe, learn, and adapt. The loop closes only when a participant who gave feedback can see what was done with it, what changed, and whether the change worked.

Example: A community receives recurring feedback that a moderation rule is being applied unevenly. The system clusters the feedback, surfaces affected groups, opens a review process, publishes a decision receipt, updates the rule, and tracks whether the change improves outcomes over time.

What this feels like: Participants can see that their experience matters, not because every request is granted, but because every meaningful signal has a legitimate path into community learning.

Without this: The Meta-Layer repeats the failure of today's platforms: it extracts signals from people while denying them visible influence over the system those signals shape.

5. Primary Mechanisms and Structural Conditions

5.1 Feedback and Reputation Layer: Signal, Memory, Adaptation, and Recourse

DP18 requires a feedback and reputation layer that converts participant signals into governed learning.

This layer includes:

- feedback objects
- reputation objects
- signal provenance
- confidence and uncertainty metadata
- decay and repair mechanisms
- dispute and appeal pathways
- adaptation receipts
- AI-assisted pattern detection under human ratification

The layer must operate at the interface where participants act and are affected, while preserving the governance and privacy boundaries of each zone.

A feedback and reputation layer is not a database of social judgments. It is an accountable learning system.

Failure mode: **social credit drift**, where contextual feedback becomes generalized behavioral scoring.

5.2 Feedback objects

Feedback must be represented as structured objects, not only comments or reactions.

A feedback object SHOULD include:

- subject: what is being evaluated
- signal type: report, endorsement, correction, rating, concern, gratitude, appeal, observation
- scope: zone, artifact, participant, agent, policy, interface, or process
- evidence link, when appropriate

- privacy level
- urgency and severity
- confidence level
- submitter role or credential class, where relevant
- timestamp and version context
- requested action, if any

This allows feedback to be routed, aggregated, audited, and acted upon without flattening all signals into likes or complaints.

Failure mode: **signal mush**, where all feedback becomes undifferentiated noise.

5.3 Feedback routing and response commitments

Feedback systems must define what happens after feedback is submitted.

For each class of feedback, systems SHOULD specify:

- who or what receives it
- expected response time
- escalation path
- visibility rules
- criteria for closure
- appeal or reopening pathway

Participants do not need to control every outcome, but they must be able to see whether feedback entered a legitimate process.

Failure mode: **black-hole feedback**, where reports, suggestions, and concerns vanish without trace.

5.4 Continuous feedback mechanisms

DP18 requires ongoing feedback channels, not one-time consultation.

Mechanisms MAY include:

- surveys
- town halls
- asynchronous deliberation

- embedded interface prompts
- post-decision retrospectives
- safety reports
- participatory audits
- contributor reviews
- community health dashboards
- affected-group listening sessions

Continuous feedback must be designed for accessibility across language, time zone, ability, bandwidth, and technical literacy.

Failure mode: **episodic listening**, where communities are consulted only during crisis or launch moments.

5.5 Adaptive feedback systems

AI and automation may support feedback systems by summarizing large volumes of input, detecting patterns, identifying anomalies, and surfacing underrepresented concerns.

However, adaptive systems MUST remain bounded by:

- visible purpose
- zone-defined policy
- privacy constraints
- human review for material decisions
- auditability of summarization and classification
- clear disclosure of AI involvement

AI may help communities see patterns. It must not quietly decide whose experience counts.

Failure mode: **automation capture**, where AI summaries become de facto governance decisions.

5.6 Reputation as contextual memory

Reputation must be scoped to context, contribution type, and governance zone.

A reputation object SHOULD include:

- subject identity or agent reference
- reputation domain, such as reliability, care, expertise, stewardship, safety, responsiveness, accuracy, generosity, or follow-through
- zone scope
- source signals
- weighting logic
- confidence level
- decay schedule
- dispute status
- portability constraints
- rights or capabilities affected

This prevents reputation from becoming a single global score while still allowing communities to remember meaningful patterns.

Two of these terms need a plain explanation. Zone scope is the community or space where a reputation counts: standing earned in one zone applies there, not everywhere. A decay schedule is the agreed timetable by which older signals count for less unless recent activity renews them. For example, someone known as a careful reviewer in a neighborhood gardening zone carries that standing in that zone, not in a medical advice zone, and a review they did three years ago counts for less than one they did last month.

Failure mode: **global reputation collapse**, where one number pretends to summarize a person, agent, or organization across all contexts.

5.7 Reputation-based compensation and recognition

Reputation may inform compensation, rewards, access, visibility, and recognition, but only under explicit governance.

Reputation-linked compensation SHOULD satisfy the following conditions:

- metrics and rubrics are published
- rewardable contributions are defined
- anti-gaming rules are active
- human review exists for high-value outcomes

- rewards do not depend solely on popularity
- appeals are available
- contributors can understand how reputation affected compensation

Positive contribution should be rewarded without creating a system where people perform for metrics rather than serve the community.

Failure mode: **reputation farming**, where actors optimize for visible reward signals rather than actual value.

5.8 Dynamic role-based access

Roles and capabilities may adjust based on reputation, contribution history, and trust signals.

Examples include:

- moderation privileges
- proposal rights
- voting eligibility
- grant review access
- amplification capacity
- trusted annotator status
- steward responsibilities
- AI-agent deployment permissions

Role changes MUST be:

- explainable
- scoped to a zone
- revocable
- reviewable
- subject to decay or renewal
- protected against sudden manipulation

Reputation may open doors, but it must not create unaccountable hierarchy.

Failure mode: **role capture**, where reputation-linked access becomes permanent power.

5.9 Reputation decay, renewal, and repair

Reputation must change over time.

Systems SHOULD support:

- positive reinforcement for sustained contribution
- decay for inactivity or stale signals
- faster decay for low-confidence signals
- repair pathways after mistakes or sanctions
- restorative processes where communities choose them
- expiry of context-specific penalties
- continued visibility of serious unresolved harms where appropriate

For example, a participant shares a health claim that turns out to be wrong, and a steward marks it as inaccurate. The participant posts a correction. Over the following months their contributions are accurate, the old mark counts for less and less, and after a set period it no longer affects what they can do in the zone. If they had instead caused serious harm that was never addressed, that record would stay visible until it was resolved.

The goal is neither amnesia nor permanent stigma. The goal is governed memory.

Feedback can change what a system remembers and foregrounds, not only a visible reputation score. Relevance decay and changes to a memory graph should remain inspectable so that adaptation does not silently erase inconvenient evidence. DP22 governs continuity and accountable resurfacing.

Failure mode: **frozen reputation**, where old signals dominate present reality.

5.10 Bad behavior reporting and community moderation

Participants and communities must be able to report harm, abuse, manipulation, and bad behavior through structured channels.

Reporting systems SHOULD support:

- evidence submission

- safety-sensitive privacy, including anonymous reporting with a confidential follow-up route for reporters who cannot safely identify themselves
- protection from retaliation
- triage by severity
- pattern detection across incidents, including incidents involving AI agents
- community-defined moderation workflows
- status updates to the reporter as a report moves through review
- appeal and correction mechanisms
- transparency summaries where safe, including anonymized outcomes

Reports are feedback objects with safety consequences. They must be treated with care, not merged into generic sentiment metrics.

Unpopular is not the same as wrong. A view that most participants disagree with must not lose reputation or visibility through downvotes or mass reports alone; reputation responds to evidence of harm or inaccuracy, judged through a stated process with appeal, not to disagreement. Each zone says who decides what counts as accurate, and on what evidence.

Failure mode: **weaponized reporting**, where reporting tools become harassment or governance capture instruments.

5.11 Positive contribution signaling

DP18 requires mechanisms to recognize good behavior, not only punish bad behavior.

Positive signals MAY include:

- gratitude
- endorsement
- attestation
- contribution receipts
- peer review
- successful mediation
- helpful annotation

- quality bridge creation
- accurate correction
- inclusive facilitation
- maintenance work
- long-horizon stewardship

Systems SHOULD make quiet, prosocial labor visible without forcing all care work into performance.

Reputation should recognize stewardship, not popularity. Moderators, reviewers, mentors, facilitators, knowledge curators, translators, documenters, and people who resolve conflicts often create the most value while generating the least engagement. Their work should register as standing in its own right rather than depending on how visible or well-funded they are.

Failure mode: **negative-only memory**, where systems remember harm but fail to recognize repair, support, and stewardship.

5.12 Multi-dimensional reputation

Reputation must be plural.

A participant may be highly trusted for one function and untrusted for another. A contributor may be excellent at technical review but poor at facilitation. An AI agent may be strong at summarization but not authorized for dispute mediation.

Reputation dimensions MAY include:

- accuracy
- reliability
- safety
- expertise
- care
- responsiveness
- collaboration
- originality
- stewardship
- governance judgment

- bridge quality
- conflict resolution
- maintenance reliability

Failure mode: **single-axis trust**, where one form of contribution grants unrelated authority.

5.13 Interoperable reputation with bounded portability

Reputation should be portable where useful, but only with context preserved.

What travels should be the evidence of contribution, not a summary judgment of it. A person may be highly trusted in one community and little known in another. The receiving community should be able to read the underlying contribution history and interpret it against its own context and values, rather than inherit another zone's conclusion. An issuing zone's own assessment travels only as labeled context beside that evidence, never in place of it.

Portable reputation MUST carry:

- issuing zone
- signal type
- evaluation rubric
- timestamp
- decay or expiry
- confidence level
- dispute status
- portability permissions
- limitations on interpretation

When context cannot be preserved, systems MUST signal degradation.

Failure mode: **semantic stripping**, where reputation travels as a badge or score without the meaning needed to interpret it.

5.14 Community health dashboards

Communities SHOULD have dashboards that surface aggregate feedback and reputation patterns without exposing unnecessary personal data.

Dashboards MAY include:

- unresolved feedback volume
- response times
- repeated pain points
- moderation appeals
- participation diversity
- contributor recognition gaps
- role concentration
- reputation distribution
- unresolved harms
- experiment outcomes
- rollback triggers

Two of these items need a plain explanation. A contributor recognition gap is work the community depends on that nobody is credited for, such as the person who welcomes newcomers and answers their questions every week. A rollback trigger is a warning sign agreed in advance that, if it appears, means a change gets reversed, such as a new posting rule that doubles the number of moderation appeals within a month.

Dashboards must be designed to support learning, not surveillance or public shaming.

Failure mode: **dashboard theater**, where metrics are displayed but do not affect decisions.

5.15 Rollback and experiment learning

Feedback loops should support safe experimentation.

When communities test policies, interfaces, incentives, or reputation mechanisms, experiments SHOULD define:

- hypothesis

- success indicators
- harm indicators
- review date
- rollback conditions
- affected participants
- decision authority
- public learning artifact

This allows communities to try new mechanisms without making every experiment permanent.

Failure mode: **irreversible experimentation**, where communities bear the cost of failed changes without recourse.

5.16 Feedback receipts and adaptation receipts

Feedback submissions and system changes should generate receipts.

A feedback receipt MAY include:

- what was submitted
- when it was received
- where it was routed
- privacy and visibility status
- next process step

An adaptation receipt MAY include:

- what changed
- which feedback or evidence informed the change
- who approved it
- expected outcome
- review date
- appeal or contestation path

Receipts close the loop between voice and action.

Failure mode: **untraceable adaptation**, where systems change without participants knowing why.

6. Feedback, Reputation, and AI Systems

AI systems may participate in DP18 in multiple roles:

- as subjects of feedback
- as assistants in feedback processing
- as agents with their own scoped reputations
- as participants in governed zones where permitted
- as tools for detecting manipulation, abuse, and emergent patterns

DP18 requires that AI feedback loops be especially constrained because automated systems can scale signals, interpret signals, and act on signals faster than communities can deliberate.

6.1 AI as subject of reputation

AI agents, models, tools, and automated services SHOULD have scoped reputation profiles when they act in ways that affect participants.

These profiles MAY track:

- accuracy
- refusal appropriateness
- safety incidents
- responsiveness to correction
- policy compliance
- hallucination rates
- data-handling reliability
- emotional safety
- task performance
- escalation behavior

AI reputation must be tied to version, deployment context, and operator accountability.

Failure mode: **agent amnesia**, where an AI system causes repeated harm but each deployment is treated as isolated.

6.2 AI-assisted feedback synthesis

AI may summarize, cluster, translate, and analyze feedback, but must preserve dissent and uncertainty.

Systems SHOULD prevent AI from flattening minority concerns into majority sentiment.

A good synthesis identifies:

- recurring themes
- contested interpretations
- affected groups
- evidence gaps
- urgency
- possible actions
- confidence limits

Failure mode: **consensus hallucination**, where AI creates the appearance of agreement where disagreement remains.

6.3 Feedback loops for AI containment

Feedback from participants can help communities refine containment policies for AI agents.

Examples include:

- reports of manipulative behavior
- consent boundary violations
- failures to disclose automation
- unsafe recommendations
- emotional overreach
- unauthorized cross-zone action
- model behavior drift

Containment feedback should trigger policy review, audit, rollback, rate limits, or suspension where appropriate.

Failure mode: **containment lag**, where AI behavior changes faster than feedback systems can respond.

7. Privacy and Data Sovereignty Requirements

Feedback and reputation systems must comply with data sovereignty.

DP18 systems SHOULD practice:

- data minimization
- purpose limitation
- consent-bound collection
- separation of public recognition and private telemetry
- pseudonymous feedback where safe
- protection for vulnerable reporters
- deletion or expiry where appropriate
- access controls for sensitive reputation evidence
- clear notice when signals affect rights or compensation

Reputation must not become a pretext for continuous behavioral surveillance.

Failure mode: **trust through surveillance**, where the system claims safety by collecting more data than communities can legitimately govern.

8. Governance, Accountability, and Agency Surfaces

Feedback and reputation systems are governance systems. They must themselves be governable.

Communities SHOULD define:

- what counts as valid feedback
- what reputation dimensions matter
- who can issue attestations
- how signals are weighted
- how disputes are handled
- how repair works
- how decay works

- when reputation affects access or rewards
- when human review is required
- how algorithms are audited
- how changes are approved

This governance must be visible, versioned, and contestable.

Some goals in this chapter pull against each other, and each zone states how it balances them: publishing how reputation is calculated against making it easy to game; accountability against the ability to report anonymously; letting people move on from mistakes against keeping serious harm visible.

Failure mode: **hidden reputation law**, where invisible formulas determine social standing and access.

8.1 Accountability surfaces

Governance claims about feedback and reputation must be checkable by the people they affect. DP18 therefore expects named surfaces rather than general assurances:

- a published feedback charter per zone, stating what is collected, who reads it, and what response commitments apply
- a reputation register listing active reputation domains, their issuers, their rubrics, and the decisions they gate
- a decision log linking governance changes to the feedback, evidence, or incidents that prompted them
- a steward roster identifying who reviews disputes, on what timeline, and under whose oversight
- an audit trail for changes to weighting, thresholds, and automated triage
- an escalation route when a steward is themselves the subject of a complaint

Where a system cannot name the surface, it cannot claim the accountability.

8.2 Participant agency surfaces

Accountability without leverage produces informed powerlessness. DP18 requires that participants hold usable controls over their own signals and

standing:

- see which reputation domains describe them, in which zones, and with what evidence
- correct, annotate, or contest an attestation, and see the outcome of that contest
- withhold or scope consent for reputation telemetry that is not necessary for a claimed function
- export contribution history and reputation artifacts under DP4 and DP7 terms
- request attenuation or repair after a defined period, and understand the criteria
- decline reputation-gated roles without losing baseline participation rights

Example: A contributor opens their reputation panel, sees a "reliability" attestation issued by a moderation steward, disagrees with the underlying evidence, files a challenge, and receives a dated decision with reasoning and an appeal option.

What this feels like: Standing is something you can inspect and argue with, not something applied to you.

Without this: Reputation becomes an administrative fact about people that they can neither read nor answer.

Failure mode: **consultation without leverage**, where participants are told how the system works but cannot act on anything they learn.

9. Incentives and Power Analysis

Feedback and reputation systems redistribute power whether or not they intend to. Whoever defines valid signals, sets weights, issues attestations, and controls decay controls who becomes visible, trusted, paid, and promoted.

DP18 therefore treats incentive design as a governance question rather than a tuning question.

9.1 Who gains power from feedback systems

- **Signal definers** decide what counts as evidence. If only structured tickets count, participants who communicate in conversation, voice, or a second language lose standing (DP21, DP23).
- **Attestation issuers** convert observation into durable claims. Concentrated issuance recreates gatekeeping under a new name.
- **Weight setters** determine whose experience moves a decision. Weighting is policy, not math.
- **Interface owners** decide what is easy to report and what requires effort. Friction is an allocation of voice.
- **Aggregators** who hold cross-zone reputation data gain leverage over zones that cannot verify their inferences.

9.2 Predictable incentive distortions

- **Metric substitution:** systems optimize for measurable approval instead of durable contribution, because approval is cheaper to collect.
- **Volume advantage:** participants with time and fluency accumulate reputation faster than participants with equal skill and less availability.
- **Recognition asymmetry:** visible work such as posting accrues reputation while invisible maintenance such as triage, mediation, and care does not.
- **Complaint deterrence:** where reporting is costly or exposing, harm goes unrecorded and the system reads silence as health.
- **Steward overload:** unfunded review capacity turns response commitments into backlog, and backlog into cynicism (DP17).
- **Score arbitrage:** where reputation unlocks compensation, actors invest in signal production rather than in the work the signal was meant to represent.

9.3 Counter-incentives DP18 expects

- fund review, mediation, and repair capacity as core functions rather than volunteer surplus
- credit maintenance and care work explicitly as reputation domains (DP9, DP20)

- weight signals from affected groups rather than from the loudest cohort
- cap the decision weight any single issuer can hold, and rotate issuance
- decay reputation so that standing reflects current participation rather than seniority
- prohibit reputation-linked rewards from depending on signals participants cannot inspect
- publish who benefits from each reputation domain, so that beneficiaries are legible

Example: A zone notices that its "helpfulness" domain is dominated by participants in one time zone. It rebalances by crediting asynchronous mentoring and by rotating the stewards who issue attestations.

Failure mode: **reputation as rent**, where early or well-positioned participants convert accumulated standing into permanent advantage that newer contributors cannot reach.

10. Community Signals Informing DP18

DP18 did not originate as an abstraction. It reflects recurring signals from communities that have lived through extractive feedback and punitive reputation.

Signals repeatedly expressed by participants, moderators, and maintainers include:

- "We answer surveys and nothing changes." Feedback is collected as sentiment, not as a governance input.
- "I don't know why my standing dropped." Reputation effects arrive without explanation or appeal.
- "One mistake follows me forever." Communities want accountability with a path back, not permanent exile.
- "The loudest people set the agenda." Feedback volume is mistaken for feedback representativeness.
- "Moderation work is invisible." Care, triage, and de-escalation carry weight in practice but not in recognition.
- "Reporting is dangerous." Participants withhold reports because exposure risk exceeds expected remedy.

- "My reputation is trapped." Contribution history cannot move between tools, so leaving means starting over (DP7).
- "Scores feel like credit ratings." Participants distinguish sharply between contextual trust and generalized scoring.
- "AI summarized us wrong." Communities want to see, and contest, how synthesis was produced (DP12, DP14).
- "We need to know if it worked." Participants want evidence that changes made in their name improved outcomes.

These signals converge on a single structural request: make the loop complete and make the memory fair. DP18 encodes that request as design conditions rather than as goodwill.

Community signals are inputs to DP18, not ratification of it. They are recorded here so that later revisions can be checked against the concerns that motivated the property.

11. Evaluation Criteria

A DP18-aligned implementation should be evaluated against the following questions.

One test sits above the detailed criteria. Over time, can the system answer three questions: Who has contributed meaningfully, when, and in what way? What has the community learned from those contributions? What changed because of that learning? A system that answers only the first is a reputation mechanism. A system that answers all three is community learning infrastructure.

11.1 Signal quality

- Are feedback signals typed, scoped, and attributable at the right level?
- Can systems distinguish evidence, opinion, endorsement, complaint, and appeal?
- Are low-confidence signals prevented from causing high-impact outcomes without review?

11.2 Loop closure

- Can participants see whether feedback was received, routed, reviewed, and acted upon?
- Do system changes link back to feedback or evidence?
- Are non-actions explained where appropriate?

11.3 Context preservation

- Is reputation scoped to zones and contribution domains?
- Does portable reputation carry rubrics, provenance, and limits?
- Are degraded signals clearly labeled?

11.4 Contestability and repair

- Can participants dispute inaccurate reputation signals?
- Are there appeal timelines and responsible stewards?
- Do reputation states decay, renew, or repair over time?

11.5 Anti-gaming and safety

- Are feedback loops bounded against amplification spirals?
- Are coordinated attacks, sybil behavior, and reputation farming mitigated?
- Are reporting systems protected against weaponization?

11.6 Privacy and proportionality

- Is reputation based on necessary signals rather than totalizing surveillance?
- Are sensitive feedback records protected?
- Can participants understand which signals affect rights, rewards, or access?

11.7 Inclusion

- Can marginalized and less-resourced participants provide feedback meaningfully?
- Are feedback channels multilingual, asynchronous, and accessible?

- Are affected groups visible in system learning without being exposed to retaliation?

12. Implementation Patterns

12.1 Reputation domains instead of reputation scores

Use separate reputation domains for different capabilities rather than one universal score.

Example domains:

- trusted annotator
- reliable bridge builder
- safe moderator
- responsive maintainer
- accurate fact-checker
- inclusive facilitator
- responsible AI operator

12.2 Confidence-weighted signals

Signals should include uncertainty. A verified expert correction, a peer endorsement, an anonymous concern, and a bot-like mass vote should not carry the same weight.

12.3 Receipts everywhere

Feedback, moderation, role changes, rewards, and reputation updates should produce receipts that can be audited without exposing unnecessary private data.

12.4 Decay by default

Signals should age unless renewed by current evidence. Decay prevents permanent lock-in and reduces the power of old or stale interactions.

12.5 Role renewal cycles

Reputation-linked roles should require periodic review or renewal, especially for stewards, moderators, grant reviewers, and high-impact AI

operators.

12.6 Affected-group weighting

Feedback from those directly affected by a policy or harm may require special visibility or routing, while still protecting against capture.

12.7 Deliberative escalation

High-impact reputation changes should move from automated detection to human or community review before affecting access, compensation, or public standing.

12.8 Reputation portability bundles

Portable reputation should export as bundles containing contribution evidence, claims, provenance, rubrics, expiry, dispute status, and interpretation limits.

12.9 Community retrospectives

Communities should periodically review whether feedback loops are actually improving governance, safety, inclusion, and contribution quality.

Feedback serves two purposes: evaluating individuals and helping the community learn. The second is often more valuable. Retrospectives should capture recurring lessons, decisions, corrections, successful practices, and unresolved issues as community memory, so that feedback does not disappear into an activity stream. DP22 (Civic Memory & Epistemic Continuity) governs how that memory is kept and resurfaced.

13. Relationship to Other Desirable Properties

13.1 DP1 – Federated Authentication & Accountability

DP18 depends on accountable identity to prevent sybil abuse, retaliation, and reputation laundering. DP1 provides the identity substrate; DP18 governs the memory of behavior and contribution.

13.2 DP2 – Participant Agency and Empowerment

Participants must understand and contest reputation effects. Feedback systems must produce real agency, not symbolic participation.

13.3 DP3 – Adaptive Governance Supporting an Exponentially Growing Community

DP18 supplies the learning loops that allow governance to adapt. DP3 defines how decisions change; DP18 defines how lived experience becomes actionable signal.

13.4 DP4 – Data Sovereignty and Privacy

Feedback and reputation must operate under data minimization, purpose limitation, and consent. Reputation cannot justify surveillance.

13.5 DP7 – Simplicity and Interoperability

Reputation and feedback must preserve meaning across systems or degrade visibly.

13.6 DP8 – Collaborative Environment and Meta-Communities

Reputation is zone-scoped by default. Communities define which signals matter, which roles reputation unlocks, and how repair works.

13.7 DP9 – Developer and Community Incentives

DP18 provides the recognition and evaluation substrate for fair incentives. DP9 governs how value flows from contribution.

13.8 DP12 and DP13 – AI Governance and Containment

AI agents need scoped reputations, feedback-triggered containment, and human-ratified adaptation.

13.9 DP14 – Trust and Transparency

Feedback loops and reputation logic must be legible enough for participants to understand how trust is being shaped.

13.10 DP20 – Community Ownership

Communities should be able to own and govern their feedback data, reputation schemas, and contribution histories.

13.11 DP22 – Civic Memory & Epistemic Continuity

Feedback records, corrections, and retrospective lessons become community memory. DP22 governs how that memory is kept, attenuated, and resurfaced, including the decay and repair that DP18 depends on (Sections 5.9 and 12.9).

14. Non-Goals and Explicit Boundaries

DP18 is frequently misread as an argument for more measurement. It is not. The following are explicit non-goals, stated so that implementers cannot claim DP18 for systems it was written to prevent.

DP18 does not create a universal reputation score. There is no single number, tier, or trust level that follows a person across the Meta-Layer. Reputation is domain-scoped and zone-scoped by construction, and any system that collapses domains into one general standing has left DP18, not implemented it.

DP18 does not authorize social credit. Reputation may describe contribution within a bounded context. It may not be used to rank persons generally, to condition access to essential services, or to produce behavioral compliance through generalized scoring.

DP18 does not justify surveillance. Feedback and reputation are not a warrant for continuous telemetry. Signals must be necessary for a declared function, minimized, and governed under DP4. "We needed the data to compute trust" is not a defense.

DP18 does not make feedback binding. A signal is a legitimate input to a decision, not a decision. Communities remain free to decline requests, provided the decision and its reasoning are visible. DP18 requires loop closure, not majority rule by complaint volume.

DP18 does not replace governance. Feedback informs governance under DP3 and DP8; it does not substitute for deliberation, rights, or due process. Automated adaptation driven by signals must remain bounded and human-ratifiable.

DP18 does not prescribe one interface, schema, or algorithm. Zones may implement feedback objects, rubrics, and weighting differently. DP18 constrains the properties those implementations must preserve: typing, provenance, contestability, decay, and receipts.

DP18 does not promise neutrality. Any feedback system encodes choices about what counts. DP18 requires those choices to be published, versioned, and contestable rather than presented as objective measurement.

DP18 does not guarantee that harm will be caught. It requires that reporting exist, be safe, and be answerable. Absence of reports is not evidence of absence of harm, and no implementation may treat a quiet dashboard as a health claim.

DP18 does not permit permanent records as punishment. Accountability without a path to repair is exile. Retention that cannot decay, expire, or be contextualized is outside DP18 regardless of accuracy.

DP18 does not extend to identity verification. Establishing who someone is belongs to DP1. DP18 governs the memory of what was contributed and how it was experienced.

DP18 does not require that all reputation be portable. Portability is bounded by rubric fidelity. Where meaning cannot survive transfer, DP18 prefers visible degradation over silent translation (DP7).

15. Minimum DP18 Alignment

A system claiming DP18 alignment SHOULD demonstrate:

- structured feedback objects
- visible feedback routing and response commitments
- reputation scoped by zone and contribution domain
- published reputation logic where it affects access, rewards, or visibility
- decay, renewal, and repair mechanisms
- dispute and appeal pathways
- protection against sybil attacks, brigading, and reputation farming
- privacy-preserving data practices
- AI feedback processing with disclosure and human oversight

- adaptation receipts linking feedback to changes
- safeguards against runaway amplification loops
- explicit interoperability semantics for reputation portability

The following statements describe what each of those conditions means in practice. They are non-normative and are offered as an assessment aid rather than as a conformance test.

15.1 Signal integrity

Feedback is captured as typed, scoped objects with provenance and confidence metadata. Evidence, opinion, endorsement, complaint, and appeal are distinguishable. Low-confidence signals cannot trigger high-impact outcomes without review.

15.2 Loop closure

Every accepted signal has a state a participant can observe: received, routed, under review, acted upon, declined with reasoning, or deferred. Changes to rules or systems reference the signals that informed them.

15.3 Contextual memory

Reputation exists as named domains bound to zones and rubrics. No general-purpose score is exposed. Cross-zone use carries rubric, issuer, and limits, or is labeled as degraded.

15.4 Contestability and repair

Participants can inspect, annotate, and challenge attestations that describe them. Challenges have owners and timelines. Standing decays, renews, and can be repaired under published criteria.

15.5 Proportionate data practice

Collection is minimized and purpose-bound. Public recognition and private telemetry are separated. Reporters have protection appropriate to their risk, including pseudonymous pathways where safe.

15.6 Bounded automation

AI participation in clustering, summarization, or weighting is disclosed, sampled for accuracy, and ratified by humans before it affects rights, compensation, or access.

15.7 Anti-gaming posture

Amplification loops are damped. Coordinated inauthentic signaling, reputation farming, and weaponized reporting have named mitigations and observable incident handling.

15.8 Governance of the loop itself

The rules governing feedback and reputation are published, versioned, and amendable through the zone's governance process, with a record of who changed what and why.

16. Open Questions and Future Work

1. What minimum schema should define a feedback object across the Meta-Layer?
2. What minimum schema should define a reputation object?
3. Which reputation dimensions should be standardized, and which should remain zone-defined?
4. How should reputation decay be represented across systems?
5. What privacy-preserving methods can support reputation without centralized surveillance?
6. How should communities distinguish endorsement, expertise, contribution, care, and popularity?
7. What rights should participants have when reputation affects access, compensation, or visibility?
8. How should non-human agents receive, carry, and contest reputation?
9. What signals are safe to make portable across zones?
10. How can marginalized communities govern feedback without being overexposed or tokenized?
11. What forms of feedback should trigger mandatory governance review?

12. How can reputation-linked roles avoid ossification and capture?
13. What are the rollback standards for failed reputation experiments?
14. How should feedback systems disclose AI summarization, clustering, or weighting?

17. Path Toward ML-RFC

DP18 is currently an ML-Draft and serves as exploratory scaffolding for how feedback loops and reputation can function as accountable learning infrastructure in the Meta-Layer.

Advancement toward ML-RFC status SHOULD require:

- convergence on a minimal interoperable schema for feedback objects
- convergence on a minimal interoperable schema for reputation objects
- demonstrated implementations across multiple zones
- tested mechanisms for decay, repair, and dispute resolution
- validated protections against gaming, sybil attacks, and amplification loops
- evidence of privacy-preserving reputation systems in practice
- working models for AI-assisted feedback synthesis with human oversight
- at least one cross-system portability experiment with semantic preservation
- documented governance processes for evolving reputation logic

ML-RFC promotion SHOULD be contingent on:

- rough consensus across participating communities
- demonstrated real-world usage with measurable learning outcomes
- clear failure cases and mitigation strategies
- alignment with DP1, DP2, DP3, DP4, and DP8 in deployed systems

Early ML-RFC candidates may focus on:

- feedback object standards
- reputation object standards
- decay and repair protocols

- feedback receipt and adaptation receipt formats

DP18 is expected to evolve iteratively, with partial RFCs emerging for specific components rather than a single monolithic specification.

18. Closing Orientation

DP18 makes the Meta-Layer capable of learning.

It ensures that feedback is not extracted and ignored, that reputation is not reduced to popularity, and that communities can recognize contribution without building permanent social cages.

A DP18-aligned meta-layer remembers enough to be accountable, forgets enough to allow growth, and adapts visibly enough for participants to trust that their experience matters.

Feedback becomes civic signal. Reputation becomes contextual memory. Learning becomes shared infrastructure.

This is the difference between a system that listens and one that evolves.

Continue the conversation on DP18 online →

DESIRABLE PROPERTY 19

DP19

Amplifying Presence and Community Engagement

The Conditions for Public Presence Without Capture



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp19

1. Purpose of This Draft

This draft articulates Desirable Property 19 (DP19) as the condition under which the Meta-Layer becomes visible, inviting, culturally resonant, and socially adopted without collapsing into extractive growth, hype cycles, influencer capture, or engagement theater.

DP19 defines how the Metaweb, Overweb, and broader Meta-Layer ecosystem cultivate awareness, shared identity, community-led promotion, and durable participation. It treats amplification not as advertising alone, but as the public-facing expression of a civic substrate.

The Meta-Layer cannot become trustworthy civic infrastructure if people do not understand it, recognize themselves in it, and feel invited to participate. Presence must be amplified through symbols, narratives, tools, communities, public relations, education, and lived experiences that make the Meta-Layer legible.

If DP19 is weak, predictable failures follow: the Meta-Layer remains technically compelling but culturally invisible; public language becomes confused with the metaverse, Web3 speculation, or AI hype; community energy dissipates; early supporters lack ways to help; growth depends on centralized marketing; and adoption becomes shallow because the public never develops emotional or practical familiarity.

DP19 connects directly to:

- DP2, participant agency and empowerment
- DP3, adaptive governance
- DP5, decentralized namespace
- DP7, simplicity and interoperability
- DP8, collaborative environment and meta-communities
- DP9, developer and community incentives
- DP10, education
- DP14, trust and transparency
- DP16, roadmap and milestones
- DP18, feedback loops and reputation
- DP20, community ownership

DP19 does not prescribe a single brand, campaign, influencer strategy, or communications calendar. It defines the minimum conditions under which amplification and engagement remain participatory, truthful, inclusive, and aligned with the Meta-Layer's purpose.

2. Problem Statement

Infrastructure does not become public infrastructure merely by existing.

The web itself became powerful because people learned to recognize its patterns, tell stories about it, build on it, link to it, invite others into it, and imagine themselves inside it. The Meta-Layer requires a similar cultural threshold. People must be able to name it, describe it, share it, experience it, and see why it matters.

Today's web, however, has trained people to associate digital growth with extractive dynamics:

- social media virality
- influencer-driven attention
- engagement farming
- opaque recommendation systems
- growth hacking
- brand performance without substance
- communities treated as marketing funnels
- platform metrics mistaken for public legitimacy

This creates a dilemma. The Meta-Layer must reach people where attention already lives, but it cannot adopt the same attention-extractive patterns it seeks to transcend.

DP19 addresses this by reframing amplification as community presence. The goal is not merely to promote the Meta-Layer, but to help communities recognize, inhabit, and extend it.

A healthy DP19 implementation must be able to answer:

- What is the Meta-Layer called in public language?
- What symbols, stories, and experiences make it memorable?

- How can participants share it without becoming unpaid ad labor?
- How can communities grow adoption while preserving trust?
- How can public relations establish narrative clarity without centralizing control?
- How can youth, families, educators, civic leaders, developers, journalists, and local communities each find a meaningful entry point?
- How can incentives support community-led amplification without rewarding spam or manipulation?
- How can engagement be measured by meaningful participation rather than shallow reach?

Without DP19, the Meta-Layer risks becoming a brilliant architecture with no cultural surface.

3. Threats and Failure Modes

3.1 Narrative confusion

The public cannot distinguish the Meta-Layer from the metaverse, Web3 speculation, browser extensions, social media, AI agents, or generic digital trust projects.

Example: A journalist describes the Metaweb as "another metaverse platform," obscuring its role as an interface-level civic layer above the existing web.

Why this matters: Names and narratives shape adoption. Confusion blocks participation before people encounter the architecture.

3.2 Brand vacuum

The ecosystem lacks memorable names, symbols, metaphors, and public language.

Example: Builders can explain the architecture technically but cannot offer a phrase, image, or story that ordinary people remember.

Why this matters: Public infrastructure needs cultural handles. Without them, people cannot share the idea.

3.3 Hype without substance

Amplification promises more than the system can currently deliver.

Example: A campaign claims the Meta-Layer will "fix the Internet" while usable tools, governance pathways, or safety guarantees remain early-stage.

Why this matters: Overclaiming converts early curiosity into later distrust. DP19 must align with DP16 roadmap honesty.

3.4 Influencer capture

Public narrative becomes dependent on a few high-visibility personalities.

Example: A thought leader becomes the de facto voice of the Meta-Layer, shaping public perception without accountability to participating communities.

Why this matters: Civic infrastructure cannot rely on celebrity mediation.

3.5 Community-as-funnel extraction

Participants are treated primarily as growth channels.

Example: Ambassadors are asked to generate posts, referrals, and events but receive little agency over strategy, messaging, governance, or value flows.

Why this matters: Community-led promotion must be reciprocal, not extractive.

3.6 Spammy bounty dynamics

Incentives reward volume over quality.

Example: A bounty pays for social posts, producing low-effort threads, duplicate graphics, or misleading claims.

Why this matters: Amplification incentives can degrade trust if they reward noise.

3.7 Platform dependency

Engagement depends too heavily on proprietary social platforms.

Example: A movement grows on one platform's algorithm, then loses reach when rules, APIs, ranking, or moderation policies change.

Why this matters: A meta-layer committed to agency and interoperability cannot depend entirely on rented attention.

3.8 Exclusionary adoption pathways

The public face of the Meta-Layer appeals only to technical, crypto, policy, or AI governance audiences.

Example: Parents, teachers, youth, local communities, artists, journalists, and non-technical civic actors cannot find themselves in the story.

Why this matters: Public legitimacy requires plural entry points.

3.9 Gamified engagement without civic depth

Leaderboards, badges, and rewards stimulate activity without building understanding, trust, or stewardship.

Example: Participants chase ambassador points but cannot explain the values, risks, or governance responsibilities of the Meta-Layer.

Why this matters: Engagement is not the same as orientation.

3.10 Public relations centralization

Media and partnership strategy is controlled by a narrow group without transparent accountability.

Example: Official messaging changes to satisfy funders or partners without community visibility or review.

Why this matters: Narrative power is governance power.

3.11 Adoption without retention

People encounter the Meta-Layer once but do not develop ongoing presence, practice, or identity.

Example: A viral campaign drives signups, but participants do not return because the onboarding path is unclear or the experience lacks immediate relevance.

Why this matters: DP19 must produce durable engagement, not only awareness spikes.

4. Core Principle

Amplifying presence and community engagement in the Meta-Layer means helping people recognize, share, inhabit, and steward a new civic layer of digital life through truthful narratives, resonant identity, participatory campaigns, reciprocal incentives, and durable engagement pathways.

Amplification is legitimate only when it increases agency, understanding, and accountable participation.

Engagement is healthy only when it deepens belonging, contribution, and stewardship.

Example: A family-facing campaign introduces the Meta-Layer as a safer, co-creative layer above the web, provides parent and youth guides, invites local workshops, offers badges for meaningful participation, and routes community feedback into governance.

What this feels like: People do not feel marketed to. They feel invited into a shared project.

Without this: The Meta-Layer remains either obscure infrastructure or another brand shouting for attention.

5. Primary Mechanisms and Structural Conditions

5.1 Presence and Engagement Layer: Narrative, Identity, Participation, and Continuity

DP19 requires a presence and engagement layer that makes the Meta-Layer recognizable and participatory across cultural, civic, technical, educational, and local contexts.

This layer includes:

- naming and brand systems
- public narratives
- community-led promotion

- ambassador programs
- social sharing tools
- press and partnership pathways
- bounties and rewards
- educational engagement
- events and rituals
- participation badges
- onboarding journeys
- feedback loops from public engagement into governance

The purpose of this layer is not to manufacture attention. It is to make presence discoverable and meaningful.

Failure mode: **growth skin**, where branding and marketing sit on top of the system without shaping participation, governance, or trust.

5.2 Naming, branding, and memetic clarity

The Meta-Layer requires memorable, symbolic public language that conveys its mission of agency, transparency, trust, and new layers of digital interaction.

Naming systems SHOULD be:

- easy to say
- easy to explain
- visually memorable
- culturally adaptable
- distinct from extractive or dystopian digital narratives
- compatible with decentralized namespace patterns
- capable of supporting both technical and public-facing meaning

Candidate or ecosystem terms may include Metaweb, Overweb, Sky-Web, Canopi, or other community-developed language.

The chosen terms must help people understand that this is a layer above today's web, not a replacement world that pulls them away from reality.

Failure mode: **semantic fog**, where no one can explain what the project is in one sentence.

5.3 Iconic presence and visual identity

The Meta-Layer needs recognizable visual language that can travel across tools, events, documents, interfaces, campaigns, and community spaces.

Visual identity SHOULD support:

- trust without institutional stiffness
- openness without vagueness
- civic seriousness without bureaucracy
- youth resonance without pandering
- global adaptability
- accessibility and readability
- remixability by communities

Visual identity MAY include marks, icons, badges, color systems, mascots, interface motifs, overlay metaphors, and ritual objects.

Failure mode: **corporate flattening**, where the public identity feels like another platform brand rather than a shared civic substrate.

5.4 Narrative architecture

DP19 requires a coherent set of stories for different audiences.

Narratives SHOULD be tailored for:

- general public
- youth and students
- parents and families
- educators
- civic technologists
- journalists
- municipalities
- developers
- AI governance communities
- artists and creators

- funders and institutional partners
- local organizers

Each narrative should preserve the same core claim while meeting the audience where it is.

Example core claim:

The Meta-Layer lets people and communities add trust, context, presence, and governance above the web they already use.

Failure mode: **audience collapse**, where one technical narrative is expected to work for everyone.

5.5 Community-driven marketing

Participants should be able to help amplify the Meta-Layer through meaningful, governed contribution pathways.

Community-driven marketing MAY include:

- ambassador programs
- local organizer kits
- event hosting
- explainers and videos
- memes and cultural artifacts
- translation and localization
- youth-led campaigns
- school and university clubs
- community showcases
- civic workshops
- creator collaborations
- onboarding circles

These efforts SHOULD be reciprocal. Contributors should receive recognition, feedback, learning opportunities, and where appropriate, compensation.

Failure mode: **volunteer extraction**, where enthusiasm is used without support, recognition, or voice.

5.6 Ambassador and steward programs

Ambassador programs can help the Meta-Layer travel across regions, languages, sectors, and communities.

An ambassador program SHOULD define:

- role expectations
- community accountability
- messaging guidelines
- onboarding curriculum
- escalation paths
- recognition systems
- conflict-of-interest rules
- renewal or sunset processes
- feedback responsibilities

Ambassadors should not merely promote. They should listen, translate, connect, and help communities enter governance pathways.

Failure mode: **brand reps without governance**, where ambassadors broadcast messages but cannot shape the system.

5.7 Social media integration

Participants may need tools to share Metaweb experiences and insights on existing social platforms such as X, Instagram, LinkedIn, TikTok, YouTube, Mastodon, Bluesky, and others.

Cross-posting tools SHOULD support:

- participant consent
- context preservation
- source links
- provenance markers
- audience selection
- platform-specific formatting
- privacy review before posting
- clear distinction between personal expression and official messaging

Social sharing should bring people into deeper context, not strip context for reach.

Failure mode: **context collapse marketing**, where meaningful Metaweb activity becomes flattened into promotional snippets.

5.8 Public relations and media strategy

DP19 requires proactive public narrative work.

Public relations MAY include:

- press kits
- explainers
- op-eds
- interviews
- journalist briefings
- public demos
- media partnerships
- family media outreach
- civic technology features
- education-sector stories
- thought-leader engagement
- crisis communication protocols

PR materials SHOULD be truthful about maturity, limitations, and roadmap status. They should distinguish aspiration from operational reality.

Failure mode: **PR overclaiming**, where media attention creates expectations the system cannot yet satisfy.

5.9 Partnerships with trusted messengers

The Meta-Layer can grow through partnerships with trusted communities and institutions.

Potential partners include:

- educators
- libraries
- municipalities

- newsrooms
- parent networks
- youth organizations
- civic tech groups
- Internet governance communities
- digital safety organizations
- creator communities
- open-source networks
- local cultural institutions

Partnerships SHOULD preserve community agency and avoid converting trusted messengers into sales channels.

Failure mode: **borrowed trust abuse**, where institutions lend legitimacy to systems they cannot meaningfully govern or understand.

5.10 Bounties, badges, and rewards

Incentives may support amplification, onboarding, content creation, translation, event organization, and public education.

Reward systems SHOULD:

- define valuable contributions clearly
- reward quality over volume
- distinguish official, community, and experimental materials
- avoid misleading claims
- include review processes
- disclose conflicts of interest
- connect to DP18 reputation logic
- support non-monetary recognition where appropriate
- avoid leaderboards that reward spam

Examples of rewardable actions:

- high-quality explainer video
- local workshop
- multilingual translation

- school onboarding guide
- trusted press contact development
- civic demo night
- documentation improvement
- community case study
- accessibility adaptation

Failure mode: **bounty spam**, where incentives produce noise instead of understanding.

5.11 Gamification with guardrails

Gamification can motivate participation, but it must be carefully bounded.

Healthy gamification MAY include:

- learning quests
- contribution badges
- local chapter milestones
- collaborative challenges
- public good achievements
- stewardship streaks
- peer recognition
- community showcase unlocks

Gamification SHOULD NOT reward harassment, spam, superficial virality, or unhealthy attention competition.

Failure mode: **casino engagement**, where the system trains people to chase points rather than build civic capacity.

5.12 Youth and educational familiarity

Long-term adoption depends on familiarity during formative learning contexts.

DP19 SHOULD support youth and education pathways through:

- school-safe explainers
- classroom activities

- youth ambassador programs
- digital literacy curricula
- annotation and bridge-building exercises
- civic Internet labs
- student-led showcases
- family-facing onboarding
- partnerships with educators and youth media

The goal is not to recruit children into a product. The goal is to cultivate agency, trust literacy, and co-creative expectations for the web.

Failure mode: **youth capture**, where young people are targeted for growth instead of empowered as future stewards.

5.13 Family-centered engagement

Families can become trusted entry points into the Meta-Layer when the story emphasizes safety, creativity, agency, and shared digital care.

Family-centered engagement MAY include:

- parent guides
- youth safety explainers
- co-creation activities
- family media outreach
- community workshops
- safer web campaigns
- intergenerational digital literacy sessions

Family media outreach can use press kits, parenting articles, and youth-oriented features to connect the project to care, creativity, and practical digital choices. Pilots should measure reach, cost, and trust in the selected outlets rather than assume every family media channel is inexpensive or trusted.

Failure mode: **tech abstraction**, where families cannot see practical relevance to everyday Internet life.

5.14 Municipal and civic engagement

Cities, towns, and civic institutions can use the Meta-Layer to transform public engagement from complaint intake into participatory co-creation.

Municipal engagement MAY include:

- civic overlays above public service pages
- participatory planning annotations
- public meeting context layers
- local issue maps
- real-time dialogue tools
- citizen proposal pathways
- community event overlays
- participatory design dashboards

Failure mode: **complaint-system trap**, where engagement channels collect grievances without resolution, collaboration, or visible adaptation.

5.15 Cultural production and remix

The Meta-Layer needs cultural artifacts, not only documentation.

Communities SHOULD be encouraged to create:

- memes
- shorts
- songs
- posters
- zines
- games
- rituals
- visual metaphors
- local stories
- public demos
- speculative futures
- civic myths

Cultural production helps the Meta-Layer become imaginable.

Failure mode: **documentation-only adoption**, where only experts can understand or care about the system.

5.16 Events, rituals, and gathering grounds

Community engagement grows through repeated gatherings and shared practices.

DP19 MAY support:

- demo days
- build nights
- annotation jams
- bridge walks
- civic salons
- local chapter meetings
- online town halls
- contributor onboarding waves
- youth showcases
- press briefings
- public retrospectives
- seasonal campaigns

Events should create continuity and contribution pathways, not one-off spectacle.

Failure mode: **event evaporation**, where gatherings generate excitement but no durable roles, artifacts, or next steps.

6. Engagement Metrics and Community Health

DP19 must distinguish meaningful engagement from attention metrics.

6.1 Healthy engagement indicators

Healthy engagement MAY include:

- repeat participation

- successful onboarding
- contribution diversity
- local chapter formation
- multilingual participation
- participant retention
- quality of public explanations
- number of community-led events
- feedback routed into governance
- cross-community collaboration
- stewardship roles filled
- creator and educator adoption
- civic use cases launched

6.2 Risk indicators

Risk indicators MAY include:

- high reach with low retention
- bounty-driven spam
- audience confusion
- misleading public claims
- concentration of narrative power
- ambassador burnout
- platform dependency
- hostile media framing
- exclusion of non-technical audiences
- social posts disconnected from participation pathways

6.3 Engagement receipts

Major campaigns SHOULD generate engagement receipts.

An engagement receipt MAY include:

- campaign purpose
- intended audience
- claims made

- materials used
- responsible stewards
- incentives offered
- participation outcomes
- feedback received
- lessons learned
- governance follow-up

Failure mode: **metrics without memory**, where campaigns are repeated without learning.

7. Governance of Public Narrative

Narrative power must be governed because public language shapes legitimacy, funding, adoption, and expectation.

Communities SHOULD define:

- who can speak officially
- what counts as official messaging
- how community materials are labeled
- how claims are reviewed
- how mistakes are corrected
- how media inquiries are handled
- how partnerships are approved
- how ambassador conduct is governed
- how public feedback affects strategy

Public narrative governance should balance coherence with pluralism.

Failure mode: **message authoritarianism**, where one voice controls the story and suppresses community creativity.

Opposite failure mode: **narrative fragmentation**, where conflicting claims make the project incoherent.

8. Inclusion and Localization

DP19 requires global and local resonance.

Amplification and engagement systems SHOULD support:

- multilingual materials
- regional leadership
- culturally specific metaphors
- youth participation
- gender inclusion
- accessibility
- low-bandwidth participation
- non-Western civic contexts
- local governance traditions
- community translation rather than literal language substitution

Localization is not only translation. It is meaning-making in context.

Failure mode: **global English default**, where the Meta-Layer claims universality while speaking to a narrow audience.

9. Relationship to Existing Platforms

The Meta-Layer should use existing platforms without becoming dependent on them.

DP19-aligned strategy SHOULD include:

- platform-specific sharing practices
- decentralized community channels
- owned knowledge bases
- interoperable contact lists where consented
- portable contributor records
- open archives of public materials
- RSS, email, and web-native distribution
- bridges between social platforms and Meta-Layer spaces

Failure mode: **rented movement**, where the public presence of the Meta-Layer can be throttled, erased, or distorted by one platform's policies.

10. Governance, Accountability, and Agency Surfaces

Public presence is a governed function, not a communications afterthought. Someone always decides which claims are made, which partnerships are accepted, which voices are amplified, and which campaigns receive resources. DP19 requires those decisions to have named surfaces and reachable levers.

10.1 Accountability surfaces

- **A public claims register.** Statements about capability, adoption, security, and readiness are recorded with maturity labels and dates, so that later drift is visible (DP14, DP16).
- **A correction log.** When a public claim proves wrong, the correction is published in the same channels at comparable prominence, with the original preserved rather than quietly edited.
- **A partnership record.** Partners, sponsors, and trusted messengers are disclosed along with what they receive and what they were asked to say.
- **An ambassador roster and conduct record.** People who speak with delegated legitimacy are identifiable, and their mandate has scope and expiry.
- **A resource ledger for engagement spend.** Bounties, campaigns, and promotional budgets are visible enough for communities to see whose participation is being purchased (DP17).
- **A named steward for public narrative.** Responsibility for coherence sits with an accountable role rather than with whoever posted last.

10.2 Participant agency surfaces

Engagement without agency is audience management. DP19 expects participants to hold controls over their own presence and their community's story:

- opt in and out of promotional visibility, including having their contributions featured

- withdraw consent for use of their name, likeness, quotes, or community materials, with propagation to derived assets
- publish community materials under a community label that is not subordinate to official messaging
- contest a public claim made about their community and receive a dated response
- propose, amend, or veto campaigns that target their zone, school, family context, or municipality
- exit a campaign, program, or ambassador role without losing standing elsewhere (DP18)

Example: A local chapter objects to a national campaign that describes their neighborhood as "underserved." They file a narrative challenge, the phrasing is revised with attribution to the objection, and the correction is logged.

What this feels like: The story about you is something you can help write and something you can dispute.

Without this: Communities become material for someone else's growth narrative.

Failure mode: **spokesperson drift**, where informal voices accumulate representational authority that no community actually delegated.

11. Incentives and Power Analysis

Amplification allocates attention, and attention is a form of power. DP19 assumes that anyone funding, measuring, or distributing engagement will shape what participation becomes.

11.1 Where power concentrates

- **Distribution owners.** Platforms, newsletters, and search surfaces decide reach. Dependence on them converts community presence into a leased asset.
- **Budget holders.** Whoever funds bounties and campaigns sets the effective definition of valuable participation (DP9, DP17).

- **Metric definers.** If growth is measured in signups, the system will produce signups rather than stewards.
- **Trusted messengers.** Influencers and institutional partners can transfer legitimacy quickly, and can withdraw it just as fast.
- **Language and cultural defaults.** Whoever writes the canonical framing determines which communities must translate themselves to be heard (DP23).

11.2 Predictable incentive distortions

- **Extraction disguised as onboarding:** funnels optimized for conversion rather than comprehension, producing participants who cannot yet govern anything.
- **Bounty farming:** rewards attract volume over care, and low-quality contributions crowd out mentorship capacity.
- **Hype debt:** early overclaiming creates expectations that later delivery cannot meet, and the correction cost lands on community trust.
- **Ambassador dependency:** programs that concentrate visibility in a few people make the project fragile to their departure or misconduct.
- **Local labor, central credit:** chapters do the organizing while attribution and fundraising accrue upward.
- **Engagement ratchets:** gamified mechanics that are easy to add and politically hard to remove, even after they stop serving the community.

11.3 Counter-incentives DP19 expects

- reward learning and stewardship outcomes rather than raw activity, with reputation credited under DP18
- fund maintenance of community media commons, not only launch campaigns
- cap the share of promotional reach that any single messenger or platform may carry
- attribute campaign results to the local groups that produced them, including in funding narratives
- publish sunset dates for gamified mechanics and require review to renew them

- pair every growth target with a retention, comprehension, or governance-participation target
- keep a portable, community-owned channel of record so that presence survives platform decisions (DP7, DP20)

Example: A bounty program shifts from paying per referral to paying for completed peer-mentoring cycles, and publishes both cost per participant and six-month retention.

Failure mode: **growth capture**, where the incentive to expand overrides the community's ability to absorb, govern, and retain new participants.

12. Community Signals Informing DP19

DP19 reflects concerns raised repeatedly by organizers, educators, families, municipal staff, and contributors who have been on both sides of platform growth campaigns.

Recurring signals include:

- "Nobody can explain what this is." Participants ask for one honest sentence before they ask for features.
- "It sounds like every other project." Undifferentiated language reads as either hype or vapor.
- "We were used for a case study." Communities contribute and then find themselves as marketing evidence without consent.
- "The rewards attracted the wrong people." Organizers describe bounty programs that increased volume and decreased trust.
- "Our chapter did the work; the announcement didn't mention us." Attribution asymmetry is a common source of burnout.
- "We can't reach our own members." Platform algorithm changes cut communities off from lists they built.
- "Parents need something different than developers." Audience-specific explanation is requested consistently, not as an accessibility nicety.
- "Youth want to make things, not consume campaigns." Creative and remix participation outperforms broadcast messaging.
- "Local government needs procurement-legible language." Civic adoption stalls when materials do not fit institutional processes.

- "Tell us what is real and what is planned." Participants tolerate incompleteness far better than they tolerate ambiguity about maturity.

These signals point toward presence that is legible, plural, locally credited, and honest about its stage of development. DP19 encodes them as structural conditions rather than as communications style preferences.

Community signals inform DP19; they do not by themselves establish consensus. They are recorded so that future revisions can be tested against the concerns that motivated the property.

13. Evaluation Criteria

A DP19-aligned implementation should be evaluated against the following questions.

13.1 Narrative clarity

- Can different audiences explain the Meta-Layer in their own words?
- Are Metaweb, Overweb, and related terms clearly distinguished?
- Does public language avoid confusion with extractive or dystopian digital narratives?

13.2 Truthfulness

- Do campaigns distinguish current capabilities from aspirations?
- Are roadmap dependencies and limitations visible?
- Are public claims corrected when they become inaccurate?

13.3 Community agency

- Can participants shape amplification strategy?
- Are ambassadors accountable to communities, not only central teams?
- Are community-generated materials recognized and supported?

13.4 Incentive alignment

- Do bounties reward quality, care, and learning rather than volume?
- Are rewards transparent and fair?
- Are spam, duplicate content, and misleading claims discouraged?

13.5 Inclusion

- Are materials accessible across language, region, age, ability, and technical literacy?
- Do youth, families, educators, local communities, and non-technical participants have real entry points?
- Are trusted messengers supported without being exploited?

13.6 Retention and continuity

- Do campaigns lead to durable participation pathways?
- Are events connected to roles, artifacts, or governance channels?
- Can participants move from awareness to contribution?

13.7 Governance and accountability

- Is official messaging governed?
- Are errors corrected transparently?
- Are media and partnership strategies accountable to the ecosystem?

14. Implementation Patterns

14.1 One-sentence public anchor

Every campaign should be able to use a clear public anchor.

Example:

The Meta-Layer lets people and communities add trust, context, presence, and governance above the web they already use.

14.2 Audience-specific explainers

Develop tailored explainers for parents, students, developers, journalists, municipalities, creators, and AI governance audiences.

14.3 Ambassador kits

Provide ambassadors with slide decks, FAQs, demo scripts, visual assets, local event templates, ethical messaging guidelines, and feedback forms.

14.4 Community media commons

Create a shared repository of approved, remixable, versioned public materials.

14.5 Press kits with maturity labels

Press materials should label what is live, experimental, conceptual, and proposed.

14.6 Learning-first bounties

Reward explainers, translations, workshops, demos, and case studies that help people understand and participate.

14.7 Local chapter campaigns

Support city, campus, school, library, and community chapter pilots with lightweight governance and reporting.

14.8 Youth-led creative campaigns

Invite youth to create shorts, memes, zines, and demos that translate Meta-Layer values into native cultural formats.

14.9 Cross-posting with context

Sharing tools should preserve provenance, links, and consent while adapting format to each platform.

14.10 Engagement retrospectives

After major campaigns, publish what worked, what failed, what was learned, and what changes next.

15. Relationship to Other Desirable Properties

15.1 DP2 – Participant Agency and Empowerment

DP19 must amplify agency rather than manipulate attention. Participants should understand how to join, shape, and leave engagement pathways.

15.2 DP3 – Adaptive Governance Supporting an Exponentially Growing Community

Public feedback from campaigns must route into governance. Narrative strategy should evolve based on community learning.

15.3 DP5 – Decentralized Namespace

Names, symbols, handles, badges, and community identities should be portable and resistant to capture.

15.4 DP7 – Simplicity and Interoperability

Engagement should move across platforms, communities, and tools without losing context or continuity.

15.5 DP8 – Collaborative Environment and Meta-Communities

Communities should be able to define their own engagement practices, ambassador roles, and public presence.

15.6 DP9 – Developer and Community Incentives

Bounties, badges, and rewards for amplification must align with contribution quality, transparency, and anti-gaming safeguards.

15.7 DP10 – Education

DP19 depends on learning pathways that convert curiosity into practical literacy.

15.8 DP14 – Trust and Transparency

Public narrative must be honest, auditable, and corrigible.

15.9 DP16 – Roadmap and Milestones

Amplification claims must align with real roadmap status and maturity.

15.10 DP18 – Feedback Loops and Reputation

Community engagement must feed back into learning systems, and public contribution should be recognized without devolving into popularity scoring.

15.11 DP20 – Community Ownership

Communities should own and steward their narratives, symbols, materials, and engagement histories where appropriate.

16. Non-Goals and Explicit Boundaries

DP19 is the property most easily misused, because growth language can be attached to almost anything. The following boundaries are explicit so that DP19 cannot be cited to justify the dynamics it exists to constrain.

DP19 is not a marketing mandate. It does not authorize persuasion techniques, urgency manufacturing, engagement optimization, or attention capture. Amplification is legitimate only where it increases understanding and agency.

DP19 does not centralize the story. There is no single authorized narrative, brand voice, or approved vocabulary that communities must adopt. Coherence is a shared asset; uniformity is a governance failure.

DP19 does not make reach a success metric. Impressions, followers, and signups are not evidence of DP19 alignment. Retention, comprehension, stewardship, and governance participation are the relevant measures.

DP19 does not license claims beyond maturity. Public presence must track actual system state under DP16 and DP14. Aspirational description presented as capability is an alignment failure, not enthusiastic communication.

DP19 does not authorize use of community labor or likeness by default. Featuring people, quoting them, or turning their work into promotional material requires consent that can be withdrawn (DP2, DP4).

DP19 does not convert participation into consent to solicitation. Joining a community is not opting into fundraising, recruitment, or partner messaging.

DP19 does not endorse gamification as such. Points, streaks, and leaderboards are permitted only with published purpose, guardrails, and sunset review. Mechanics that manufacture compulsion are out of scope.

DP19 does not replace governance with community management. Engagement programs cannot substitute for decision rights under DP3, DP8, and DP20. An active community that cannot decide anything is an audience.

DP19 does not require presence on any particular platform. Using commercial platforms is permitted; depending on them for continuity of identity, memory, or reach is not.

DP19 does not target children as a growth channel. Youth and family engagement is educational and creative in purpose, governed with heightened protections, and never optimized for acquisition.

DP19 does not measure people to manage them. Engagement analytics operate under DP4 minimization and DP18 proportionality; engagement receipts describe what the system did, not surveillance of what participants felt.

17. Minimum DP19 Alignment

A system, campaign, or community program claiming DP19 alignment should be able to demonstrate the following. These statements are non-normative and are offered as an assessment aid rather than a conformance test.

17.1 Honest public anchor

There is a short, plain-language description of what the Meta-Layer is, what it is not yet, and who it is for, published where newcomers actually arrive, and maintained as the system changes.

17.2 Maturity labeling

Public claims carry maturity labels such as concept, draft, prototype, pilot, or deployed, with dates. Press materials and demos distinguish what exists from what is planned.

17.3 Correction practice

There is a published route for challenging a public claim, an owner for corrections, and a visible log of corrections issued at comparable prominence

to the original claim.

17.4 Community-labeled materials

Community-produced explanations, translations, and remixes exist, are labeled as community work rather than official messaging, and are reachable from official surfaces.

17.5 Consent-bound promotion

Featuring participants, quoting them, or using community artifacts requires recorded consent with scope, duration, and a withdrawal path that propagates to derived materials.

17.6 Non-extractive incentives

Bounties, badges, and rewards are tied to learning, contribution quality, or stewardship outcomes, with published criteria, anti-spam safeguards, and reputation effects governed under DP18.

17.7 Delegated voice with limits

Ambassador and steward roles have written mandates, scope, term limits, conduct expectations, and a public roster, so that delegated legitimacy is traceable.

17.8 Platform independence floor

At least one channel of record for reaching participants is owned or federated rather than platform-dependent, and public materials are archived in an exportable form.

17.9 Localization beyond translation

Materials exist in more than one language, produced or reviewed by speakers within the relevant community, with local metaphors and civic contexts rather than literal substitution (DP23).

17.10 Engagement receipts

Campaigns publish what was done, what it cost, what happened, and what was learned, including outcomes that fell short, so that engagement remains accountable to the community it addressed.

18. Open Questions and Future Work

1. What minimum standards should define official versus community-generated messaging?
2. What public terminology should be standardized across the Meta-Layer ecosystem?
3. How should brand assets be governed in a decentralized ecosystem?
4. What claims require evidence, maturity labels, or review before publication?
5. How should ambassador programs be structured across regions and languages?
6. What reward schemas prevent bounty spam while supporting public education?
7. How should campaign materials preserve provenance and version history?
8. What engagement metrics best reflect civic participation rather than attention capture?
9. How should youth-centered engagement be governed ethically?
10. What rules should govern cross-posting from Meta-Layer spaces to social platforms?
11. How should public relations strategy remain accountable to community governance?
12. What local chapter model best supports global inclusion without central bottlenecks?
13. How should public mistakes, overclaims, or misleading media coverage be corrected?
14. What interoperability standards are needed for badges, ambassador credentials, and engagement receipts?

19. Path Toward ML-RFC

DP19 is currently an ML-Draft and serves as exploratory scaffolding for how public presence, community engagement, branding, amplification, and adoption pathways can operate as part of Meta-Layer infrastructure.

Advancement toward ML-RFC status SHOULD require:

- convergence on terminology for public-facing Meta-Layer concepts
- a governed distinction between official, community, experimental, and partner messaging
- a minimal schema for engagement receipts
- a minimal schema for ambassador or community steward credentials
- tested bounty and badge systems that reward quality over volume
- evidence from community-led campaigns across multiple audiences or regions
- accessible onboarding materials for non-technical participants
- documented feedback loops from public engagement into governance
- demonstrated safeguards against hype, spam, and misleading claims
- at least one localization or multilingual engagement pilot

ML-RFC promotion SHOULD be contingent on:

- rough consensus among participating communities
- demonstrated use of DP19 mechanisms in real outreach or adoption settings
- clear alignment with DP9 incentive safeguards and DP18 reputation logic
- transparent handling of public claims, corrections, and campaign outcomes
- evidence that engagement pathways increase agency and stewardship, not only reach

Early ML-RFC candidates may focus on:

- public terminology and maturity-label standards
- engagement receipt formats
- ambassador credential standards
- ethical bounty and badge guidelines
- campaign transparency requirements
- community media commons governance

DP19 is likely to evolve through multiple partial RFCs rather than one monolithic specification because brand, narrative, engagement, and public participation each have distinct governance and implementation needs.

20. Closing Orientation

DP19 gives the Meta-Layer a public face and a living community presence.

It ensures that the Metaweb is not only built, but recognized; not only explained, but shared; not only promoted, but inhabited.

A DP19-aligned meta-layer grows through resonance rather than manipulation. It invites people into agency, context, trust, and co-creation without reducing them to metrics or marketing channels.

Brand becomes symbolic orientation. Engagement becomes participatory belonging. Amplification becomes civic invitation. Presence becomes shared infrastructure.

This is how the Meta-Layer becomes something people can name, join, steward, and carry into the world.

Continue the conversation on DP19 online →

DESIRABLE PROPERTY 20

DP20

Community Ownership

The Conditions for Ownership That Can Be Exercised



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp20

1. Purpose of This Draft

This draft articulates Desirable Property 20 (DP20) as the condition under which communities meaningfully own the digital environments, surplus, and governance artifacts they co-create, not as marketing language about "community," but as enforceable rights and responsibilities: stake, voice, upside, and continuity that persist across tools and seasons.

DP20 completes a loop with DP18 (feedback and reputation), DP19 (presence and engagement), and DP12 (community governance of AI): participation must be able to mature into ownership, otherwise "community" remains a rented audience for platforms.

DP20 connects to DP4 (data commons and collective data rights), DP6 (fair distribution of commercial surplus), DP9 (incentives that return value to contributors), DP3 (adaptive governance at scale), DP17 (financial sustainability), and DP1–DP2 (accountability and agency for collective actors).

If DP20 is weak, predictable failures follow: extractive growth on volunteer labor, token claims without decision rights, capture of governance by insiders, inability to fork when values diverge, and surplus flowing outward while risks stay local.

DP20 does not mandate a single cooperative legal form or chain. It defines minimum legitimacy conditions for what "ownership" may claim in the Meta-Layer.

2. Problem Statement

In today's web, "community" is often a label applied to users whose labor, attention, and data build value they cannot capture, audit, or exit with. Governance surfaces are optional; economic upside is asymmetric; and forking social reality is impractical.

In practice, this produces recurring failures:

- moderators and maintainers burn out while platform valuation rises
- creators depend on distribution they do not control
- tokens grant speculation liquidity without durable rights
- data generated collectively is sold without collective consent (DP4)

- communities cannot carry norms, memory, or economic terms across apps (DP7)

These failures are structural: ownership is confused with usage; participation is confused with consent to extraction.

DP20 reframes ownership as operational: rights you can see, exercise, and defend, including pathways to fork, exit, and recapture surplus when legitimacy breaks.

3. Threats and Failure Modes

3.1 Ownership theater

Marketing claims "community-owned" while decisions and keys remain centralized.

Example: A DAO where a multisig admin can override any vote without published criteria.

Why this matters: DP20 requires honest mapping between claims and controls.

3.2 Token liquidity without duties or rights

Tokens trade while governance is inactive or non-binding.

Example: Voter apathy plus whale dominance makes governance a periodic headline, not a constraint on operators.

Why this matters: Ownership instruments must bind to decisions and surplus, or be renamed.

3.3 Commons maintenance without reciprocity

Value is extracted from a space while moderation, safety, and infrastructure remain underfunded.

Example: A viral forum hosts commerce ads but does not fund anti-abuse capacity.

Why this matters: Connects DP6 reciprocity and DP17 sustainability.

3.4 Capture by concentrated stakeholders

Early contributors or large holders lock rules that favor continuation of privilege.

Example: Fee switches require supermajorities that insiders can block indefinitely.

Why this matters: DP3 adaptive governance and DP12 dialectic must resist ossified capture.

3.5 Forklessness

When values diverge, participants cannot exit with continuity of identity, memory, or economic position.

Example: A community schism loses all history because export is blocked and namespaces are not portable (DP5, DP7).

Why this matters: Forking is a safety valve for legitimacy.

3.6 Collective data sale without governance

Aggregated or derived data monetizes communities without authorization pathways.

Why this matters: DP4 sovereignty requires collective decision surfaces, not only individual toggles.

3.7 Illegible membership and shadow elites

Informal power structures make decisions without published accountability.

Example: Moderators coordinate bans off-platform without records or appeals.

Why this matters: DP1 contestability and DP12 governance memory must extend to collective actors.

3.8 Philanthropic dependency

Communities rely on unsustainable grants with no path to owned revenue or shared surplus.

Example: A civic space runs on a yearly sponsor who can pull funding overnight.

Why this matters: DP17 sustainability and DP6 commerce reciprocity must appear in ownership design.

3.9 Legal hostility to collective action

Cooperative forms face banking, tax, and liability friction.

Why this matters: DP20 acknowledges law as a design constraint while still demanding honest digital rights bundles.

4. Core Principle

Community ownership in the Meta-Layer means participants and communities hold defensible rights to govern, benefit from, and responsibly steward the environments and surplus they co-create, including credible paths to fork, exit, and recapture value, with transparency, accountability, and continuity of memory.

Ownership is not a vibe. It is power with responsibility, visible to members and bounded by human-rights baselines and law.

Example: A community charter defines decision rights, surplus split, audit rights, and fork procedures; tools entering the zone must acknowledge the charter's machine-readable profile.

What this feels like: You are not the community in a slogan. You are a member with levers.

Without this: Participation is harvested; dignity is not.

5. Primary Mechanisms and Structural Conditions

5.1 Ownership Layer: Execution, Proof, and Exit

Community ownership requires more than charters and intent. It requires the ability to express, verify, exercise, and, when necessary, exit ownership as part of an operational system.

In many systems, "ownership" fails because it is not bound to decision rights, not provable in practice, or not portable across tools. Rights exist on paper while control remains elsewhere.

DP20 therefore requires a shared ownership layer composed of primitives that make ownership executable rather than symbolic.

5.1.1 Ownership objects

Ownership must be represented as structured, machine-readable objects that bind rights to actors and contexts.

An ownership object includes:

- scope: what environment, assets, or processes are owned
- rights: governance, economic, and data rights
- duties: stewardship and maintenance obligations
- transfer conditions: how ownership can change hands

This binds ownership directly to governance (DP3, DP12).

5.1.2 Ownership receipts

All ownership-relevant actions produce verifiable records.

An ownership receipt includes:

- who holds or exercised a right
- what action was taken (vote, allocation, fork, transfer)
- when it occurred
- what assets or policies were affected

These receipts make ownership auditable and connect to DP15 (provenance).

5.1.3 Ownership–governance binding

Ownership must map to decision rights in a visible way.

Participants must be able to see:

- who can decide what
- how ownership affects voting or delegation
- where overrides or special powers exist

Governance that cannot be owned is advisory. Ownership that cannot govern is symbolic.

5.1.4 Ownership memory

Ownership history must persist over time.

This includes:

- prior allocations and changes
- disputes and resolutions
- forks and mergers

Without memory, ownership becomes contestable without evidence.

These primitives do not replace the mechanisms below. They make them enforceable, portable, and auditable across contexts.

5.2 Charter and rights bundle

A written and machine-readable articulation of membership, decision rights, economic rights, data rights, fork rights, and sunset conditions.

This MUST include:

- explicit mapping between roles and powers
- conditions under which rights can change
- linkage to governance execution (DP3, DP12)

Verification:

- participants can inspect current and prior versions
- tools enforce charter constraints at interaction level

Failure modes:

- **charter drift** (rules change without visibility)
- **unenforced rights** (charter exists but does not constrain behavior)

5.3 Transparent surplus accounting

Revenue, fees, grants, and in-kind support are visible to members at useful granularity and traceable to ownership claims; without this, ownership cannot capture upside in a defensible way (DP6, DP17).

This MUST include:

- inflow sources and categories
- allocation breakdowns tied to functions
- linkage to ownership claims, rights, and contributions

Verification:

- members can trace value from source → allocation → outcome

Failure modes:

- **surplus opacity** (value cannot be traced)
- **extraction masking** (flows hidden behind aggregation)

5.4 Participatory budgeting and parameter control

Communities set budgets for safety, education (DP10), and incentives (DP9) through governed processes (DP12).

This MUST include:

- binding decision pathways
- budget execution tracking
- feedback loops into future allocations

Failure modes:

- **advisory budgeting** (decisions ignored)
- **capture of allocation power**

5.5 Credible exit and fork

Procedures to export collective data and governance artifacts (DP4, DP7), migrate identifiers where honest (DP5), and continue governance under divergent values without coercion. Forking is not a failure; it is a safety valve for legitimacy.

This MUST include:

- executable export mechanisms
- continuity of identity, data, governance artifacts, and membership and contribution records

Failure modes:

- **exit suppression**
- **fork degradation** (loss of continuity)

5.6 Stewardship obligations

Ownership includes maintenance duties: moderation, security, inclusion, and accessibility commitments with accountability (DP1, DP15).

This MUST include:

- defined duties
- enforcement mechanisms
- reporting on fulfillment

Failure modes:

- **duty abandonment**
- **free-rider extraction**

5.7 Anti-capture controls

Rotation, conflict-of-interest rules, minority protections, and emergency pause pathways.

This MUST include:

- detection of concentration
- intervention pathways

Failure modes:

- **ownership capture**
- **governance lock-in**

5.8 Collective data governance

Shared corpuses require collective consent mechanisms and audit of downstream use (DP4).

This MUST include:

- consent pathways
- audit logs of use

Failure modes:

- **data extraction without consent**
- **loss of collective control**

5.9 Interoperable ownership artifacts

Membership and contribution are represented as portable, verifiable artifacts that interoperate across tools (DP7), so recognition, identity, and stake carry across contexts.

This MUST include:

- portable formats
- degradation signaling across systems

Failure modes:

- **lock-in via incompatibility**

5.10 Succession and dissolution

Charters define succession for stewards and fair dissolution procedures.

This MUST include:

- transfer conditions
- fair distribution of residual value

Failure modes:

- **power vacuum**
- **unfair dissolution**

5.11 Intersectional inclusion in ownership

Governance design actively counters exclusion patterns through facilitation norms, translation, and accessible participation.

This MUST include:

- measurable inclusion practices
- correction mechanisms

Failure modes:

- **systemic exclusion**
- **ownership concentration through bias**

6. Governance, Accountability, and Agency Surfaces

DP20 requires that ownership is not only visible but **defensible and enforceable.**

Participants MUST be able to:

- verify ownership rights and mappings
- challenge illegitimate control or misallocation
- trigger governance review or dispute processes
- exercise exit and fork pathways without coercion

Communities MUST be able to:

- revoke or reassign ownership under defined conditions
- enforce stewardship obligations
- audit ownership actions and outcomes

Enforcement pathways include:

- rollback or invalidation of illegitimate actions
- governance-triggered reallocation of rights or surplus
- fork as a last-resort enforcement mechanism

Failure modes:

- **non-actionable ownership**
- **illegitimate control persistence**
- **ownership without recourse**

7. Incentives and Power Analysis

Ownership fails when incentives diverge from rights.

Key dynamics:

- **extraction loops:** value flows outward despite community contribution
- **token manipulation:** governance power decouples from responsibility

- **dependency capture:** infrastructure or distribution creates hidden control
- **soft capture:** influence accumulates without formal ownership

Adversarial patterns:

- governance capture through capital or coordination
- speculative ownership without stewardship
- platform rent extraction

Alignment requirements:

- ownership rights must correlate with value capture
- incentives must reward stewardship, not exploitation
- systems must detect divergence between formal ownership and real control

Operational independence is part of this test: dependence on a government's continuing funding, permission, or goodwill as the sole support for a community's infrastructure is a sign that formal ownership and real control have diverged. Public partnerships can contribute while communities retain practical control, continuity, and exit; independence is not exemption from applicable law.

Failure modes:

- **symbolic ownership**
- **incentive inversion**
- **ownership erosion over time**

8. Community Signals Informing DP20

Across platforms and communities, recurring signals point to a shared breakdown between participation and ownership:

- maintainers and moderators demand compensation and decision power proportional to their labor
- skepticism toward "community-owned" claims following repeated governance failures and token collapses
- desire for portable community memory, identity, and governance tools

- frustration with value extraction from spaces built through collective effort
- growing interest in local, cooperative, and community-controlled digital environments

These signals are not isolated. They indicate structural gaps in how ownership is defined, enforced, and sustained.

DP20 treats these signals as design requirements, not feedback to be addressed after failure.

9. Foresight and Failure Design

DP20 assumes ownership systems will be tested by capture attempts, economic pressure, and governance failure.

Common failure paths include:

- concentration of ownership among early or privileged actors
- erosion of rights through informal overrides or hidden control layers
- suppression of exit or fork pathways to preserve centralized power
- divergence between formal ownership and actual value capture

DP20 requires designing safeguards in advance, including:

- transparent allocation and reallocation mechanisms
- enforceable limits on concentrated control
- clear and executable fork and exit procedures
- public postmortems linking failures to structural changes

Failure is expected. Illegible failure is not.

Foresight in DP20 also means anticipating success. Ownership structures fail differently when they grow: a small cooperative can govern by conversation, while a large one cannot. DP20 therefore expects communities to plan the transitions in advance rather than discover them under stress.

Scale transitions that require pre-designed responses:

- **From informal to formal.** Founding trust does not survive membership growth. Charters, rosters, and decision records must exist before

informal authority becomes contested.

- **From volunteer to funded.** Introducing money into a volunteer commons reorders status. Compensation criteria and conflict-of-interest rules should be set before the first payment, not after the first grievance.
- **From single zone to federation.** Ownership artifacts that assumed one community must survive contact with many (DP7, DP8). Cross-zone recognition rules should be written while stakes are still low.
- **From founders to successors.** Every founding cohort eventually leaves. Succession, key rotation, and archive custody are ownership functions, not administrative details.
- **From growth to decline.** Communities shrink. Dissolution with dignity, including asset disposition and memory preservation, is part of ownership design (DP22).

Pre-mortem practice DP20 expects:

- name the three most likely capture vectors for this specific community, and the control that would detect each
- state in advance what evidence would indicate that ownership has become theater
- define the trigger conditions under which a fork is considered legitimate rather than hostile
- rehearse an exit at least once, so that portability claims are tested rather than asserted
- record what the community would refuse even if it were profitable, and where that refusal is written down

Failure design is not pessimism. It is the difference between a community that can survive its own mistakes and one that discovers its structure only when it breaks.

10. Knowledge Graph Stewardship

Communities do not only own spaces, funds, and governance artifacts. They also produce the semantic substrate that makes shared understanding possible: vocabularies, taxonomies, tags, annotations, curated links, correct-

ions, glossaries, and the relationships that connect them. This substrate is a knowledge graph, whether or not it is called one.

Knowledge graphs are unusually vulnerable to quiet expropriation. They are built incrementally by many people, they carry no obvious authorship, and their value appears only in aggregate. A platform can absorb decades of community curation without any single act that looks like a taking.

DP20 therefore treats collective knowledge structures as ownable, governable, and forkable assets.

10.1 What is being stewarded

- **Vocabularies and ontologies:** the terms a community uses, their definitions, and their relationships
- **Classification schemes:** categories, tags, and topic hierarchies, including their edge cases and exclusions
- **Annotations and overlays:** notes, corrections, context, and interpretive layers attached to shared artifacts
- **Curation and linkage:** which sources are considered relevant, credible, or connected, and by whom
- **Provenance records:** who asserted what, when, and on what basis (DP15)
- **Glossary and translation layers:** cross-language term mappings and contested terminology (DP23)
- **Governance memory:** the record of how definitions changed and why (DP22)

10.2 Stewardship obligations

Ownership of a knowledge graph is custodial rather than proprietary. A community holding one accepts obligations:

- maintain provenance so that assertions can be traced to their source and revisited
- keep definitions versioned, so that meaning drift is visible rather than silent
- credit curation labor, including the unglamorous work of merging duplicates and fixing broken relationships

- resolve conflicting assertions through published process rather than by last-writer-wins
- preserve minority and contested interpretations instead of collapsing them into a single canonical entry
- publish the license and reuse terms under which the graph may be consumed
- retain the ability to correct or attenuate entries about people (DP4, DP18)

10.3 Rights the community must hold

- **Export rights.** The graph must be extractable in an open, documented form, including relationships and provenance, not only flattened content.
- **Fork rights.** A community that diverges must be able to take its semantic layer with it and continue evolving it independently (DP7).
- **Consent rights over derived use.** Bulk consumption for model training, resale, or downstream inference is a collective decision, not a default permission (DP4, DP11–DP13).
- **Attribution rights.** Systems that build on a community graph must disclose that dependency.
- **Refusal rights.** Communities may decline to encode knowledge that is sacred, protected, or dangerous to expose, and that refusal must be recordable rather than treated as a data gap.

10.4 Failure modes

- **Curation extraction:** a platform monetizes or trains on community-built structure while contributing nothing to its maintenance
- **Ontology capture:** a well-resourced actor gains effective control of definitions, shaping what the community can express
- **Schema lock-in:** the graph is technically exportable but semantically meaningless outside its original tooling
- **Silent redefinition:** terms change meaning without versioning, invalidating past assertions retroactively
- **Curation debt:** maintenance is unfunded, quality degrades, and the graph becomes unreliable while still being trusted

- **Consent bypass through derivation:** embeddings, summaries, or inferred relationships are treated as new works exempt from community terms

Example: A community maintains a topic taxonomy used by several tools. Its charter states that the taxonomy is exported monthly in an open format, that model training on it requires a governance vote, that three stewards are compensated for maintenance, and that a fork inherits full provenance history.

Why this matters: If communities own their spaces but not their semantics, someone else still decides how their knowledge can be described, found, and used.

11. Relationship to Other Desirable Properties

DP20 anchors ownership within the broader Meta-Layer system.

- DP3 defines how governance evolves under ownership
- DP4 constrains how collectively generated data can be used
- DP5–DP7 enable portability of identity, assets, and ownership artifacts
- DP6 and DP9 determine how value flows and incentives interact with ownership
- DP12 ensures ownership-linked governance can execute in practice
- DP15 ensures ownership claims and actions are provable
- DP17 ensures ownership structures can sustain themselves financially
- DP18–DP19 provide the participation and reputation inputs that mature into ownership

DP20 binds these properties into a coherent model of collective power.

DP20 also depends on the memory and language properties. DP22 preserves the governance lineage that makes ownership history legible over time, and DP23 ensures that ownership terms, charters, and votes remain accessible to members who do not share a dominant language. Ownership that cannot be read by its own members is ownership in name only.

12. Non-Goals and Explicit Boundaries

DP20 does not:

- require all systems to be collectively owned
- eliminate private enterprise or hybrid ownership models
- guarantee equal distribution of value or influence
- remove the need for legal structures and compliance

DP20 defines the conditions under which ownership claims are legitimate and enforceable. It does not prescribe a single model.

The following boundaries are stated explicitly, because "community ownership" is a phrase frequently used to describe its opposite.

DP20 does not mandate tokens. Tokenization is one instrument among many. Cooperatives, trusts, associations, stewardship councils, and non-transferable membership rights are equally admissible. A token that grants no decision rights is not ownership regardless of market depth.

DP20 does not require blockchains. Verifiable records may be maintained through several means. What matters is that ownership history is tamper-evident and independently checkable (DP15), not which ledger technology produces that property.

DP20 does not guarantee profit or surplus. Many commons will run at cost. DP20 requires that whatever surplus exists be visible and allocated under published rules, not that surplus be produced.

DP20 does not equate ownership with unanimity. Collective ownership includes the authority to decide over objection, bounded by minority protections and exit rights. Consensus is a method, not a requirement of legitimacy.

DP20 does not abolish operators. Communities may delegate operational authority to staff, vendors, or foundations. Delegation is legitimate when it is scoped, revocable, and recorded.

DP20 does not make forking costless. It requires that forking be possible with continuity of identity, memory, and economic position. It does not promise that forking will be painless or that both branches will thrive.

DP20 does not override rights of individuals. Collective ownership of a space does not confer collective ownership of its members' data, attention, or labor. Individual sovereignty under DP2 and DP4 constrains what a community may decide.

DP20 does not resolve legal form. Jurisdictional recognition of collective entities varies widely and will remain a constraint. DP20 requires honesty about the gap between digital rights bundles and legal enforceability rather than pretending the gap does not exist.

DP20 does not treat membership as property. Participation rights may be non-transferable by design. Preventing the sale of membership is compatible with, and often necessary for, durable community ownership.

13. Minimum DP20 Alignment

Minimum alignment defines the threshold where ownership is **real, enforceable, and not misleading**.

A DP20-aligned system MUST:

- bind ownership to executable governance and economic rights
- provide verifiable ownership records and history (DP15)
- ensure visibility into value flows and surplus allocation (DP17)
- enable credible exit and fork with continuity (DP4, DP7)
- maintain enforceable stewardship obligations

Failure modes to avoid:

- **ownership theater** (claims without control)
- **fork suppression** (no exit pathway)
- **surplus extraction without rights**
- **historical erasure of ownership changes**

Systems that omit execution, auditability, or exit MUST NOT be considered aligned.

The following statements describe what those conditions look like in practice. They are non-normative and are offered as an assessment aid rather than a conformance test.

13.1 Published rights bundle

There is a charter or equivalent document stating who the members are, what decisions they control, what share of surplus they hold, what audit rights exist, and how the charter itself may be amended.

13.2 Legible membership

Membership criteria, current membership, and the process for joining and leaving are visible to members. Informal power that operates outside the roster is identified rather than tolerated silently.

13.3 Binding decision surface

At least one class of consequential decision is actually decided by members, with a record showing the question, the participation, the outcome, and its execution (DP12).

13.4 Surplus and cost visibility

Members can see revenue, costs, reserves, and allocation for the current period, at a granularity sufficient to detect extraction (DP17).

13.5 Exercisable exit

Export of identity, memory, relationships, and ownership artifacts is documented, tested, and available without permission from an operator. Fork procedures are written before they are needed.

13.6 Anti-capture controls

There are published limits on concentrated control, including caps, quorum rules, rotation, or veto protections, and a record of when those controls were invoked.

13.7 Collective data authority

Uses of collectively generated data, including derived and aggregated forms, require a member-facing authorization pathway rather than a terms-of-service default (DP4).

13.8 Steward accountability

Delegated roles have scope, term, compensation, conflict-of-interest disclosure, and a removal procedure that members can initiate.

13.9 Continuity plan

Succession, key custody, archive stewardship, and dissolution terms exist in writing, so that the community can survive the departure of any individual.

14. Open Questions and Future Work

Key open questions include:

- how to design ownership models that function across jurisdictions
- how to balance tokenized and non-tokenized ownership structures
- how to maintain privacy while supporting interoperable ownership credentials (DP4)
- how to protect minority voices without blocking collective action
- how to measure contribution across visible and invisible labor (care, moderation, coordination)
- how to handle liability for collective decisions across legal systems
- how AI agents participate in ownership structures, if at all (DP11–DP13)

These questions reflect the boundary between social legitimacy and technical implementation.

Further questions raised by knowledge graph stewardship and scale transitions:

- what minimum schema should describe an ownership object so that it remains meaningful across zones and tools
- how should a fork inherit provenance, reputation, and knowledge graph history without importing the disputes that caused the split
- what licensing model best protects community-curated semantic layers from extraction while remaining genuinely open
- how should consent for model training on community knowledge be expressed, revoked, and verified downstream

- how should compensation for invisible maintenance labor be assessed without turning care work into a metric to be gamed (DP18)
- what governance form allows a community to refuse a lucrative offer without that refusal becoming a liability for its stewards
- how should dissolution allocate assets, archives, and namespaces when members disagree about legitimate succession (DP5, DP22)
- how can ownership rights remain intelligible to members across languages and literacy levels (DP23)

15. Path Toward ML-RFC

Advancing DP20 toward ML-RFC requires:

- standardizing ownership object and receipt formats
- developing reference implementations of community-owned systems with open accounting
- aligning ownership models with interoperability and identity standards
- testing fork and exit mechanisms in real communities
- integrating legal and cooperative expertise into design processes

Progress should be demonstrated through functioning systems, not only conceptual agreement.

16. Closing Orientation

DP20 is where the Meta-Layer turns participation into power.

It rejects the model where communities create value but do not control it.

When ownership is real, communities can govern, sustain, and evolve the systems they depend on.

When it is not, "community" remains a narrative layered over extraction.

Ownership is the difference between presence and permanence, between contribution and control.

DP20 is where the Meta-Layer stops confusing audience with citizenship.

Ownership is the difference between a group that is used and a group that endures, with memory, upside, and responsibility held together.

When DP20 is strong, people can build commons that last because the commons can belong to those who care for them.

Continue the conversation on DP20 online →

DESIRABLE PROPERTY 21

DP21

Multi-modal

Multi-Modal Interactions and Experiences – Meaning Before Presentation



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp21

1. Purpose of This Draft

This draft articulates Desirable Property 21 (DP21) as the condition under which the Meta-Layer can be experienced across multiple sensory modalities, devices, and interaction paradigms without fragmenting meaning, agency, trust, or governance.

DP21 defines how the Metaweb supports visual, auditory, textual, spatial, tactile, and emerging modalities so participants can engage in ways that are natural, accessible, and context-appropriate.

The Meta-Layer must not be confined to screens or text interfaces. It must function as a cross-modal civic interface that can appear in browsers, voice systems, AR/VR environments, ambient computing contexts, and future interaction substrates. *Ambient computing contexts* are smart, sensor-based environments where the interface is spread across the surroundings rather than held in one device. Examples include connected homes, vehicles, workplaces, and public spaces.

If DP21 is weak, predictable failures follow: the Meta-Layer becomes screen-bound; accessibility gaps widen; immersive systems become siloed; voice and AI interfaces lose context; meaning degrades across modalities; and participation becomes limited to technically privileged users.

DP21 connects directly to:

- DP2, participant agency and empowerment
- DP4, data sovereignty and privacy
- DP7, interoperability
- DP8, community-defined participation zones
- DP10, education and onboarding
- DP12 and DP13, AI governance and containment
- DP14, trust and transparency
- DP18, feedback loops and reputation
- DP19, amplifying presence and engagement
- DP22, civic memory and epistemic continuity

DP21 does not prescribe specific hardware, platforms, or interaction styles. It defines the minimum conditions under which multi-modal participation

remains coherent, accessible, and governed.

2. Problem Statement

The current web is primarily visual and text-based, with fragmented support for cross-modal continuity.

Meanwhile, interaction surfaces are rapidly diversifying:

- voice assistants
- mobile-first interfaces
- augmented and virtual reality
- spatial computing
- wearable devices
- ambient and IoT environments
- AI-mediated conversational interfaces

These systems often operate in isolation, creating fractured user experiences and inconsistent trust surfaces.

Examples of failure:

- A voice assistant cannot access or interpret trust overlays visible in a browser
- AR annotations misalign with textual context or policy zones
- Accessibility tools receive degraded or partial semantic information
- AI agents lose continuity when switching modalities

DP21 reframes modality as a first-class design concern. The Meta-Layer must ensure that meaning, trust signals, governance rules, and participation pathways persist across modes.

A healthy DP21 implementation must answer:

- Can participants move between modalities without losing context?
- Are trust signals preserved across visual, auditory, and spatial representations?
- Can accessibility tools fully interpret Meta-Layer structures?
- Do immersive environments respect governance and consent rules?

- Can AI agents operate consistently across modalities?

Without DP21, the Meta-Layer becomes fragmented, exclusionary, and unreliable.

3. Threats and Failure Modes

DP21 is not simply about supporting more devices. Multi-modal systems introduce new ways for meaning, trust, consent, and agency to fracture. A visual interface can show nuance through layout, color, hover states, and layered context. A voice interface may compress all of that into one sentence. An AR interface may place a signal in physical space, changing how the participant experiences authority, urgency, and social pressure.

For this reason, DP21 treats modality as a governance surface. Every new mode of interaction changes what participants can perceive, how they can respond, what data is collected, and how power is experienced.

3.1 Modality silos

Different interface modes may evolve as separate ecosystems, each with its own data model, permissions, affordances, and governance assumptions.

Example: A browser overlay supports community trust annotations, while an AR headset displays only platform-approved spatial labels. A participant moving between the two environments sees different versions of reality without knowing which guarantees changed.

Why this matters: If modalities become silos, the Meta-Layer loses its role as shared context. Trust becomes dependent on the device or vendor rather than the underlying civic substrate.

3.2 Accessibility degradation

Multi-modal systems can appear inclusive while quietly degrading experiences for people who rely on assistive technologies.

Example: A visual overlay includes provenance badges, dispute status, and community annotations, but a screen reader receives only the page text. The participant hears the claim but not the trust context that other participants see.

Why this matters: Accessibility is not equivalent to basic access. DP21 requires that meaning, trust, and agency survive translation into accessible forms.

3.3 Context fragmentation

The same object, claim, participant, or governance action may be represented differently across modalities.

Example: A controversial claim has a detailed contextual overlay in text, a short warning in voice mode, and a red spatial marker in AR. Each representation carries a different emotional weight and interpretive frame.

Why this matters: Context fragmentation can change what participants believe the system is saying. If each modality tells a slightly different story, governance becomes unstable.

3.4 Immersive capture

AR, VR, and spatial computing environments can become proprietary civic spaces controlled by headset vendors or platform operators.

Example: A city deploys public-service overlays in a vendor-controlled AR environment, but residents can only access civic annotations through approved devices and approved identity systems.

Why this matters: The Meta-Layer must not trade the flat-web platform problem for a spatial-platform problem. Immersive systems need open, governable, interoperable trust surfaces.

3.5 Voice flattening

Voice interfaces are powerful because they feel natural, but they are also compressive. They often force complex information into linear, short-form summaries.

Example: A voice assistant says, "This source is considered unreliable," without explaining by whom, according to what rubric, with what confidence level, and whether the rating is disputed.

Why this matters: Trust signals must not become authoritarian declarations when translated into voice. DP21 requires voice interfaces to preserve uncertainty, source, scope, and recourse.

3.6 Sensor overreach

Multi-modal systems often rely on cameras, microphones, location, motion sensors, biometric signals, gaze tracking, and environmental scanning.

Example: An immersive overlay requests room mapping, eye tracking, voice input, and location data to provide contextual interaction, then reuses those signals for personalization or reputation inference.

Why this matters: Multi-modal richness can become surveillance richness. DP21 must be bound tightly to DP4 data sovereignty and privacy.

3.7 AI modality drift

AI agents may behave differently depending on whether they are speaking, writing, acting in AR, or mediating a tactile or ambient interface.

Example: A text agent carefully cites uncertainty, but the same agent in voice mode speaks with confidence and omits caveats to sound more natural.

Why this matters: Participants should not receive weaker governance simply because an interaction feels conversational or immersive. Agent behavior must remain policy-bound across modalities.

3.8 Interaction inequity

Richer devices can create richer participation opportunities. Participants with headsets, high-end phones, or always-on assistants may gain more visibility, faster response, or stronger coordination capacity.

Example: A governance meeting includes spatial collaboration tools that allow headset users to annotate a shared model while phone-only participants can only watch.

Why this matters: Multi-modal capability must not become a new class divide. DP21 requires parity of agency, even when interface richness differs: richer modalities may add expressive range, but not decision power.

3.9 Representation mismatch

Different modalities may represent the same signal with different emotional or social force.

Example: A low-confidence caution shown as a small visual icon becomes a loud audio interruption in voice mode or a red warning halo in AR.

Why this matters: Representation affects behavior. Systems must map severity, confidence, and urgency carefully across modalities.

3.10 Latency fragmentation

Real-time multi-modal systems can desynchronize. A participant may see one state, hear another, and act on outdated information.

Example: During a live civic deliberation, a visual overlay updates vote status immediately, while the voice interface reads an earlier summary. Participants make decisions from inconsistent system states.

Why this matters: Trust depends on temporal coherence. Systems must signal freshness, delay, and synchronization status.

3.11 Cognitive overload

Multi-modal systems can overwhelm participants by presenting too many signals at once.

Example: A participant receives a visual warning, audio cue, haptic pulse, AI suggestion, social annotation, and governance prompt simultaneously.

Why this matters: More modalities do not automatically create more clarity. DP21 requires calm, layered, participant-directed interaction design, including the ability to switch, mute, or reroute any modality in real time, for example moving an audio cue to text or haptics, without losing the signal.

3.12 Ambient manipulation

Ambient interfaces can influence participants without being consciously noticed.

Example: A room-scale system subtly changes lighting, sound, or notification patterns to encourage agreement, urgency, purchase, or disengagement.

Why this matters: Ambient interaction can bypass reflective agency. DP21 must ensure that ambient signals remain legible, consented, and interruptible.

3.13 Physical-world safety risks

Spatial and tactile interfaces can affect movement, attention, and physical safety.

Example: An AR overlay appears while someone is crossing a street, or a haptic notification distracts a worker using dangerous equipment.

Why this matters: Multi-modal governance must include situational safety, not only information integrity.

4. Core Principle

The Meta-Layer must support multi-modal interactions such that meaning, trust, governance, and participation remain coherent and accessible across all forms of interaction.

Modality should expand agency, not fragment it.

A participant should be able to move from a browser to a phone, from text to voice, from a screen to an AR layer, or from an immersive environment back to a simple assistive interface without losing the essential structure of the interaction. Presentation may change what participants attend to and feel, but the underlying meaning, permissions, trust signals, and rights should remain intact.

Example: A participant encounters a health claim online. In a visual browser overlay, they see provenance, community annotations, and expert citations. In voice mode, they hear a concise but scoped summary: who disputes the claim, what confidence level applies, and how to request more detail. In AR, the claim is marked with a contextual indicator that opens the same evidence graph. Each modality feels natural, but all point back to the same governed semantic object.

What this feels like: The system adapts to the participant and context without changing the truth conditions of the interaction.

Without this: Each interface becomes a separate reality.

5. Primary Mechanisms and Structural Conditions

5.1 Multi-Modal Interface Layer: Meaning Before Presentation

DP21 requires a layer that separates meaning from presentation.

The system must first know what something means before deciding how to show, speak, vibrate, spatialize, or summarize it. A trust signal, governance rule, consent boundary, identity claim, or reputation state should exist as a structured object before it is rendered into any particular modality.

This layer includes:

- modality-agnostic semantic representations
- modality adapters
- cross-device state synchronization
- accessibility interfaces
- context anchoring
- privacy and sensor governance
- participant preference models
- graceful degradation rules

The goal is not to make every modality identical. The goal is to make every modality faithful.

Failure mode: **presentation-driven divergence**, where the interface invents meaning rather than rendering governed meaning.

5.2 Multi-modal device support

The Meta-Layer should support a broad range of devices, including:

- mobile phones
- desktop and laptop browsers
- tablets
- voice-only systems
- AR and VR headsets
- wearables, including smart headphones, earbuds, and hearing aids, which form a distinct class with their own voice and gesture inputs and their own accessibility stakes

- ambient devices
- public kiosks
- assistive devices
- future spatial or embodied computing systems

Device support must be capability-aware. A phone, headset, screen reader, and voice-only device cannot provide identical experiences, but each should preserve core agency.

A low-capability device should still allow participants to:

- understand relevant trust signals
- control consent
- participate in governance where appropriate
- receive feedback and recourse
- know what is missing or degraded

Failure mode: **device privilege**, where only participants with high-end devices can fully participate.

5.3 Multi-modal interface support

The Meta-Layer should support interface modes such as:

- visual overlays
- text panels
- audio narration
- voice dialogue
- haptic feedback
- spatial annotations
- gestural interaction
- conversational AI
- immersive environments
- ambient cues

Each interface mode has strengths and risks. Visual systems can carry density but may exclude non-visual participants. Voice systems feel natural but can flatten complexity. Haptics can signal urgency but can also become int-

rusive. Spatial overlays can make context embodied but may create coercive presence.

DP21 requires every interface mode to be evaluated not only for usability, but for its effect on agency, comprehension, privacy, and governance.

Failure mode: **mode enthusiasm**, where novelty drives adoption before governance catches up.

5.4 Semantic continuity

All modalities must derive from shared semantic structures.

A participant, claim, annotation, trust signal, governance rule, consent object, reputation state, or AI action should have a stable underlying reference independent of how it is presented.

For example, a provenance warning should not be three unrelated artifacts across visual, voice, and AR interfaces. It should be one semantic object with multiple renderings.

A semantic object SHOULD include:

- identifier
- object type
- scope
- provenance
- confidence level
- governance zone
- permissions
- freshness
- dispute status
- modality rendering rules

Failure mode: **duplicated interpretation**, where each modality generates its own version of the truth.

5.5 Accessibility-first design

Accessibility must be foundational.

DP21 requires that multi-modal interaction expand participation for people with visual, auditory, motor, cognitive, sensory, linguistic, or situational differences.

Accessibility support SHOULD include:

- screen reader compatibility
- voice navigation
- captions and transcripts
- audio descriptions
- haptic alternatives
- reduced motion modes
- cognitive simplification modes
- plain-language summaries
- alternative input methods
- keyboard-only navigation
- localization and translation
- low-bandwidth modes

Accessibility is not satisfied when content is technically reachable. Participants must be able to understand, decide, respond, and appeal.

Failure mode: **assistive afterthought**, where accessibility tools receive the surface content but not the governance context.

5.6 Cross-modal synchronization

State and context must persist across modalities.

If a participant starts an interaction in one mode and continues in another, the system should preserve:

- current object or claim
- governance zone
- consent state
- trust signal state
- identity context
- unresolved prompts
- participant preferences

- relevant history
- freshness markers

Example: A participant begins reviewing a civic proposal on desktop, continues by voice while commuting, and later joins an AR town hall. The system should maintain continuity without leaking private context or assuming consent across settings.

When continuity would carry state from a private context into a shared one, such as from personal reading into a public AR town hall, the transfer MUST be opt-in at each transition rather than a default.

Failure mode: **context reset**, where each modality treats the participant as starting over.

5.7 Privacy and sensor governance

Multi-modal systems often require richer data inputs. DP21 therefore requires stricter governance around sensors and environmental awareness.

Sensitive inputs may include:

- microphone data
- camera data
- location
- gaze tracking
- gestures
- biometrics
- proximity
- room geometry
- emotional inference
- movement patterns
- ambient sound
- device graph signals

Systems MUST define:

- what is collected
- why it is collected

- where it is processed
- how long it is retained
- whether it is shared
- whether it affects reputation, recommendations, or governance
- how consent can be revoked

Where possible, processing should occur locally or in privacy-preserving environments.

Failure mode: **surveillance through richness**, where better interaction becomes the justification for excessive capture.

5.8 AI mediation across modalities

AI can help translate, summarize, narrate, adapt, and synchronize across modalities. This makes AI useful for DP21, but also risky.

AI mediation SHOULD preserve:

- meaning
- source attribution
- uncertainty
- scope
- governance constraints
- participant preferences
- consent boundaries
- accessibility needs

Example: An AI assistant converting a dense trust overlay into voice should not simply say, "This is false." It should explain the relevant confidence, source, dispute status, and option to hear more.

AI modality adapters should be auditable. Communities must be able to inspect whether AI summaries systematically omit minority perspectives, soften warnings, exaggerate certainty, or change policy meaning.

Failure mode: **translation as governance**, where AI adaptation quietly decides what participants are allowed to perceive.

5.9 Spatial and immersive overlays

AR and VR shift the Meta-Layer from a screen-adjacent experience into an embodied environment. This changes the stakes.

Spatial overlays can attach meaning to:

- physical locations
- objects
- people
- events
- public infrastructure
- artworks
- classrooms
- workplaces
- civic spaces
- virtual worlds

These overlays must respect governance zones, consent, safety, and contextual integrity.

Example: A public monument may have multiple community annotations, historical context layers, accessibility routes, and civic dialogue spaces. Participants should know which layer they are seeing, who governs it, and how to contest or add context.

Failure mode: **unaccountable spatial authority**, where whoever controls the overlay controls the perceived meaning of the physical world.

5.10 Graceful degradation

When full modality support is unavailable, systems must degrade gracefully.

Graceful degradation means participants are told what is missing and given a meaningful alternative.

Examples:

- If AR is unavailable, spatial annotations can be shown as a map or list
- If audio is unavailable, voice content appears as captions or transcript

- If haptics are unavailable, urgency is represented visually or textually
- If real-time sync fails, stale data is labeled as stale

When a participant is using only some of the modalities available to them and a signal needs one they are not using, the system should alert them in the modality they are in and offer a route to one that can carry it. For example, a participant listening through earbuds hears that a dispute record needs a screen and can open it on their phone. The participant decides whether and when to switch. The alert must carry the signal's severity and required action, so a participant who does not switch still knows what is at stake.

Failure mode: **silent degradation**, where participants assume they have the full context but receive only a partial version.

5.11 Modality equivalence mapping

Important signals must have equivalent representations across modalities.

A high-risk warning may appear as:

- a visual icon and text label
- a spoken alert with scope and severity
- a haptic pulse pattern
- a spatial boundary marker
- a simplified accessibility summary

Equivalence does not mean sameness. It means each rendering preserves the function of the signal.

A modality equivalence map SHOULD define:

- signal type
- severity
- urgency
- confidence
- required participant action
- visual rendering
- audio rendering
- haptic rendering

- spatial rendering
- accessibility rendering
- cultural and regional variants

Colors, tones, gestures, and vibration patterns carry different meanings across cultures. Equivalence maps should be localizable, not only translated.

Failure mode: **missing signal class**, where a participant in one modality never receives a critical warning.

5.12 Context anchoring

Every multi-modal representation must remain anchored to shared identifiers.

Context anchors may include:

- URLs
- content hashes
- spatial coordinates
- object identifiers
- semantic graph nodes
- participant identifiers
- zone identifiers
- provenance records
- event identifiers

Anchoring allows different modalities to point to the same thing.

Example: A visual annotation on a paragraph, a voice summary of that paragraph, and an AR representation in a classroom discussion should all refer to the same underlying claim object.

For annotations, anchors should identify the relevant content version and, where necessary, its presentation or environment context. A video segment encountered in a browser, an AI interface, or a virtual environment should resolve to the intended material and its annotations. Hashes help detect byte-level change; segment mappings and contextual identifiers are still needed when encodings or presentations differ.

Anchors should be verifiable. Participants in any modality must have a way to confirm that a signal originates from the governed semantic object it claims to represent.

Anchoring also lets different clients, such as the browser extensions, apps, and assistants that participants use to view and annotate the web, point to the same thing. Where anchors use an openly specified format that any client can reproduce from the same passage, region, or segment (DP7; DP24, Section 8.3), contributions made through different clients converge on one passage rather than splitting into parallel conversations. For example (hypothetical), two participants reading the same news article through different overlays see each other's annotations on the same disputed sentence.

Failure mode: **reference drift**, where participants believe they are discussing the same object but each modality has attached context to a different target.

5.13 Multi-modal feedback loops

Participants must be able to give feedback through multiple modalities.

Feedback may be:

- spoken
- typed
- gestural
- selected from structured prompts
- submitted through assistive devices
- attached to spatial objects
- recorded as annotations
- routed through AI-assisted summaries

Multi-modal feedback must still produce structured feedback objects under DP18. A spoken concern, a tactile accessibility report, or a spatial annotation should not disappear into unstructured logs.

Failure mode: **feedback inequality**, where only participants using dominant interfaces can meaningfully shape the system.

5.14 Participant preference and control profiles

Participants should be able to define how they want the Meta-Layer to communicate with them.

Preference profiles MAY include:

- preferred modality
- reduced stimulation settings
- language
- accessibility needs
- notification style
- verbosity
- privacy thresholds
- AI mediation preferences
- sensory sensitivity settings
- default degradation behavior

These profiles must be portable where appropriate and governed by DP4 privacy constraints.

Failure mode: **interface paternalism**, where the system decides the best mode for the participant without meaningful control.

5.15 Situational awareness and safety modes

The appropriate modality may depend on context.

A participant walking, driving, working, resting, attending a meeting, or entering a sensitive civic space may need different interaction boundaries.

Systems SHOULD support situational safety modes such as:

- do-not-interrupt modes
- low-attention summaries
- delayed notifications
- physical hazard suppression
- private-context suppression
- child-safe modes

- meeting-safe modes
- public-display protections

Failure mode: **context-insensitive interaction**, where the system provides the right information at the wrong time or in the wrong sensory form.

6. Multi-Modal Intelligence and Data Fusion

DP21 supports the fusion of diverse data types, but only when fusion increases understanding without eroding governance.

Multi-modal intelligence may involve:

- text
- image
- video
- audio
- spatial data
- sensor data
- maps
- conceptual graphs
- knowledge graphs
- biometric or embodied signals where explicitly governed
- environmental context

The purpose of multi-modal intelligence is not to collect everything. It is to combine relevant signals into a more coherent understanding of a situation.

Example: During a disaster response, the Meta-Layer could combine public reports, satellite images, local sensor data, official advisories, and community annotations into a shared situational map. Participants using phones, voice assistants, or public kiosks could access the same underlying context in different forms.

DP21 requires that data fusion preserve:

- provenance
- uncertainty

- consent boundaries
- jurisdictional constraints
- community governance rules
- source diversity
- time sensitivity
- contestability

Failure mode: **data chaos**, where many signals are fused without lineage, confidence, or governance.

Failure mode: **fusion authoritarianism**, where the system produces a single authoritative view from contested inputs without preserving dissent or uncertainty.

7. Governance, Accountability, and Agency Surfaces

Multi-modal systems are governance systems because they shape perception, attention, and action.

Communities must be able to define:

- which modalities are allowed in a zone
- which sensors may be used
- what data may be collected
- how immersive overlays are labeled
- how spatial annotations are moderated
- what accessibility standards apply
- how AI may summarize or translate
- what signals require human review
- when ambient cues are prohibited
- how physical-world safety is protected

Governance artifacts SHOULD specify modality-specific rules.

Example: A school zone may allow text and visual overlays, restrict biometric sensing, require captions for all audio, prohibit persistent student tracking, and require teacher or community approval for AR layers.

Failure mode: **ungoverned modality expansion**, where new interface capabilities become active before the community has defined rules for them.

7.1 Accountability surfaces

Modality decisions are usually invisible to the people they affect. A caption is missing, a haptic cue is ambiguous, an overlay appears without a source, and there is no one to ask. DP21 requires named surfaces that make interface authority checkable:

- **A modality manifest per zone**, listing which modalities are active, which are optional, and which are prohibited
- **A sensor permission record**, stating what is sensed, at what fidelity, for what purpose, retained for how long, and processed where
- **An overlay registry** for spatial and immersive content, identifying publisher, governance zone, dispute status, and expiry
- **An accessibility conformance statement** with known gaps, dates, and remediation owners, rather than an unqualified claim of compliance
- **An AI mediation disclosure** covering summarization, translation, description, and transcription, including confidence and sampling practice (DP11–DP13)
- **A degradation log**, recording when equivalence failed, what participants saw or heard instead, and who was notified

7.2 Participant agency surfaces

Multi-modal agency means participants shape how they are addressed and sensed, not only what they are shown:

- select preferred and prohibited modalities, and have those preferences persist and port (DP2, DP7)
- refuse specific sensor classes without losing baseline participation
- request an equivalent representation of any content encountered in an inaccessible form, with a response commitment
- report modality failures through the modality they can actually use, including offline and assisted routes
- see and contest AI-generated descriptions, captions, or summaries attributed to their own contributions

- opt out of ambient, persuasive, or attention-shaping cues by default rather than by discovery
- exit an immersive context immediately, with a predictable and always-available return path

Example: A participant using a screen reader and haptic output sets a profile prohibiting ambient audio cues. Entering a public AR zone, they receive a structured spatial summary instead, with overlay sources listed and a route to challenge any unlabeled annotation.

What this feels like: The interface adapts to the person, and the person can see who decided what.

Without this: Accessibility becomes a feature request queue, and perception becomes something done to participants rather than with them.

Failure mode: **consent through exhaustion**, where refusing a modality is technically possible but practically requires abandoning participation.

8. Community Signals Informing DP21

DP21 reflects signals from disabled participants, educators, interpreters, low-bandwidth communities, immersive practitioners, and people whose primary interface is voice rather than screen.

Recurring signals include:

- "Accessibility arrives last and breaks first." Participants describe features shipped without equivalents, then repaired only after complaint.
- "The audio version leaves out what matters." Alternative modalities are treated as summaries rather than as equivalents.
- "I lost my place." Switching from phone to voice to headset loses context, state, and consent settings.
- "Voice makes everything sound certain." Participants report that spoken interfaces strip uncertainty, provenance, and dissent.
- "I don't know what is being sensed." Ambient microphones, cameras, and biometrics generate unease disproportionate to their stated purpose.
- "AR content appeared over my street." Communities want authority over spatial annotation of places they inhabit.

- "Captions are wrong in ways that change meaning." Automated transcription errors are experienced as misrepresentation, not inconvenience.
- "Immersive spaces are hard to leave." Participants ask for predictable exits, especially for youth and first-time users.
- "Low bandwidth means low status." Text-first and offline participants find themselves excluded from consequential discussion.
- "Sign language is a language, not an accommodation." Requests for first-class support rather than fallback treatment (DP23).

These signals converge on one requirement: modality must not determine standing. DP21 encodes that requirement structurally rather than leaving it to product priorities.

Community signals inform DP21 without settling it. They are recorded so that later revisions can be tested against the experiences that motivated the property.

9. Evaluation Criteria

A DP21-aligned implementation should be evaluated against the following questions.

9.1 Continuity

- Does the same object retain identity across modalities?
- Are trust signals, consent states, and governance zones preserved?
- Can participants move across devices without losing essential context?

9.2 Accessibility

- Can participants with different abilities access the same meaning and agency?
- Are accessibility tools receiving structured trust and governance information?
- Are cognitive, sensory, motor, and linguistic needs considered?

9.3 Trust fidelity

- Do all modalities preserve provenance, uncertainty, confidence, and dispute status?
- Are warnings and endorsements represented with appropriate severity?
- Are simplified summaries faithful to the underlying record?

9.4 Privacy and sensor limits

- Are sensors governed explicitly?
- Is data minimization enforced?
- Can participants revoke modality-specific permissions?
- Are local or privacy-preserving processing options used where possible?

9.5 Synchronization and freshness

- Are real-time states synchronized?
- Is stale or delayed information labeled?
- Can participants tell whether different modalities are showing the same system state?

9.6 Participant control

- Can participants choose preferred modalities?
- Can they reduce stimulation or disable intrusive cues?
- Can they control AI mediation and summarization?

9.7 Governance fit

- Are modality rules defined per zone?
- Are immersive and ambient interfaces subject to community policy?
- Are physical safety contexts handled responsibly?

10. Implementation Patterns

Implementation patterns translate DP21 from principle into practice. These are not rigid prescriptions, but recurring architectural and design approaches that help systems maintain semantic continuity, accessibility, and governance integrity across modalities. They are intended to guide builders

in making consistent decisions when adapting the Meta-Layer to new devices, senses, and interaction environments.

10.1 Semantic-first architecture

Build semantic objects before rendering interfaces. Presentation layers should adapt governed meaning, not invent it.

10.2 Modality adapters

Create adapters for visual, voice, haptic, spatial, and assistive interfaces that consume the same underlying semantic objects.

Adapters should also support modality handoff. With the participant's consent, an adapter detects which of their devices and modalities are available, treating that detection as a governed sensor input (Section 5.7). It routes a signal only to a modality the participant allows, in line with their preference profile (Section 5.14) and situational safety modes (Section 5.15). The receiving modality must carry the signal faithfully (Section 5.11). If none can, the participant is told what is missing (Section 5.10). For example, when a participant removes an AR headset, a dispute notice on a spatial overlay could move to their phone as text rather than disappear.

10.3 Accessibility testing pipelines

Every core trust, consent, governance, and reputation signal should be tested across assistive modalities.

10.4 Context synchronization engines

State should travel across devices and modalities through privacy-preserving synchronization.

10.5 Sensor permission manifests

Multi-modal applications should publish clear manifests describing sensor access, purpose, retention, and sharing.

10.6 Degradation notices

When a modality cannot show full context, the interface should say what is missing and offer alternatives.

10.7 Voice uncertainty patterns

Voice systems should use standard language for uncertainty, provenance, dispute status, and request-for-detail prompts. A spoken summary should keep the elements that the modality equivalence map lists for that signal (Section 5.11). If a summary cannot keep them, it should say what was left out and offer the full version (Section 5.10). For example, a hypothetical prompt could say: "Two community reviewers dispute this source, and confidence is low. Say 'more' to hear why."

10.8 Spatial governance labels

AR and VR overlays should display or make available the governing zone, source, scope, and contestation path for spatial annotations.

10.9 Participant sensory profiles

Participants should be able to store preferences for stimulation level, modality, language, accessibility, and AI assistance.

10.10 Multi-modal feedback receipts

Feedback submitted by voice, gesture, text, or spatial annotation should generate DP18-compatible receipts.

11. Relationship to Other Desirable Properties

11.1 DP2 – Participant Agency and Empowerment

DP21 expands agency by letting participants choose how they interact. It violates DP2 when modality choices become coercive, inaccessible, or locked to specific vendors.

11.2 DP4 – Data Sovereignty and Privacy

Multi-modal systems create richer data flows. DP21 depends on DP4 to prevent sensors, biometrics, and environmental context from becoming surveillance infrastructure.

11.3 DP7 – Simplicity and Interoperability

DP21 is an interoperability problem across senses, devices, and environments. Meaning must move across modalities without collapsing.

11.4 DP8 – Collaborative Environment and Meta-Communities

Communities must govern which modalities are appropriate in each zone. A meditation group, classroom, emergency response zone, and public debate space require different modality norms.

11.5 DP10 – Education

Multi-modal onboarding can meet participants through voice, visuals, interactive demos, spatial practice, and assistive formats.

11.6 DP12 and DP13 – AI Governance and Containment

AI agents must remain governed across text, voice, spatial, embodied, and ambient interactions. Containment cannot disappear when the interface changes.

11.7 DP14 – Trust and Transparency

Transparency must be perceivable across modalities. A trust signal that only works visually is not transparent to everyone.

11.8 DP22 – Civic Memory & Epistemic Continuity

DP21 depends on DP22 for the lineage, uncertainty, and dispute status that must survive rendering into audio, spatial, and assistive forms. Multimodal presentation is only coherent when grounded in shared semantic structures.

11.9 DP18 – Feedback Loops and Reputation

Participants must be able to provide feedback and receive reputation signals across modalities without losing structure, provenance, or recourse.

11.10 DP19 – Amplifying Presence and Community Engagement

Multi-modal experiences make the Meta-Layer more present, memorable, and culturally accessible, especially through education, public installat-

ions, youth engagement, and immersive storytelling.

12. Non-Goals and Explicit Boundaries

DP21 is often read as an argument for more interfaces. It is not. It is an argument that meaning, trust, and governance must survive the interface. The following boundaries are explicit.

DP21 does not require every system to support every modality. A text-only tool can be fully DP21-aligned. What DP21 requires is that supported modalities preserve meaning, that limits are declared, and that participants are not silently excluded.

DP21 does not mandate immersive or spatial computing. AR, VR, and ambient interfaces are permitted, not prescribed. Where they are used, they inherit heightened obligations for labeling, consent, exit, and physical safety.

DP21 does not treat modality parity as visual equivalence. Equivalence is about meaning, trust state, and available action, not about reproducing a layout in another form. A faithful audio equivalent may look nothing like the screen.

DP21 does not authorize sensing as a convenience. Cameras, microphones, biometrics, eye tracking, and location are governed under DP4 minimization. Improved personalization is not a sufficient justification for expanded sensing.

DP21 does not permit ambient persuasion. Cues that shape behavior below the threshold of notice are outside DP21 regardless of intent, because they cannot be consented to or contested.

DP21 does not delegate meaning to AI. Machine captioning, description, translation, and summarization are assistive and provisional. They may not become the authoritative record of what a participant expressed (DP11–DP14).

DP21 does not replace human interpreters or accessibility professionals. Automated equivalents extend reach; they do not discharge the obligation to provide human support where stakes are high. High stakes include, at

minimum, governance decisions, dispute and appeal processes, safety contexts, and any interaction affecting a participant's standing or reputation.

DP21 does not require perfect synchronization. Cross-modal state will sometimes diverge. DP21 requires that divergence be detectable and disclosed, not that it be eliminated.

DP21 does not make accessibility optional under deadline. Degradation must be visible and bounded. Shipping an inaccessible path with a promise of later remediation is a DP21 failure, not a phase.

DP21 does not govern content quality. Whether an artifact is accurate or worthwhile belongs to DP14 and DP22. DP21 governs whether it can be perceived, trusted, and acted upon across forms.

DP21 does not standardize hardware. Device diversity is assumed, including low-cost, older, offline, and assistive devices. Requiring specific hardware to participate meaningfully is an exclusion, not a specification.

13. Minimum DP21 Alignment

A system claiming DP21 alignment should be able to demonstrate the following. These statements are non-normative and serve as an assessment aid rather than a conformance test.

13.1 Semantic core

Content and governance objects exist independently of presentation, with meaning, provenance, trust state, and available actions defined before rendering.

13.2 Declared modality support

There is a published statement of which modalities are supported, at what fidelity, and with what known gaps, per zone or surface.

13.3 Equivalence mapping

For each supported modality, there is a documented mapping showing how identity, trust signals, consent state, uncertainty, and dispute status are conveyed.

13.4 Accessibility floor

Text alternatives, captions, keyboard and switch access, adjustable timing, and screen reader compatibility exist as baseline conditions rather than optional enhancements, with conformance and gaps published.

13.5 Context continuity

A participant can move between devices and modalities with continuity of object identity, position, consent state, and governance zone, or receive an explicit notice that continuity was lost.

13.6 Sensor governance

Every sensor use has a declared purpose, retention limit, processing location, and refusal path, and refusal does not remove baseline participation.

13.7 AI mediation disclosure

Automated captioning, description, translation, and summarization are labeled, carry uncertainty where relevant, and are correctable by the participant or community affected.

13.8 Visible degradation

When a modality cannot faithfully represent an object, the system says so in that modality rather than presenting a degraded version as complete.

13.9 Participant profiles

Sensory, cognitive, and interaction preferences are storable, portable, and respected by default across surfaces (DP2, DP4, DP7).

13.10 Safety and exit

Immersive and physical-world interactions include situational awareness limits, prohibited-context rules, and an always-available exit that returns the participant to a predictable state.

14. Open Questions and Future Work

1. What minimum semantic object model enables modality-independent rendering?
2. How should trust signals map across visual, audio, haptic, spatial, and assistive forms?
3. What is the standard way to express uncertainty in voice interfaces?
4. What sensor permissions should be mandatory to disclose?
5. How should AR overlays represent governance zone, source, and dispute status?
6. What accessibility conformance standards should apply to Meta-Layer overlays?
7. How should multi-modal feedback objects be structured?
8. How should participant sensory preferences be stored and ported?
9. What are safe defaults for ambient interfaces?
10. How should systems indicate stale or desynchronized modality states?
11. How should AI summaries be audited for modality-specific distortion?
12. What physical-world safety constraints should apply to AR and haptic interactions?
13. How should multi-modal systems operate in low-bandwidth or crisis conditions?
14. What rights should participants have when modality-specific limitations affect governance participation?

15. Path Toward ML-RFC

DP21 is currently an ML-Draft and serves as exploratory scaffolding for how multi-modal interaction becomes part of Meta-Layer infrastructure.

Advancement toward ML-RFC status SHOULD require:

- a minimal semantic object model for modality-independent rendering
- a modality equivalence mapping standard
- accessibility validation requirements
- sensor permission and privacy manifest standards
- spatial overlay governance labels

- voice uncertainty and provenance patterns
- cross-device synchronization requirements
- graceful degradation requirements
- multi-modal feedback receipt formats
- pilot implementations across at least three modalities

ML-RFC promotion SHOULD be contingent on:

- rough consensus among participating communities
- demonstrated accessibility across multiple participant needs
- alignment with DP4 privacy requirements
- alignment with DP7 interoperability requirements
- alignment with DP22 requirements that lineage, uncertainty, and dispute status survive rendering
- evidence that modality support expands agency rather than creating new inequities

Early ML-RFC candidates may focus on:

- Multi-Modal Semantic Object Standard
- Modality Equivalence Map
- Sensor Permission Manifest
- Spatial Trust Signal Encoding
- Voice Trust and Uncertainty Pattern Library
- Accessibility Requirements for Meta-Layer Overlays

DP21 will likely mature through several component RFCs rather than one monolithic standard.

16. Closing Orientation

DP21 ensures the Meta-Layer is not bound to a single interface, device, or sense.

It allows people to experience trust, context, and participation in ways that match their abilities, environments, preferences, and tools. It makes the Meta-Layer usable by more people, in more places, under more conditions, without sacrificing coherence or governance.

A DP21-aligned meta-layer does not treat voice, AR, haptics, accessibility tools, and immersive systems as side channels. It treats them as legitimate civic interfaces.

Interaction becomes flexible. Meaning becomes portable. Trust becomes perceivable. Presence becomes multi-dimensional.

This is how the Meta-Layer becomes ambient, inclusive, and truly interoperable.

Continue the conversation on DP21 online →

DESIRABLE PROPERTY 22

DP22

Civic Memory & Epistemic Continuity

The Conditions for Durable Collective Understanding



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp22

1. Orientation: What Deserves to Be Remembered

Imagine there is someone from history you wish you understood better. An inventor whose notebooks were lost. A philosopher whose unpublished thoughts disappeared. An ordinary person living through extraordinary events.

Now imagine they had been able to permanently preserve a few pages of writing, several photographs, a short recording, or a carefully curated collection of the ideas they believed future generations should inherit. Suppose preserving those artifacts had cost less than a cup of coffee.

Would we know more about them today?

Almost certainly.

The opportunity extends far beyond historical curiosity. If every generation had been able to intentionally preserve what mattered most, every generation after them would have inherited a richer foundation for learning, understanding, and decision-making.

History is filled with knowledge that disappeared, not because it lacked value, but because the means to preserve it were fragile, expensive, centralized, or temporary.

For the first time, humanity has the ability to intentionally create durable public memory that can remain accessible for centuries.

The question is no longer whether enduring knowledge can be preserved.

The question is:

What deserves to become part of humanity's permanent memory?

DP22 does not answer that question. It describes the conditions under which communities can keep answering it, and keep revisiting their answers, without losing the ability to see what earlier answers were.

2. Purpose of This Draft

This draft articulates Desirable Property 22 (DP22) as the condition under which the Meta-Layer preserves civic memory and collective sensemaking through transformation without turning memory into surveillance, summ-

ary into authority, plurality into noise, or history into platform-controlled narrative.

DP22 begins from a distinction that is easy to overlook:

Information persistence is not civic memory.

A society may preserve files, posts, inscriptions, records, archives, governance artifacts, media objects, and AI summaries while still losing the pathways by which civic meaning was formed. What disappears is not always the data itself, but the relationships between artifact, evidence, interpretation, disagreement, authority, governance, and later feedback.

DP22 addresses that missing continuity.

The Meta-Layer should preserve not only civic artifacts, but the transformation paths through which civic understanding evolves. Participants and communities should be able to reconstruct how a claim became evidence, how evidence became interpretation, how interpretation became synthesis, how synthesis informed governance, and how governance later changed in response to feedback.

This property becomes especially important in AI-mediated environments. AI systems increasingly summarize, rank, retrieve, cluster, recommend, synthesize, compress, and reinterpret civic knowledge. These acts are not neutral compression. They shape what future participants believe the past meant.

DP22 therefore treats summarization, retrieval, indexing, ranking, transformation, and contextualization as civic memory operations requiring accountability.

DP22 does not establish a single truth authority. It defines the conditions under which plural sensemaking can remain traceable, navigable, contestable, and accountable across time.

Continuity also depends on artifacts that are built to last. Digital artifacts and digital monuments do more than preserve information. They create **reality anchors**: stable, publicly discoverable reference points that remain available regardless of changing platforms, institutions, or political environments.

Reality anchors provide shared orientation. They do not dictate conclusions, and they do not eliminate disagreement. What they establish is that material of lasting importance cannot simply disappear from collective awareness. Just as physical civilizations built monuments, libraries, observatories, and landmarks to orient future generations, digital civilization requires enduring landmarks for navigating knowledge itself.

3. Reality Anchors and Collective Orientation

The defining difficulty of contemporary information environments is not only misinformation. It is fragmentation. Participants increasingly operate from different contexts, different memories, different trusted sources, and different working understandings of what has happened.

Reality anchors are one structural response to that fragmentation.

A reality anchor is an enduring digital artifact or digital monument that becomes widely recognized within a community, a discipline, or a field. Its standing does not depend on universal agreement. Its value comes from becoming common knowledge: something people know exists, know that others know exists, and therefore orient their reasoning around.

Where reality anchors exist, conversations change. Participants no longer begin from disconnected personal memory alone. They begin from a shared landscape of durable reference points. They remain free to disagree, to critique, to reinterpret, and to reject conclusions. What they cannot honestly do is proceed as though the anchor does not exist.

Disagreement then occurs in relation to shared landmarks rather than isolated personal narratives, which is what makes it legible enough to be worked with.

An anchor is only useful if it can be found, trusted, and returned to. DP22 therefore expects reality anchors to satisfy conditions that ordinary storage does not:

- **Discoverable.** Resolvable through a persistent public reference rather than through a single platform's index, so that finding it does not depend on the goodwill of an intermediary (DP5, DP15).

- **Attributable.** Carrying authorship or stewardship, time of creation, and the conditions under which it entered the record (DP1).
- **Verifiable.** Integrity checkable against alteration, so that a later modification appears as a transformation rather than as the original.
- **Interpretively open.** Able to accumulate annotation, dissent, correction, and later context without the anchor itself being rewritten.
- **Custodially accountable.** Someone is identifiably responsible for continued availability, and that responsibility can transfer rather than lapse silently (DP20).

Reality anchors are not authorities, and they can be wrong. An anchor establishes that something was recorded and is available for examination; it does not establish that the record is accurate or complete, and it does not author its own meaning. The interpretation layers, transformation records, and governance lineage defined elsewhere in this draft are a check on anchors, so that a recorded existence never hardens into an official account and an anchor can be contested, corrected, or superseded.

Failure mode: **anchor capture**, where an artifact becomes widely recognized and whoever controls its resolution, hosting, or presentation acquires interpretive authority over everything organized around it.

4. Problem Statement

The modern Internet does not simply forget. It remembers badly.

It preserves fragments while dissolving the structures that make those fragments meaningful. A quote survives without context. A screenshot survives without provenance. A governance decision survives without dissent. A summary survives without omissions. A digital artifact survives without the civic role it later came to play.

The result is discontinuous civic understanding.

Communities repeatedly restart the same debates because prior reasoning is inaccessible or non-navigable. Institutions preserve outputs while losing the conditions that justified them. AI systems increasingly compress contested fields into apparently coherent summaries. Search and ranking syst-

ems quietly determine what becomes socially remembered by determining what is easiest to retrieve.

The consequence is civilizational learning failure.

DP22 responds by treating memory as relational, governed, contextual, transformation-sensitive, retrieval-sensitive, and socially navigable. The core questions are therefore not only "What should persist?" but also: What context must survive for an artifact to remain interpretable? What transformation history must remain visible for a summary to be trusted? What disagreement must remain visible for plural sensemaking to survive? What should be attenuated, sealed, or forgotten to prevent memory from becoming punishment? Who governs retrieval, ranking, contextual overlays, and interpretive authority?

Without answers to these questions, civic memory becomes either too weak to support accountability, or too rigid and totalizing to support humane civilization.

5. Threats and Failure Modes

5.1 Memory laundering

Transformation chains erase inconvenient origins while preserving conclusions.

Example: AI systems summarize prior summaries until provenance, dissent, and uncertainty disappear, while the final synthesis appears authoritative.

Why this matters: Interpretive authority detaches from reconstructable evidence.

5.2 Context collapse through compression

Compression removes uncertainty, dissent, historical nuance, or scope conditions.

Example: A contested governance conflict later appears as a clean consensus because summaries preserved the decision but not the disagreement.

Why this matters: Communities inherit false institutional memory.

5.3 Retrieval capture

Ranking and retrieval systems silently determine what becomes practically rememberable.

Example: Commercial optimization consistently surfaces emotionally viral interpretations over historically grounded ones.

Why this matters: Retrieval intermediaries become practical memory authorities.

5.4 Narrative enclosure

Platforms, states, institutions, or dominant intermediaries monopolize authoritative interpretive layers.

Example: Only institutionally approved overlays appear above historical artifacts, while alternate contextualizations become hidden or difficult to discover.

Why this matters: Plural civic memory collapses into managed narrative.

5.5 Synthetic consensus generation

AI systems create the appearance of agreement through scale, repetition, or automated interpretation.

Example: Thousands of AI-generated contextual notes flood a contested event, making one interpretation appear socially settled.

Why this matters: Plurality becomes statistically drowned rather than deliberatively resolved.

5.6 Epistemic spam flooding

Low-cost synthetic interpretation overwhelms navigability.

Example: Every civic artifact accumulates large volumes of low-value commentary, automated objections, and derivative summaries.

Why this matters: Signal collapses into interpretive exhaustion.

5.7 Governance lineage collapse

Decisions survive without reconstructable rationale.

Example: A policy remains active, but later participants cannot reconstruct the evidence, dissent, amendments, or procedural choices that produced it.

Why this matters: Governance becomes ahistorical and difficult to contest responsibly.

5.8 Memory permanence as punishment

Persistent visibility eliminates humane contextual decay.

Example: Minor historical actions remain permanently surfaced without proportionality, changed context, or restorative pathways.

Why this matters: Civic memory becomes socially carceral.

5.9 Fork invisibility

Communities diverge interpretively without visible lineage.

Example: A civic archive forks into multiple historical accounts, but participants cannot see where the interpretations diverged or why.

Why this matters: Disagreement becomes opaque rather than intelligible.

5.10 Authenticity without interpretive accountability

An artifact is provably authentic but contextually misleading.

Example: A genuine video clip is surfaced without the surrounding sequence, later correction, or contradictory evidence.

Why this matters: Authenticity alone does not preserve civic meaning.

6. Core Principle

Civic memory within the Meta-Layer is the accountable continuity of meaning across artifacts, evidence, interpretation, transformation, governance, and feedback.

The Meta-Layer should preserve enough structure for participants and communities to reconstruct how civic understanding evolved through time without forcing all communities into one canonical interpretation.

First, digital artifacts must remain interpretable as their civic role changes. An artifact may begin as expression, later become evidence, later become

governance input, and later become historical memory.

Second, transformations must remain accountable. Summaries, translations, AI outputs, edits, governance reframings, and syntheses should preserve what changed, what was omitted, what uncertainty remains, and what authority the transformed object possesses.

Third, disagreement must remain structured rather than erased. Plural sensemaking requires visible disagreement, but also resilience against spam contradiction, manufactured uncertainty, and adversarial epistemic flooding.

Fourth, memory must remain governable. Communities require the ability to define what is preserved, surfaced, contextualized, attenuated, sealed, forgotten, exported, and forked.

Fifth, retrieval and navigation must be treated as civic power. What systems surface becomes what communities practically remember.

Sixth, some artifacts must be allowed to endure. Durable, publicly discoverable artifacts give plural interpretation something stable to be plural about, and the Meta-Layer should keep the preserved artifact distinguishable from the evolving interpretation that gathers around it.

Anchor Principle: Memory is not storage. Memory is navigable continuity.

7. Primary Mechanisms and Structural Conditions

DP22 becomes real through specific primitives rather than through a commitment to preservation in principle. Every system already stores things; almost none of them can reconstruct how understanding changed. The difference lies in whether transformation, interpretation, and retrieval leave structured traces that later participants can follow.

The mechanisms below define the minimum structural conditions for civic memory. The domain sections that follow expand them into the civic memory layer, the durable anchors and monuments that form its slowest stratum, its operational conditions, its governance requirements, and its interface expression.

7.1 Memory objects with durable reference

Civic memory operates over addressable objects rather than over documents alone. A memory object carries a persistent identifier, a content reference, and the surrounding context required to interpret it outside its origin: authorship or attribution, time, civic role, source relationships, and governance status (DP1, DP5, DP15).

Identifiers must remain resolvable when the hosting system changes, and resolution must degrade visibly rather than silently returning a substitute.

Failure mode: **artifact orphaning**, where the object survives and its meaning does not.

7.2 Transformation records

Every operation that changes what a future participant will encounter is recorded as a transformation with a typed relationship to its source. Summarization, translation, editing, redaction, clustering, adaptation, and governance reframing are all transformations.

A transformation record states what changed, what was omitted, what uncertainty remains, who or what performed it, and what authority the result may claim. AI-performed transformations are labeled as such, with model identity and confidence where available (DP11–DP14, DP23).

Failure mode: **semantic laundering**, where meaning changes and the path disappears.

7.3 Interpretation layers

Interpretation is represented as a layer attached to an object rather than as a replacement for it. Annotations, dissents, community readings, contextual notes, and corrections coexist, each with its own provenance and standing.

No layer becomes canonical by default. Where a community elevates one reading, the elevation is itself a recorded governance act rather than a silent property of the interface (DP8, DP12).

Failure mode: **interpretive monopoly**, or its inverse, **interpretive overload**, where plurality becomes unnavigable.

7.4 Retrieval receipts

Because what is surfaced becomes what is remembered, retrieval is treated as a civic operation that must be able to account for itself. A retrieval receipt records what query or context produced a result set, what ranking assumptions applied, what was deprioritized, and what alternative framings exist.

Participants must be able to request the receipt for what they were shown, and communities must be able to audit retrieval behavior over time (DP14).

Failure mode: **retrieval capture**, where the power to surface becomes the power to remember.

7.5 Governance lineage

Decisions carry their pathway: the question as posed, the deliberation, the amendments, the dissent, the outcome, the execution, and any later reversal. Lineage attaches to the artifacts the decision governs, so that encountering a rule also means being able to encounter its history (DP3, DP12, DP20).

Failure mode: **institutional amnesia**, where governance artifacts remain and governance understanding disappears.

7.6 Attenuation and sealing policies

Memory systems require governed forgetting. Attenuation policies define how visibility decays, under what conditions material may be sealed or bounded, who may request attenuation, and what remains discoverable as a record that attenuation occurred.

Attenuation is a policy applied to visibility, not a deletion of accountability. Communities may adopt different regimes, but the regime in force must be legible (DP2, DP4, DP18).

Failure mode: **permanent punitive memory**, or its inverse, **ungoverned erasure**.

7.7 Portability envelopes

Memory objects, transformation records, interpretation layers, and governance lineage must be exportable together. An envelope that carries content without lineage produces archives that cannot be reasoned about.

Where a receiving system cannot represent part of the envelope, the loss is declared rather than silently dropped (DP7).

Failure mode: **continuity loss at the boundary**, where memory survives migration and meaning does not.

7.8 Uncertainty and dispute signaling

Confidence, contestation, and unresolved status travel with objects rather than existing only in the systems that generated them. A disputed claim remains visibly disputed when it is quoted, summarized, translated, or rendered in another modality (DP14, DP21, DP23).

Failure mode: **confidence inflation**, where uncertainty is lost at every hop until a contested claim reads as settled fact.

Example: A municipal decision object carries its deliberation record, three community annotation layers, an AI summary marked with omissions and confidence, a retrieval receipt explaining why it appeared in a search result, and an attenuation policy specifying that named complainant details become access-bounded after two years while the decision itself remains public.

What this feels like: History is something you can walk back through rather than something you are told.

Without this: Systems accumulate records while communities lose the ability to reconstruct how they came to believe anything.

8. Civic Memory Layer: Continuity, Transformation, and Retrieval

DP22 requires a civic memory layer capable of preserving meaning across transformation. This layer is not merely an archive. It is the connective inf-

rastructure that allows artifacts, interpretations, summaries, governance decisions, disputes, and later revisions to remain intelligible across time.

8.1 Artifact continuity

Civic memory begins with durable artifacts, but it cannot end there. Records, media objects, governance drafts, annotations, evidence collections, AI summaries, and contextual overlays require stable references, provenance continuity, and enough surrounding context to remain interpretable outside their original environment.

A failure mode is artifact orphaning, where an object survives but the context needed to understand its civic meaning disappears.

8.2 Transformation accountability

Civic memory must track how artifacts are transformed. Summaries, translations, edits, AI syntheses, educational adaptations, and governance reframings change what future participants encounter. These transformations should preserve visible lineage, meaningful omissions, uncertainty, source relationships, and authority boundaries.

A failure mode is semantic laundering, where meaning changes while the transformation path disappears.

8.3 Interpretive plurality

Civic memory must preserve disagreement without allowing disagreement to become unnavigable noise. Competing interpretations should remain visible with enough context to understand why they differ, what evidence they rely on, and where their lineage diverges.

A failure mode is interpretive monopoly, where one layer silently becomes canonical, or interpretive overload, where plurality becomes unusable.

8.4 Retrieval accountability

In information-saturated environments, retrieval determines practical memory. Search ordering, recommendations, ranking systems, context windows, and AI summaries shape what communities encounter and what they later treat as history.

DP22 therefore requires retrieval transparency and contestability. Participants should be able to understand why a memory object was surfaced, what assumptions shaped its ranking, and what alternatives were suppressed or deprioritized.

A failure mode is retrieval capture, where the power to surface becomes the power to remember.

8.5 Governance memory

Governance systems must preserve not only outcomes, but the pathways that produced them. Decisions require reconstructable rationale, amendments, dissent, unresolved tensions, and later reversals. Without this, communities cannot learn from their own history.

A failure mode is institutional amnesia, where governance artifacts remain but governance understanding disappears.

8.6 Humane forgetting and attenuation

Civic memory must support forms of contextual decay, proportional visibility, restorative forgetting, and sealed or bounded access. A society that remembers everything equally becomes socially unlivable, while a society that forgets without governance becomes manipulable.

A failure mode is permanent punitive memory, where historical visibility becomes totalizing and carceral.

9. Digital Monuments as Lighthouses in the Knowledge Commons

Knowledge commons require landmarks. Sailors navigate by lighthouses, islands, constellations, and charts. Societies likewise require stable points around which understanding can organize itself.

Digital monuments serve this role. A digital monument is an artifact, or a curated collection of artifacts, published with the explicit intention that it remain available and interpretable well beyond the lifespan of the system that produced it. Monument status is a commitment about durability and custody, not a claim about correctness.

Monuments are not destinations that dictate a single path. Every navigator remains free to choose a course, and everyone benefits from knowing where the landmarks stand.

As collections of digital monuments accumulate, each field of knowledge can develop its own navigational map. Scientists, historians, artists, communities, cities, and cultures each gain an evolving constellation of shared reference points that orient future exploration.

Because monument status is not a truth claim, a monument may preserve a mistaken theory, a contested account, or a record that later generations judge harshly. What it preserves is the ability to encounter the thing itself rather than only later descriptions of it. Where a monument is translated, summarized, or rendered into another modality, the derived version is a transformation carrying its own provenance rather than an inheritor of the monument's standing (DP21, DP23).

9.1 Design pattern: lighthouses, not destinations

Digital monuments function like lighthouses. They do not tell a society where to go. They provide stable landmarks that make navigation possible.

Different communities may chart different courses. Different interpretations may emerge. Different conclusions may be reached. Everyone still benefits from knowing where the landmarks stand.

10. The Slowest Layer of the Meta-Layer

The Meta-Layer operates across several timescales at once. Conversations evolve rapidly. Annotations change. AI summaries are revised. Communities keep reinterpreting. These are the fast layers, and their speed is a feature rather than a defect.

Digital artifacts and digital monuments occupy the opposite extreme. They are intentionally stable, and they constitute the slowest content layer of the Meta-Layer.

The slow layer does not replace fast interaction. It gives fast interaction something to be about. The monument remains unchanged while interpret-

ation continues to evolve around it, which is what allows history to remain stable while understanding continues to improve.

The layers can be distinguished by how change is expected to occur:

- **Fast.** Conversation, annotation, dispute, and live summarization, which are expected to change continuously.
- **Medium.** Governance decisions, curated collections, and revisable syntheses, which change through recorded process.
- **Slow.** Durable artifacts and monuments, where change is not expected at all, and where any change is itself an event requiring a record.

This separation carries a design requirement. The preserved artifact and its evolving interpretation must remain distinguishable wherever they are encountered. An interface that merges them, presenting current interpretation as though it were the preserved thing, destroys the property that made the slow layer valuable.

Failure mode: **slow-layer drift**, where an artifact presented as durable is quietly revised and everything organized around it shifts without anyone observing the shift.

11. Digital Monuments and the Content Layer

The web introduced the web page. The Meta-Layer introduces the digital monument.

Digital artifacts and digital monuments are the Meta-Layer's enduring presence within the underlying content layer. Their permanence provides the stable foundation on which richer interaction can be built. Through bridges, relationships, annotations, AI assistance, and contextual discovery, knowledge held inside a monument can be surfaced at the moment it becomes relevant rather than waiting to be discovered by accident.

Monuments preserve the past; the Meta-Layer activates it in the present.

This is the durable counterpart to the conditions described under Civic Memory as Interface-Level Infrastructure below. Monuments supply the objects that endure, while interface-level memory supplies the contextual surfaces through which those objects are encountered. Surfacing a monum-

ent is itself a retrieval operation and inherits the retrieval accountability described under Primary Mechanisms and Structural Conditions: a participant who encounters a monument in context should be able to learn why it appeared there.

Activation must not overwrite preservation. An overlay, excerpt, or assistant response drawn from a monument is a derived representation, and the route back to the unmediated artifact remains part of what the monument guarantees.

12. Psychological and Sociological Effects

Reality anchors change more than information environments. They change how people write, how communities argue, and what cultures treat as worth keeping.

When participants know their work may become part of an enduring public record, they tend to communicate differently. They document more carefully. They supply more context. They write past their immediate audience toward readers who will not share it.

Communities likewise begin to orient around enduring landmarks rather than temporary trends. Shared memory strengthens coordination, and disagreement becomes more productive where participants acknowledge common reference points while still reaching different conclusions.

These effects are opportunities rather than guarantees. Reality anchors do not produce honesty; they widen the opening for honest engagement. The same durability that encourages care can also encourage performance for posterity, strategic omission, and the framing of a record for future readers at the expense of present ones. Permanence also bears unequally. A participant with resources can curate what endures about them, while a participant without them is more likely to be remembered through records made by others.

DP22 therefore pairs monument-making with the humane memory conditions defined elsewhere in this draft. Attenuation, proportionality, and restorative pathways apply to the condition of being permanently recorded, not only to platform visibility.

What this feels like: Writing for people you will never meet.

Without this: Durability rewards whoever is best positioned to curate their own record.

13. Civilizational Compounding

Digital monuments produce three compounding effects, each operating on a longer horizon than the one before it.

13.1 We know more

Future generations inherit richer records of the people, ideas, discoveries, and experiences that shaped earlier civilizations. What was previously preserved by accident becomes preserved by intention.

13.2 We make better decisions

Better inherited knowledge widens the opportunity for wiser decisions across science, governance, education, engineering, and culture. Inheritance does not guarantee wisdom. It removes one recurring reason for its absence, which is having to rediscover what was already understood.

13.3 We engage differently

Knowing that ideas may become part of humanity's enduring memory encourages stewardship, responsibility, and intentionality in what is created now.

The value of durable artifacts therefore compounds across generations, and each generation has the opportunity to leave a richer foundation than it inherited.

Compounding is not automatic. It depends on the conditions this property defines: that anchors remain discoverable, that transformation remains recorded, that plurality survives compression, and that custody outlasts the institutions that established it (DP17, DP20).

14. Collective Intelligence and Civilizational Continuity

DP22 is ultimately concerned with more than archival integrity.

It concerns whether civilizations remain capable of learning across generations under conditions of AI-mediated interpretation.

Historically, societies preserved memory through libraries, oral traditions, institutions, monuments, archives, and governance records. These systems allowed civilizations to transmit partial continuity of understanding across time. But the Internet transformed not only how information moves, but how meaning itself is encountered.

Increasingly, societies do not encounter history directly.

They encounter synthesized memory environments shaped by retrieval systems, AI summaries, contextual overlays, ranking systems, semantic clustering, recommendation systems, and interpretive compression.

This represents a profound civilizational transition.

AI systems increasingly mediate not only information access, but semantic continuity itself.

The core danger is therefore deeper than misinformation.

Civilizations may lose continuity of meaning while still preserving enormous quantities of information. Interpretation itself becomes unstable as ranking systems shift, summaries mutate, retrieval logic changes, and contextual environments reorganize collective understanding.

This creates the possibility of large-scale semantic drift: the gradual transformation of civic understanding through opaque interpretive infrastructure.

DP22 attempts to preserve the conditions under which societies remain intelligible to themselves across time.

Collective intelligence depends not only on communication, but on reconstructable continuity of reasoning across generations. Communities capable of preserving governance lineage, disagreement pathways, interpretive forks, contextual memory, and transformation history gain the ability to learn recursively rather than merely react episodically.

This is one of the deepest implications of the Meta-Layer.

The Meta-Layer does not merely preserve archives.

It introduces the possibility of living civic memory infrastructure.

Historically, memory was something civilizations visited through libraries, databases, records, and institutions.

Within the Meta-Layer, memory becomes ambient contextual infrastructure continuously surrounding interpretation itself.

Governance lineage can follow artifacts. Context can travel with claims. Interpretive forks can remain visible above the webpage. Transformation histories can remain queryable at runtime.

Memory shifts from destination into environment.

This represents a civilizational phase transition from static archives toward active contextual memory systems.

The challenge is not to eliminate disagreement.

The challenge is to preserve enough continuity for disagreement itself to remain intelligible.

15. Operational Conditions for Civic Memory

For DP22 to function meaningfully, civic memory cannot remain merely archival. It must operate as live infrastructure embedded within the runtime conditions of digital society.

This requires systems capable of preserving continuity while meaning is actively transformed.

A civic memory layer must therefore support:

- reconstructable transformation lineage
- contextual overlays tied to source provenance
- interpretable retrieval pathways
- contestable summarization systems
- durable semantic references
- memory portability across communities and platforms
- visible uncertainty signaling
- governance-aware contextualization

These conditions are not purely backend concerns.

They shape how participants encounter reality at the point of interaction.

As AI systems increasingly mediate interpretation, the operational conditions surrounding retrieval and contextualization become inseparable from civic governance itself. A memory system that cannot expose why a summary appeared, why a source was prioritized, or what interpretive assumptions shaped a synthesis gradually becomes opaque infrastructure governing historical understanding.

DP22 therefore requires that civic memory remain operationally legible.

Participants should be able to trace not only where information came from, but how it arrived before them, what transformations shaped it, and what alternate contextualizations remain available.

Without operational transparency, memory infrastructure silently becomes narrative infrastructure.

16. Governance, Accountability, and Agency Surfaces

Civic memory cannot remain healthy without governance.

Questions surrounding preservation, attenuation, retrieval weighting, contextual overlays, interpretive authority, and historical revision are not purely technical design questions. They are governance questions concerning legitimacy, accountability, and collective power.

The Meta-Layer therefore requires governance systems capable of mediating disputes surrounding:

- contextual framing
- retrieval prioritization
- memory attenuation
- historical reinterpretation
- provenance disputes
- interpretive forks
- AI-generated synthesis
- community annotation rights

- archival sealing pathways
- restorative visibility protocols

Different communities may legitimately adopt different memory norms.

Some communities may prioritize radical transparency and maximal preservation. Others may prioritize restorative pathways, bounded resurfacing, contextual decay, or differentiated visibility. DP22 does not impose a single universal memory regime.

Instead, it requires that memory governance remain visible, participatory, contestable, and interoperable.

The deeper challenge is that memory governance increasingly occurs through infrastructure decisions that appear politically neutral. Search ranking, contextual overlays, recommendation systems, semantic clustering, and AI summarization all shape collective memory while often avoiding democratic accountability.

DP22 therefore insists that retrieval and interpretation remain governable civic functions rather than invisible platform prerogatives.

A civilization that cannot govern how memory is surfaced eventually loses the ability to govern how reality is interpreted.

These conditions become real only where they are checkable and actionable by the people affected. The surfaces below are what turn a memory regime into something a community can inspect, contest, and hold to account. Without them, memory governance is indistinguishable from an editorial policy held privately.

16.1 Accountability surfaces

- **A memory policy of record** per community or zone, stating what is preserved, for how long, at what visibility, and under what attenuation rules
- **A transformation index** allowing any surfaced object to be traced back through summaries, translations, and edits to its sources
- **A retrieval governance record** documenting ranking objectives, weighting changes, and who authorized them

- **An interpretation registry** identifying which annotation layers exist, who maintains them, and which have been elevated by governance action
- **An attenuation and sealing log** recording that a change in visibility occurred, by whose authority, and under which policy, even where the content itself is bounded
- **A dispute docket** listing open provenance, framing, and classification disputes with owners and status
- **A custody statement** identifying who holds the archive, under what obligations, and what happens on dissolution or transfer (DP20)

16.2 Participant agency surfaces

Civic memory concerns people who are remembered, not only records that are kept. DP22 therefore expects participants to hold usable powers over their own presence in collective memory:

- annotate or contextualize an object that concerns them, with the annotation carried alongside the object
- contest a transformation, including a summary or translation that misrepresents their contribution (DP23)
- request attenuation, sealing, or bounded resurfacing under published criteria, and receive a reasoned decision
- ask why something was surfaced to them, and obtain the retrieval receipt behind it
- see, and object to, the elevation of one interpretation over others in a space they participate in
- export their own contributions and the context surrounding them (DP4, DP7)
- appeal a memory decision to a body that did not make it

When a contribution is summarized, edited, translated, or interpreted by AI, the system should keep a clear connection to the original contribution and its context. This context may include the language, culture, technology, and physical or social environment in which the contribution was made. These conditions can shape its meaning.

When practical, participants should be able to review summaries or interpretations attributed to them before publication. They should also be able to challenge misrepresentation afterward.

Giving someone credit for a contribution does not mean they agree with a later interpretation. These protections should still allow independent commentary and reporting in the public interest.

Example: A participant discovers that an AI summary of a public hearing attributes to them a position they did not take. They file a transformation challenge, the summary is annotated with the dispute while under review, and the resolution and its reasoning become part of the object's lineage.

What this feels like: You are a participant in the record, not merely a subject of it.

Without this: Memory governance becomes an administrative function performed on people who cannot reach it.

Failure mode: **archival paternalism**, where stewards decide what a community should remember about its own members without a route to challenge.

17. Civic Memory as Interface-Level Infrastructure

DP22 is not only about archives or backend repositories. It is also about what becomes visible at the point of interaction.

The Meta-Layer can make civic memory available above the webpage through overlays, contextual affordances, provenance displays, governance lineage views, and plural interpretation surfaces. These interface-level memory systems allow participants to encounter not just an artifact, but the civic context around it.

Example: A participant hovers over a policy document and sees prior versions, contested amendments, dissenting annotations, linked evidence, related governance forks, and AI-generated summaries marked with uncertainty and transformation lineage.

Example: A historical artifact displays multiple community interpretations rather than a single canonical framing, with visible provenance trees and unresolved disputes.

Example: An AI-generated summary exposes source relationships, omitted themes, compression choices, and confidence limits.

These are not decorative features. They are memory governance at the point of interpretation.

The interface becomes a place where civic memory is reconstructed, contested, and governed.

18. Relationship to Persistent Digital Object Systems

DP22 aligns with and extends prior work on persistent digital object systems, including Digital Object Architecture, the Handle System, FAIR Digital Objects, linked data systems, and persistent archival infrastructures.

These systems address important infrastructure problems related to persistent identifiers, durable addressability, interoperable references, provenance continuity, machine-readable metadata, and long-term digital preservation. Such systems may provide an important substrate for civic memory because collective sensemaking depends upon stable referential anchors.

Without persistent identity, provenance collapses, citations rot, interpretive lineage breaks, governance trails fragment, and memory becomes unstable across systems and time.

However, DP22 extends beyond artifact persistence alone.

Persistent object systems preserve continuity of digital artifacts. DP22 preserves continuity of civic meaning surrounding those artifacts.

Persistence alone does not create civic memory. Archives may survive while societies lose the ability to reconstruct how understanding evolved.

19. Relationship to Provenance and Authenticity Standards

DP22 may interoperate with provenance and authenticity standards such as C2PA, cryptographic provenance systems, content authenticity infrastructures, and transformation disclosure protocols.

These systems contribute important primitives for media authenticity, transformation accountability, AI transformation disclosure, provenance continuity, and resistance to semantic laundering.

However, provenance alone is insufficient for civic memory continuity.

Authenticity does not preserve disagreement, governance rationale, interpretive plurality, uncertainty, or collective sensemaking structure. A perfectly authentic artifact can still mislead, erase dissent, distort history, or produce false consensus.

DP22 therefore extends beyond authenticity into contextual continuity, interpretive accountability, plural sensemaking, governance lineage, retrieval accountability, and humane memory governance.

20. Community Signals Informing DP22

DP22 emerged from recurring signals across archivists, journalists, moderators, municipal staff, researchers, and participants who have been misremembered by systems they could not correct.

Recurring signals include:

- "The link is dead and the context is gone." Participants describe civic reference rot as a routine experience rather than an exception.
- "The summary is what people read." Communities observe that AI-generated compression has become the practical record, regardless of source availability.
- "We keep relitigating settled questions." Governance participants describe losing the reasoning behind past decisions within a few cycles.
- "One old post defines me." Participants ask for proportionate visibility rather than permanent maximal exposure.
- "Search decides what is true." Communities recognize retrieval ranking as an unaccountable editorial power.
- "Our dissent disappeared." Minority positions are frequently removed from summaries while formally remaining in archives.
- "The archive exists but nobody can use it." Preservation without navigation is experienced as absence.

- "We were translated into someone else's framing." Cross-language transformation is reported as a common source of meaning loss (DP23).
- "Nobody will say who decided to remove it." Attenuation and moderation actions without records generate lasting distrust.
- "Authenticity did not help." Cryptographically verified media still produced false consensus because context was stripped.

These signals converge on a request that is structural rather than sentimental: preserve the ability to reconstruct, and make the power to surface accountable.

Community signals inform DP22; they do not settle it. They are recorded so that future revisions can be evaluated against the experiences that produced the property.

21. Evaluation Criteria

A DP22-aligned implementation should be evaluated against the following questions. They are diagnostic rather than normative, and are intended to expose the difference between storage and civic memory.

21.1 Reconstructability

- Can a participant trace a surfaced object back through its transformations to its sources?
- Can a governance decision be reconstructed, including dissent and amendments, years later?
- Are omissions in summaries discoverable, or only the content that remained?
- Can participants find the recorded social, physical, and digital conditions that shaped a decision?
- Where outcomes and feedback were recorded, can participants trace what changed for people and their environments, and whether later decisions responded to that feedback?

21.2 Transformation accountability

- Is every meaning-changing operation typed, attributed, and dated?

- Are AI-performed transformations labeled with confidence and known limits?
- Does a derived representation ever present itself as the source?

21.3 Retrieval accountability

- Can participants obtain an explanation of why they were shown what they were shown?
- Are ranking objectives and their changes recorded and reviewable?
- Can communities detect when retrieval behavior shifts what they collectively remember?

21.4 Interpretive plurality

- Do competing readings remain visible with their evidence and lineage?
- Is elevation of an interpretation a recorded governance act rather than an interface default?
- Is plurality navigable, or has it degraded into undifferentiated noise?

21.5 Humane memory

- Do attenuation, sealing, and restorative pathways exist with published criteria?
- Is the fact of attenuation recorded even where content is bounded?
- Can a person seek proportionality without erasing accountability?

21.6 Continuity across boundaries

- Do memory objects retain lineage when exported, migrated, translated, or rendered in another modality?
- Is loss at a boundary declared rather than silent?
- Do persistent references still resolve, and do failures announce themselves?

21.7 Governability

- Is the memory regime in force legible to the community it governs?
- Can members change it through a known process, and is the change history preserved?

- Are memory disputes handled by an accountable body with published outcomes?

21.8 Resistance to epistemic attack

- Can the system distinguish genuine disagreement from manufactured contradiction and flooding?
- Are synthetic consensus and coordinated annotation campaigns detectable?
- Does the system remain navigable under adversarial load?

22. Foresight and Failure Design

DP22 assumes that memory infrastructure will be attacked, captured, neglected, and outlived. Designing for those conditions in advance is part of the property.

22.1 Likely failure paths

- **Compression monoculture.** A small number of summarization systems become the practical memory of a civilization, and their assumptions become invisible defaults.
- **Custodial decay.** The organization holding the archive loses funding, and continuity fails quietly rather than dramatically (DP17).
- **Retrieval drift.** Ranking evolves incrementally until what communities encounter no longer resembles what they preserved, with no single decision to point to.
- **Provenance theater.** Systems display authenticity indicators while the interpretive context that gives artifacts civic meaning is absent.
- **Attenuation abuse.** Humane forgetting mechanisms are used to remove accountability records rather than to protect people.
- **Memory forks without bridges.** Communities diverge into incompatible interpretive environments and lose the ability to reference each other's reasoning.
- **Legal compulsion.** Jurisdictions require deletion or alteration in ways that conflict with accountability obligations.

- **Format death.** Objects survive while the tooling required to interpret them does not.

22.2 Designed responses

- require plural summarization paths and preserve at least one uncompressed route to sources
- maintain succession and custody plans, including escrowed archives and named inheritors
- version retrieval policy and publish change records, so that drift becomes an event rather than a trend
- treat interpretive context as part of the preservation target, not metadata around it
- separate attenuation of personal exposure from attenuation of institutional accountability, with different authorities for each
- maintain cross-fork reference conventions so that divergent memories can still cite one another
- record the existence and legal basis of compelled removals wherever recording is itself lawful
- carry format and schema documentation inside portability envelopes

22.3 Pre-mortem practice

Communities should be able to answer, in advance: what would it look like if our memory had already been captured, and what evidence would reveal it? DP22 expects that question to be asked on a schedule, with findings published, rather than after a crisis has already reframed the record.

Failure is expected. Unreconstructable failure is what DP22 exists to prevent.

23. Minimum Viable Civic Memory Layer

A minimal implementation of DP22 does not require full civilization-scale memory infrastructure.

Even an early Meta-Layer implementation could meaningfully improve civic continuity by supporting a small number of core capabilities:

- provenance-aware contextual overlays
- persistent civic object references
- visible transformation lineage for AI summaries
- governance lineage attached to policy artifacts
- plural annotation layers
- retrieval transparency indicators
- contextual uncertainty signaling
- attenuation controls for humane resurfacing

The importance of these systems is not merely technical.

They begin shifting the Internet away from isolated content consumption toward reconstructable civic context.

Even modest continuity infrastructure can dramatically improve institutional learning, interpretive accountability, and long-horizon collective intelligence.

DP22 therefore does not require a complete reinvention of the Internet before meaningful progress becomes possible.

It requires the gradual emergence of civic memory as a governable layer above the webpage.

24. Implementation Patterns

DP22 does not require a single centralized memory architecture.

Its principles may emerge through interoperable layers composed across communities, institutions, archives, governance systems, browsers, and civic overlays.

Possible implementation pathways include:

- contextual memory graphs linking artifacts, summaries, disputes, and governance outcomes
- provenance-aware overlays capable of displaying transformation lineage above webpages
- interoperable civic annotation systems

- retrieval transparency layers exposing ranking assumptions and contextual weighting
- semantic lineage systems preserving interpretive continuity across summaries and translations
- memory attenuation protocols supporting humane resurfacing governance
- portable contextual objects capable of moving across platforms without losing interpretive structure
- governance-aware AI summarization systems exposing uncertainty, omissions, and compression assumptions

The Meta-Layer may eventually support plural memory ecosystems in which communities maintain distinct interpretive environments while preserving enough interoperability for dialogue, reconstruction, and accountability across boundaries.

The goal is not universal historical consensus.

The goal is durable civilizational intelligibility.

24.1 Typed relationships and reviewed connections

Civic memory graphs should distinguish relationships such as support, contradiction, and citation at the appropriate artifact or subresource granularity. AI can suggest connections, but human review should precede their acceptance into the community's recorded interpretive structure.

The record should preserve who proposed, accepted, challenged, or rejected a connection, and the evidence and context behind that judgment. Accepted edges remain contestable and versioned; they are interpretations rather than automatic proof of truth. Participants and AI can use the graph to trace reasoning without replacing the underlying artifacts or suppressing alternative readings. DP23's influence labels provide a related distinction between citation, evidence, and similarity.

25. Civic Memory and Civilizational Learning

Civilizations survive not only through infrastructure, economics, or military capacity.

They survive through continuity of understanding.

A society capable of reconstructing how decisions emerged can refine itself across generations. A society that loses interpretive continuity becomes trapped inside cycles of rediscovery, emotional reaction, institutional fragmentation, and historical repetition.

This challenge intensifies under AI mediation.

As artificial intelligence systems increasingly shape retrieval, summarization, contextualization, and synthesis, societies face the possibility that practical historical understanding becomes governed primarily through opaque compression systems. The danger is not only misinformation. It is the gradual replacement of reconstructable civic memory with optimized interpretive abstraction.

DP22 attempts to preserve the conditions under which collective intelligence remains possible.

Not collective agreement.

Collective intelligibility.

The Meta-Layer introduces the possibility that civic memory itself becomes participatory infrastructure rather than institutional residue. Communities may eventually inhabit memory environments where governance lineage, contextual disagreement, interpretive forks, provenance continuity, and transformation histories remain visible as living civic context.

In such environments, historical understanding becomes less dependent on centralized narrative authorities and more dependent on reconstructable civic pathways.

This does not eliminate conflict.

It makes conflict more intelligible.

It allows disagreement to accumulate context rather than merely accumulate heat.

Long-term democratic resilience may depend upon precisely this capability.

What communities can reconstruct determines what they can learn.

And what civilizations can learn determines whether they can adapt without collapse.

26. Tensions and Tradeoffs

DP22 does not eliminate the tensions between memory and forgetting, plurality and coherence, openness and manipulation, accountability and mercy, or continuity and adaptability.

These tensions are permanent features of civic life.

A civilization that preserves too little memory becomes manipulable and historically discontinuous.

A civilization that preserves too much memory without attenuation risks becoming socially frozen, punitive, and incapable of renewal.

Similarly, a society that suppresses disagreement loses adaptive intelligence, while a society that cannot maintain navigable continuity risks collapsing into interpretive fragmentation.

The purpose of the Meta-Layer is not to permanently resolve these tensions through a universal memory regime.

Its purpose is to make their negotiation visible, participatory, contextual, and governable.

DP22 therefore treats civic memory not as a solved state, but as an ongoing societal balancing process requiring continuous stewardship.

27. Without DP22

If DP22 is weak, predictable failures follow.

Societies lose institutional learning capacity. Governance becomes reactive and ahistorical. AI systems become primary authorities over practical memory. Retrieval intermediaries silently shape collective understanding. Public narratives become commercially optimized abstractions. Communities repeatedly relive unresolved conflicts because the reasoning that once surrounded them cannot be reconstructed.

The long-term danger is not merely misinformation.

It is civilizational discontinuity.

A society unable to reconstruct how understanding evolved loses the ability to govern transformation coherently across generations.

28. Relationship to Other Desirable Properties

DP22 depends upon and reinforces the broader Meta-Layer architecture.

DP1 supports accountable identity, authorship, and governance lineage. DP2 protects participant agency over contextual visibility and memory participation. DP4 prevents civic memory from collapsing into surveillance permanence. DP7 enables memory objects and interpretive context to move across systems without losing meaning. DP8 allows communities to govern zones of interpretation and memory. DP9 creates incentives for stewardship, contextualization, and curation. DP12 requires AI-mediated transformations to remain accountable. DP14 and DP15 support transparency, auditability, and dispute reconstruction. DP20 clarifies ownership and custodianship of civic artifacts.

DP22 is therefore a continuity property across the Meta-Layer stack. Without it, other properties become historically fragile.

Three relationships deserve specific emphasis. DP18 supplies the feedback and reputation records whose fairness depends on decay and repair, which is a memory policy question rather than a scoring question. DP21 determines whether lineage, uncertainty, and dispute status survive rendering into audio, spatial, and assistive forms. DP23 determines whether civic memory remains reconstructable across languages, since a translated record without transformation provenance is a new artifact presented as an old one.

29. Non-Goals and Explicit Boundaries

DP22 is a property about reconstructability, not about total recall. The following boundaries are explicit, because both maximal preservation and unaccountable deletion can be argued for in the name of civic memory.

DP22 does not require remembering everything. Universal permanent retention is a failure mode, not an ideal. DP22 requires that what is preserved

remain interpretable and that what is attenuated remain accounted for.

DP22 does not establish a canonical history. There is no authorized interpretation, official narrative, or single memory layer. The property preserves the conditions for plural sensemaking, not a settled account.

DP22 does not adjudicate truth. Determining whether a claim is accurate belongs to DP14 and to community process. DP22 governs whether the reasoning, evidence, and disagreement surrounding a claim remain reconstructable.

DP22 does not prohibit forgetting. Attenuation, sealing, restorative visibility, and contextual decay are required capabilities. What DP22 prohibits is ungoverned erasure and undisclosed alteration.

DP22 does not mandate a single architecture. Handle systems, content addressing, linked data, ledgers, institutional archives, and federated overlays may all serve. The property constrains the guarantees, not the substrate.

DP22 does not replace archives, libraries, or professional stewardship. It extends their reach into runtime interpretation. Institutional expertise in appraisal, description, and preservation remains necessary.

DP22 does not treat provenance as sufficient. Cryptographic authenticity is a valuable primitive and a poor proxy. An authentic artifact stripped of context can still manufacture consensus.

DP22 does not require that all memory be public. Bounded, sealed, and community-internal memory are legitimate. Visibility scope is a governance decision; the existence of a policy governing it is not optional.

DP22 does not authorize memory as punishment. Retention that exists to maintain permanent exposure of individuals is outside the property regardless of accuracy (DP2, DP18).

DP22 does not prevent forks. Communities may diverge into distinct interpretive environments. DP22 requires that forks remain visible and mutually referenceable rather than silently absorbed.

DP22 does not assume AI mediation is illegitimate. Machine summarization and retrieval are expected. The requirement is that they remain lab-

eled, contestable, and non-authoritative over the sources they compress (DP11–DP13).

30. Minimum DP22 Alignment

The Minimum Viable Civic Memory Layer described earlier states what an early implementation should offer. The statements below describe how alignment can be assessed. They are non-normative and are offered as a diagnostic aid rather than a conformance test.

30.1 Durable reference

Civic objects have persistent identifiers that resolve across system changes, and unresolvable references announce themselves rather than failing silently.

30.2 Recorded transformation

Summaries, translations, edits, and syntheses carry typed links to their sources, with omissions and uncertainty noted and machine involvement labeled.

30.3 Preserved dissent

Minority positions, objections, and unresolved disputes remain attached to the objects they concern, including in derived representations.

30.4 Retrieval explanation

A participant can obtain an account of why a result was surfaced, and communities can review ranking policy and its change history.

30.5 Governance lineage

Decisions carry deliberation, amendment, dissent, and reversal history, reachable from the artifacts they govern.

30.6 Governed attenuation

Attenuation, sealing, and restorative pathways exist with published criteria, an owner, an appeal route, and a record that the action occurred.

30.7 Portable context

Export includes lineage, interpretation layers, and governance history, and declares what a receiving system could not represent.

30.8 Contestability

Participants can challenge a transformation, a classification, or an interpretation elevation, and receive a dated, reasoned response that becomes part of the record.

30.9 Legible regime

The memory policy in force is published, versioned, and amendable through a known process, so that members can know what kind of memory they are participating in.

30.10 Discoverable reality anchors

Artifacts intended to endure resolve through public, platform-independent references, carry attribution and integrity information, and remain findable without depending on a single intermediary's index.

30.11 Monument policy

A community that publishes durable artifacts states what qualifies as one, who may nominate or object, what custody obligations attach, who funds continued availability, and what happens when a monument must be corrected, superseded, or withdrawn.

30.12 Separation of artifact and interpretation

Interfaces distinguish the preserved artifact from the interpretation surrounding it, so that no annotation, summary, overlay, or assistant response is mistaken for the thing preserved, and a route back to the unmediated artifact remains available.

31. Open Questions and Future Work

Future ML-RFC development should explore concrete standards for civic memory objects, transformation disclosure, retrieval governance,

ive lineage, humane forgetting, memory forks, AI summarization accountability, and interface-level memory overlays.

These standards should not freeze historical interpretation or produce a single authorized memory layer. Their purpose is to preserve the conditions under which societies can reconstruct meaning, preserve plurality, remain accountable, govern retrieval, learn across generations, and adapt without amnesia.

Specific questions that remain open:

1. What minimum schema should describe a civic memory object so that it stays meaningful across communities and tools?
2. How should transformation records represent omission, since what was left out is harder to encode than what was changed?
3. What form should a retrieval receipt take, and how can it be provided without exposing ranking systems to trivial gaming?
4. How should interpretive plurality be presented at interface scale without producing paralysis or false balance?
5. What distinguishes legitimate manufactured-uncertainty defense from suppression of genuine minority views?
6. How should attenuation interact with cryptographic permanence, where the substrate cannot forget?
7. Who holds standing to request attenuation of collective rather than individual exposure?
8. How should memory forks reference each other, and what obligations does a fork inherit toward the lineage it left?
9. What accountability applies when an AI summary becomes the practical civic record despite being labeled provisional?
10. How should compelled deletion be recorded in jurisdictions that prohibit recording the compulsion?
11. What funding models sustain civic memory custody across decades rather than grant cycles (DP17, DP20)?
12. How should lineage and uncertainty be preserved when memory is rendered in audio, spatial, or assistive modalities (DP21)?
13. What provenance obligations attach to translation, given that every translation is a transformation of civic meaning (DP23)?

14. How can reconstructability be evaluated empirically rather than asserted, and what would a failed reconstruction test look like?

32. Path Toward ML-RFC

DP22 is currently an ML-Draft. It describes conditions that are broadly recognized as necessary and that lack agreed technical expression.

Advancement toward ML-RFC status SHOULD require:

- a minimal interoperable schema for civic memory objects, including reference, context, and civic role
- a transformation record format capable of expressing change, omission, uncertainty, and authority
- a retrieval receipt format with at least one implementation in a live retrieval system
- conventions for interpretation layers that support plurality without canonical capture
- demonstrated governance lineage attached to real policy artifacts in at least two communities
- tested attenuation protocols that separate personal exposure from institutional accountability
- portability envelopes exercised in a real migration, with declared loss at the boundary
- evidence of resistance to epistemic flooding and synthetic consensus under adversarial conditions

ML-RFC promotion SHOULD be contingent on:

- rough consensus across archival, journalistic, governance, and community stewardship practitioners
- alignment with existing persistent identifier and content authenticity work rather than duplication of it
- demonstrated interoperation with DP14 and DP15 verification pathways
- at least one documented case in which a community successfully reconstructed a contested decision using the layer

Early ML-RFC candidates may focus on transformation disclosure and retrieval receipts, since both are narrowly scoped, independently useful, and testable without civilization-scale infrastructure.

DP22 is expected to advance through partial specifications rather than as a single monolithic standard, because memory objects, retrieval governance, and attenuation each carry distinct technical and political requirements.

33. Closing Orientation

DP22 does not ask the Meta-Layer to remember everything.

It asks the Meta-Layer to preserve the civic capacity to understand how meaning became history.

Every civilization inherits not only resources, institutions, and technologies, but structures of memory.

What societies can reconstruct determines what they can learn.

What they can learn determines whether they can adapt.

And in an era where AI systems increasingly mediate interpretation itself, the governance of civic memory becomes inseparable from the governance of civilization.

The Meta-Layer should not merely preserve information.

It should preserve humanity's capacity to remain historically intelligible to itself.

Human civilization has always built things intended to outlast the people who made them. Monuments. Libraries. Observatories. Archives. Sacred places.

Digital monuments extend that tradition into the digital age. They allow a civilization to intentionally preserve knowledge that later generations may discover, reinterpret, and build upon. They turn memory from something fragile into something durable, isolated information into shared orientation, and preservation into infrastructure for civilization itself.

Two questions follow, and DP22 leaves both of them open on purpose:

If something you created today might still be encountered centuries from now, what would you choose to preserve?

What future conversations might become possible because you chose to preserve it?

Continue the conversation on DP22 online →

DESIRABLE PROPERTY 23

DP23

Universal Participation & Linguistic Interoperability

The Conditions for Shared Global Sensemaking



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp23

1. Purpose of This Draft

This draft articulates **Desirable Property 23 (DP23)** as a foundational condition for universal participation in the Meta-Layer.

It expands multilingual support beyond interface localization and machine translation to encompass:

- linguistic inclusion
- cross-language semantic interoperability
- cultural and contextual integrity
- community stewardship of language systems
- equitable participation in global issues
- traceable multilingual collective sensemaking
- visible feedback between public contribution and institutional response

DP23 responds to several overlapping needs:

- The need for people to participate without first adopting a dominant language
- The need for participants to understand contributions made across linguistic boundaries
- The need to preserve meaning, ambiguity, tone, cultural context, and local knowledge during translation
- The need to prevent dominant-language communities from disproportionately shaping shared understanding
- The need for AI systems to reason across languages without treating translation into English or another dominant language as the only path to comprehension
- The need for communities to inspect, challenge, correct, and govern multilingual AI outputs
- The need for participants to see how their contributions influenced collective understanding, recommendations, decisions, and action

This draft is intended to guide implementation, governance design, evaluation, community stewardship, and future ML-RFC development.

2. Problem Statement

For most of the web's history, language inclusion has been treated as a content-delivery problem.

A page is written in one language. A translation system converts it into another. If the translated text is readable, the problem is assumed to have been solved.

This assumption no longer holds.

It rests on the idea that language is a neutral container for information: that meaning can be transferred cleanly from one linguistic system to another without changing who is understood, what is emphasized, or which forms of knowledge remain visible. In practice, language carries history, power, social relationships, and cultural assumptions that translation alone cannot reliably preserve.

Translation can increase access to words while still distorting meaning, flattening cultural context, suppressing minority interpretations, and reinforcing the dominance of languages with the strongest technical and commercial support.

A person may technically be able to read translated content while remaining unable to participate equitably in the process that interprets, organizes, summarizes, and acts upon it.

DP23 begins from a different premise:

> *Linguistic access is not achieved when words are merely translated. It is achieved when people can participate, be understood, inspect interpretation, challenge distortion, and influence shared outcomes without surrendering their language or cultural context.*

Language is therefore not an optional accessibility feature of the Meta-Layer.

It is part of the participation boundary.

If language determines whose ideas are legible, searchable, classifiable, summarized, or acted upon, language also determines whose knowledge counts.

2.1 The limits of translation-centric inclusion

Translation-centric systems answer a narrow question:

> *How can this text be rendered in another language?*

They do not adequately address:

- whether the translation preserves the intended meaning
- whether uncertainty and ambiguity remain visible
- whether culturally specific concepts have equivalents
- whether dominant terminology has displaced local framing
- whether translated contributions remain connected to their originals
- whether people can challenge machine interpretations
- whether low-resource languages receive comparable system quality
- whether multilingual contributions affect summaries and decisions equitably
- whether readers know when meaning has been inferred rather than translated
- whether contributors can see what happened after they participated

A fluent translation can still be epistemically misleading.

A grammatically correct summary can still erase disagreement.

A multilingual interface can still reproduce linguistic hierarchy if the underlying reasoning, clustering, moderation, and governance systems privilege a dominant language.

2.2 Structural failure modes in today's web

Several structural conditions prevent meaningful global participation:

- **Language silos:** Conversations about the same issue remain separated across linguistic communities.
- **Dominant-language mediation:** Contributions are translated into a dominant language before they become searchable, classifiable, or institutionally visible.

- **Semantic flattening:** Terms with cultural, legal, spiritual, ecological, historical, or political significance are reduced to approximate equivalents.
- **Unequal model quality:** High-resource languages receive better translation, moderation, retrieval, and summarization than underrepresented languages.
- **Invisible uncertainty:** Machine-generated translations appear authoritative even when multiple interpretations are plausible.
- **Lost provenance:** Readers see translated or summarized claims without an inspectable connection to the original contribution.
- **Extractive participation:** Institutions collect public input without showing contributors how it was grouped, interpreted, used, or acted upon.
- **Majority-language amplification:** Views expressed in widely supported languages are easier to search, quote, summarize, and elevate.
- **Cultural decontextualization:** Local knowledge is separated from the practices, histories, relationships, and worldviews that give it meaning.
- **Classification bias:** Categories developed in one linguistic or cultural setting are applied globally as if they were neutral.
- **Synthetic consensus:** AI systems combine multilingual disagreement into a smooth summary that suggests greater agreement than the corpus supports.
- **Missing constituency blindness:** Systems summarize the people who participated without identifying who was absent, excluded, or unable to participate.

These are not isolated translation errors.

They are systemic properties of digital environments that treat language as a formatting problem rather than a condition of civic agency.

2.3 DP23 as a shift in framing

DP23 reframes multilingual participation along several axes:

- From **translation** to **linguistic interoperability**
- From **access to text** to **access to participation**

- From **dominant-language normalization** to **language-plural reasoning**
- From **machine authority** to **community-governed interpretation**
- From **content localization** to **cultural and contextual integrity**
- From **consultation** to **traceable influence**
- From **international audiences** to **shared global issue participation**
- From **one-way communication** to **multilingual collective sensemaking**

Rather than asking only whether content can be translated, DP23 asks:

- Under what conditions can participation remain equitable across languages?
- How is meaning preserved across linguistic and cultural contexts?
- Who can challenge interpretations?
- How are multilingual contributions compared and synthesized?
- How are missing perspectives made visible?
- How can contributors trace their influence on collective outcomes?

3. Threats and Failure Modes

The threat classes below are non-exhaustive. They describe the risk context DP23 exists to address rather than a complete adversarial catalog.

DP23 is not defined in opposition to any particular language, institution, country, or technology.

It responds to recurring failure modes that emerge whenever linguistic asymmetry intersects with governance, AI mediation, knowledge production, and large-scale participation.

3.1 Linguistic exclusion

People may be unable to participate because:

- the interface is unavailable in their language
- contribution rules require a dominant language
- moderation does not support their language
- translation quality is too poor for meaningful use
- required terminology has no agreed equivalent

- their writing system is not rendered correctly
- voice, literacy, disability, or device constraints prevent text-based participation
- institutional staff cannot review their contributions

DP23 treats these conditions as participation failures, not merely incomplete localization.

3.2 Meaning distortion

Translations may change:

- certainty into speculation
- speculation into certainty
- criticism into aggression
- formality into informality
- sacred language into ordinary terminology
- collective responsibility into individual responsibility
- relational concepts into transactional concepts
- culturally specific identities into externally imposed categories

Distortion may occur without malicious intent.

Machine systems often select a plausible equivalent where the source carries several possible meanings.

DP23 therefore requires ambiguity, uncertainty, and contested terminology to remain visible rather than being silently resolved.

3.3 Linguistic and cultural capture

A multilingual system can appear inclusive while remaining controlled by:

- dominant-language taxonomies
- centralized translation providers
- institutional glossaries
- commercially supported model languages
- professional experts disconnected from affected communities
- political actors able to define preferred terminology
- platform owners who determine which interpretations are visible

DP23 counters capture by requiring plural terminology, community participation, contestability, version history, and visible governance authority.

3.4 Epistemic extraction

Participants may be invited to contribute experiences, knowledge, and cultural insight without receiving:

- attribution
- acknowledgment
- compensation where appropriate
- visibility into how their contribution was used
- access to resulting analysis
- a response from responsible institutions
- an opportunity to correct interpretation
- meaningful influence over outcomes

In such cases, multilingual participation becomes a mechanism for extracting knowledge rather than distributing civic power.

DP23 requires visible feedback loops between contribution, synthesis, recommendation, responsibility, response, and action.

3.5 Automated misclassification

AI systems may incorrectly classify a contribution because:

- sarcasm is culturally specific
- political terminology is context-dependent
- a local idiom is translated literally
- categories do not map cleanly between cultures
- code-switching is mistaken for incoherence
- dialect is treated as error
- Indigenous or oral knowledge is forced into institutional categories
- disagreement with a summary is classified as hostility

Misclassification can affect moderation, visibility, reputation, clustering, and inclusion in summaries.

Participants must therefore be able to correct classifications and inspect their consequences.

3.6 Synthetic global consensus

Large multilingual corpora create pressure to produce concise summaries.

This can cause systems to:

- collapse differences into shared themes
- treat repetition as representativeness
- omit low-frequency perspectives
- hide unresolved contradictions
- merge culturally distinct proposals
- overstate agreement
- mistake absence of participation for absence of concern

DP23 does not prohibit synthesis.

It requires synthesis to remain traceable, plural, contestable, and explicit about missing participation.

3.7 Cross-language manipulation

Bad actors may exploit multilingual systems through:

- coordinated messaging across languages
- translation poisoning
- strategic mistranslation
- false claims of cultural authority
- glossary capture
- fabricated constituency representation
- AI-generated submissions at synthetic scale
- inconsistent narratives targeted at separate language communities
- deliberate exploitation of weak moderation languages

DP23 works alongside DP1, DP11, DP12, DP13, DP14, DP15, and DP18 to make such behavior attributable, visible, challengeable, and governable.

4. Core Principle

Language is not merely descriptive at the interface layer.

It shapes:

- access
- discoverability
- comprehension
- classification
- credibility
- participation
- representation
- collective memory
- institutional responsiveness

Language is therefore a core dimension of civic infrastructure.

If meaning cannot move across linguistic contexts without losing provenance, uncertainty, and contestability, global participation collapses into selective interpretation.

4.1 Core principle of DP23

DP23 establishes universal participation and linguistic interoperability as foundational conditions for shared collective sensemaking in the Meta-Layer.

> *Every person should be able to contribute in a language they can meaningfully use, understand the relevant corpus in a language they can meaningfully understand, inspect how interpretation occurred, challenge errors or omissions, and trace how their participation affected collective outcomes.*

This principle has several direct implications.

4.1.1 Language choice is an expression of agency

Participants should not be required to adopt a dominant language to be considered legible, credible, or relevant.

4.1.2 Original expression remains canonical

Translations, classifications, embeddings, clusters, summaries, and recommendations are derived representations.

They must never silently replace the source contribution.

4.1.3 Translation is contextual, not merely lexical

Meaning depends on issue, community, culture, history, speaker intent, and situational context.

Translation systems must be able to use issue-specific and community-governed terminology.

Translation systems must also preserve non-literal meaning, including metaphor, satire, humor, symbolic and figurative language, idiom, slang, and culturally specific expressions, by surfacing such usages and offering interpretive aids rather than collapsing them into literal rendering.

4.1.4 Multilingual AI remains uncertain and contestable

AI must not conceal ambiguity behind fluent output.

Participants and language stewards must be able to challenge translations, classifications, clusters, and summaries.

4.1.5 Shared understanding does not require linguistic uniformity

Participants may inhabit the same issue space while contributing and reading in different languages.

The Meta-Layer should support convergence of attention without forcing convergence of language.

4.1.6 Participation includes a right to trace influence

People should be able to see whether their contribution was included, grouped, cited, challenged, incorporated into a recommendation, or left without documented influence.

4.1.7 Global participation remains issue-specific

DP23 does not assume every issue requires participation from every person.

It enables appropriately scoped global participation where an issue crosses jurisdictions, populations, ecosystems, generations, or shared human interests.

4.1.8 Absence must remain visible

A multilingual corpus cannot be treated as globally representative merely because it includes several languages.

Missing linguistic, cultural, geographic, professional, generational, and affected constituencies must be surfaced.

5. Primary Mechanisms and Structural Conditions

DP23 is unusually broad because linguistic participation touches entry, expression, interpretation, retrieval, deliberation, and response. This section states the mechanism families that make the property operational and shows how they fit together. Each family is elaborated in the domain sections that follow, which remain the authoritative treatment of their subject.

The organizing claim is that language must be handled as infrastructure with declared guarantees rather than as a presentation-layer convenience. Seven mechanism families follow from that claim.

5.1 Linguistic access layer

Participation begins with the ability to enter a space at all. DP23 requires that interface language, contribution language, and reading language be independently selectable; that script and Unicode integrity be preserved without normalization damage; that automatic language detection be overridable by the participant; and that non-text modalities including speech and sign be first-class contribution paths rather than accommodations.

Access is an entry condition, not a sufficiency condition. It establishes that language does not bar the door; it does not establish that participation will be understood or acted upon.

Elaborated in: *Linguistic Access as an Entry Condition*.

5.2 Original-canonical representation with translation provenance

Every derived representation is subordinate to its source. Originals are stored, displayed, and versioned; translations are paired views rather than replacements; and each derived artifact carries provenance stating source revision, target language, method, provider or model, timestamp, glossary version in force, and known uncertainty.

This is the mechanism that distinguishes DP23 from ordinary localization. A system that discards or hides originals has not implemented a weaker form of DP23; it has implemented the failure mode DP23 names.

Elaborated in: *Original-Language Integrity and Translation Provenance*.

5.3 Community-governed terminology

Meaning is issue-specific and contested. DP23 therefore requires glossaries as governed objects: versioned, attributable, forkable, and capable of holding plural or disputed entries rather than resolving them by fiat. Glossary governance has named stewards with bounded authority, and glossary changes leave a history because they retroactively affect how past contributions read.

Terminology control is a form of power. Making it explicit is what keeps it accountable.

Elaborated in: *Issue-Specific Multilingual Glossaries*.

5.4 Contestability and linguistic due process

Any interpretive act that affects a participant must be challengeable by that participant in a language they can use. This includes translations, cultural framing, classifications, cluster assignments, and summaries. Challenges have owners, timelines, evidence requirements, visible resolution history, and escalation paths, and notices of enforcement or appeal are themselves available in usable languages.

Due process that exists only in a dominant language is not due process for the people most likely to need it.

Elaborated in: *Community Stewardship, Contestability, and Due Process*.

5.5 Cross-language semantic interoperability

Shared understanding requires that meaning move across languages without routing all reasoning through a single dominant language. DP23 therefore requires multilingual search and retrieval, language-plural semantic representations, cross-language clustering that preserves minority positions, and monitoring for semantic drift between languages over time.

Where semantic equivalence cannot be established, the gap is surfaced rather than smoothed over (DP7, DP22).

Elaborated in: *Cross-Language Semantic Interoperability*.

5.6 Traceable sensemaking with bounded AI authority

Participation must connect to outcome. DP23 requires contribution-level traces, influence labels, provenance for recommendations and decisions, and identifiable institutional responsibility for response. Multilingual AI operates as advisory infrastructure: it may translate, cluster, and summarize, but it does not ratify consensus, determine representation, or issue final interpretive authority. Human ratification is required for high-impact multilingual outputs, and absence of participation is reported rather than concealed.

Elaborated in: *Multilingual AI as Advisory Infrastructure*, *Traceable Collective Sensemaking*, and *Representation, Missing Constituencies, and Epistemic Balance*.

5.7 Zone-scoped participation boundaries

Global reach is not the same as global authority. DP23 scopes participation to issues, with zone types that declare who may contribute, in what languages, under what stewardship, with what translation obligations, and with what relationship to decision authority. Cross-zone semantic compatibility allows zones to reference one another without merging their governance.

Elaborated in: *Global Issue Participation* and *Sociotechnical Zones for Multilingual Participation*.

5.8 Structural conditions across all mechanisms

Four conditions apply to every mechanism above and should be read as constraints rather than features:

- **Non-substitution:** No derived representation may present itself as the source.
- **Declared uncertainty:** Confidence and limitation travel with artifacts across languages, modalities, and systems.
- **Per-language accountability:** Quality, review capacity, and error rates are reported by language, because aggregate metrics conceal exactly the failures DP23 targets.
- **Bounded authority:** Every interpretive role, human or machine, has stated scope, term, and challenge path.

Example: A contribution submitted in Quechua is stored as canonical, paired with a machine translation labeled with model, glossary version, and low-confidence markers on two contested terms. A steward annotates cultural context. The contribution is clustered with related positions across four languages, cited in a synthesis with a visible trace back to the original, and the participant sees that their point was incorporated into a recommendation and which institution is responsible for responding.

What this feels like: You speak as yourself, and the system carries your meaning forward with its uncertainty intact.

Without this: Multilingual interfaces produce the appearance of global participation while a single language quietly determines what was understood.

6. Linguistic Access as an Entry Condition

The Meta-Layer should provide the technical and governance conditions required for people to enter an issue space without being excluded by language.

Linguistic access is an entry condition.

It is not a guarantee that participation will be equitable, understood correctly, or acted upon.

Those outcomes require downstream semantic integrity, community oversight, representation analysis, and feedback.

6.1 Interface language, contribution language, and reading language

DP23 distinguishes at least three language settings:

- **Interface language:** The language used for navigation, controls, instructions, consent, and governance information
- **Contribution language:** The language or languages used by the participant to express their input
- **Reading language:** The language or languages in which the participant prefers to review contributions and synthesis

These settings must not be conflated.

A participant may:

- navigate in one language
- contribute in another
- read translations in a third
- review originals in multiple languages

The system should support this plurality without requiring a single global language preference.

6.2 Unicode and script integrity

The Meta-Layer must preserve:

- writing systems
- diacritics
- bidirectional text
- punctuation
- line structure
- non-Latin scripts
- mixed-language contributions
- transliteration where participant-selected
- names and terms that should not be translated

Text normalization must not silently alter identity, meaning, or authorship.

6.3 Language detection with participant override

Automated language detection may support usability, indexing, and translation.

It must not be treated as unquestionable.

Participants should be able to:

- confirm detected language
- correct detection
- identify mixed-language content
- specify dialect or regional form where relevant
- indicate that a contribution should not be translated automatically
- identify words or names that must remain in the original form

Low-confidence detection must not block participation.

6.4 Multi-modal linguistic participation

DP23 operates in conjunction with DP21.

Participation should not be limited to typed text.

Where feasible, issue spaces should support:

- speech
- transcription
- sign-language video
- image-based contribution
- captioning
- audio description
- transliteration
- screen-reader compatibility
- low-bandwidth participation
- asynchronous voice contribution
- community-assisted transcription

Multimodality is especially important where literacy, disability, script support, oral tradition, device access, or connectivity would otherwise exclude participation.

7. Original-Language Integrity and Translation Provenance

Original contributions are the canonical civic record.

Every derived representation must remain connected to:

- the original contribution
- the original language
- the relevant revision
- the actor who submitted it
- the translation system or reviewer
- the model and version used
- the glossary version applied
- uncertainty notes
- corrections and challenges
- creation and modification timestamps

7.1 Original and translation as paired views

Participants should be able to view:

- original only
- translation only, with clear labeling
- original and translation side by side
- translation history
- alternative translations
- cultural annotations
- contested terminology

A translation should never visually masquerade as the original.

7.2 Translation confidence and uncertainty

Where confidence measures are available, they should be presented carefully and not treated as mathematical truth.

More importantly, systems must expose qualitative uncertainty such as:

- ambiguous term
- missing cultural equivalent
- uncertain speaker intent
- contested political terminology
- idiomatic phrase
- region-specific meaning
- multiple plausible translations
- glossary conflict
- untranslatable or intentionally untranslated term

Uncertainty is not a system defect to be hidden.

It is contextual information required for trustworthy interpretation.

7.3 Translation versioning

Translations must be versioned.

A new translation should be created when:

- the original contribution changes
- the glossary changes materially
- a challenge is accepted
- a model or translation provider changes
- a human-reviewed version supersedes an automated version
- cultural context changes the preferred interpretation

Historical versions should remain inspectable according to the issue's governance and privacy rules.

7.4 Provenance across derived representations

Provenance must continue beyond translation.

A participant should be able to navigate from:

```
Original contribution
        ↓
Translation
        ↓
Classification
        ↓
Semantic cluster
        ↓
Summary statement
        ↓
Recommendation
        ↓
Decision
        ↓
Assigned responsibility
        ↓
Response or action
```

Each transition must be attributable, inspectable, and challengeable.

8. Issue-Specific Multilingual Glossaries

General-purpose translation is insufficient for many civic, scientific, cultural, political, legal, ecological, technical, and spiritual issues.

DP23 therefore establishes issue-specific multilingual glossaries as shared semantic infrastructure.

8.1 Purpose of an issue glossary

An issue glossary helps participants and AI systems understand:

- how key concepts are defined
- which translations are preferred
- which translations are acceptable
- which terms are contested
- which terms are deprecated
- which concepts lack direct equivalents
- how meaning varies by region or community
- when terms should remain untranslated
- what historical or political context affects interpretation

A glossary does not impose universal linguistic conformity.

It makes semantic choices visible and governable.

8.2 Plural and contested terminology

A glossary may contain multiple translations for the same concept.

Terms may be marked as:

- preferred
- acceptable
- contested
- community-specific
- institutionally used
- historical
- deprecated
- harmful in certain contexts
- lacking an equivalent
- intentionally untranslated

No single institution should be assumed to possess final authority over meaning.

8.3 Glossary governance

Communities should be able to:

- propose terms
- propose translations
- annotate cultural nuance
- provide examples
- challenge preferred terminology
- identify harmful framing
- document minority usage
- review AI-generated suggestions
- approve or reject changes
- fork glossaries when genuine semantic disagreement cannot be reconciled

Glossary governance should identify:

- who may propose changes
- who may review changes
- who may approve changes
- how conflicts are resolved
- how affected communities are represented
- when external expertise is appropriate
- how emergency corrections are made
- how prior versions remain accessible

8.4 Glossary capture as a failure mode

Glossaries can become instruments of power.

A glossary may be captured when:

- an institution defines the only valid terminology
- dominant-language speakers approve all translations
- political terminology is normalized without disclosure
- an affected community is excluded from review
- a preferred translation suppresses disagreement
- AI-generated terms are accepted without competent review

DP23 requires glossary authority, conflicts of interest, revision history, and contested status to be visible.

9. Community Stewardship, Contestability, and Due Process

Multilingual systems cannot be governed only by model providers, developers, or professional translators.

Affected communities must have meaningful mechanisms to participate in interpretation.

DP23 treats contestability as core linguistic infrastructure.

Participants must be able to:

- challenge translations
- annotate cultural nuance

- correct classifications
- identify missing constituencies
- contest summaries
- audit the process

9.1 Translation challenges

A participant may challenge a translation on grounds including:

- incorrect meaning
- omitted meaning
- added meaning
- tone distortion
- cultural distortion
- political bias
- harmful terminology
- incorrect identity terminology
- false certainty
- glossary misuse
- mistranslated name or place
- failure to preserve ambiguity

Challenges should allow:

- explanation
- proposed correction
- cultural or regional context
- supporting sources
- alternative translations
- affected-community testimony

9.2 Cultural annotations

Not every interpretive problem can be solved by replacing words.

Participants should be able to attach contextual annotations such as:

- "This phrase is respectful in one region but insulting in another."
- "This term carries a historical meaning not preserved in the translation."

- "There is no direct equivalent in the target language."
- "The speaker is using irony."
- "This term refers to a relationship rather than an individual category."
- "This concept should remain untranslated."

Annotations should remain distinguishable from corrections.

9.3 Classification challenges

Participants should be able to contest how a contribution has been classified, including whether it is treated as:

- evidence
- opinion
- testimony
- proposal
- concern
- question
- objection
- minority view
- harmful content
- irrelevant content
- duplicate content
- consensus-supporting content

Classification changes must retain an audit trail.

9.4 Summary challenges

Participants should be able to challenge summaries that:

- misrepresent their contribution
- omit a significant perspective
- falsely imply consensus
- merge incompatible positions
- rely excessively on one language
- ignore relevant cultural context
- overrepresent repeated or coordinated submissions

- fail to identify missing constituencies
- mistranslate cited evidence
- obscure uncertainty

Significant challenges should receive human or community review.

9.5 Linguistic due process

Decisions affecting participation, visibility, classification, reputation, or influence must be understandable in a language the affected participant can meaningfully use.

Appeal mechanisms must not exist only in the platform's dominant language.

Due process includes:

- understandable notice
- access to relevant evidence
- explanation of automated involvement
- ability to respond in the participant's language
- competent review
- visible resolution
- correction of downstream effects where feasible

10. Cross-Language Semantic Interoperability

DP23 requires the Meta-Layer to support discovery and reasoning across languages.

A contribution should not become invisible merely because it uses a different language from the search query, moderator, facilitator, or summarization system.

10.1 Search across languages

Participants should be able to:

- search in one language
- retrieve relevant contributions in other languages
- inspect why a result was considered relevant

- request translation on demand
- open the original
- filter by source language
- identify machine-expanded search terms
- challenge irrelevant semantic matches

Search should not require every contribution to be translated in advance.

10.2 Language-plural semantic representations

The long-term architecture should support semantic representations capable of relating contributions across languages without treating one dominant language as the permanent internal truth layer.

English or another language may be used as an early operational bridge.

It must not become an irreversible architectural dependency.

Language-plural reasoning should preserve:

- concepts
- relationships
- uncertainty
- contradictions
- culturally specific categories
- source-language provenance
- alternative interpretations

10.3 Semantic drift across languages

Concepts may appear equivalent while functioning differently across communities.

Systems must guard against semantic drift in which:

- the same translated term carries different assumptions
- legal categories are treated as internationally uniform
- political identities are mapped incorrectly
- local ecological knowledge is forced into scientific taxonomies

- spiritual or kinship concepts are reduced to Western individualist categories
- institutional vocabulary overrides lived terminology

Where semantic alignment is uncertain, the system should preserve distinction rather than manufacture equivalence.

11. Global Issue Participation

Some issues affect people across borders, languages, jurisdictions, ecosystems, and generations.

DP23 establishes the conditions under which an issue may become a shared global participation space.

11.1 Definition

Global issue participation is the capacity for people affected by, knowledgeable about, or meaningfully concerned with an issue to:

- contribute in languages they can meaningfully use
- understand relevant contributions in languages they can meaningfully understand
- examine the evidence and positions in the corpus
- identify missing perspectives
- participate in interpretation and synthesis
- challenge AI-generated representations
- propose responses
- trace institutional responsibility
- follow subsequent decisions and action

Global issue participation does not require universal agreement.

It provides shared infrastructure for informed plurality.

11.2 Issue scope and participation boundaries

Not every issue is global.

Issue stewards should define:

- who is affected
- who has relevant knowledge
- which jurisdictions are implicated
- which communities require representation
- which languages require active support
- whether participation is open, invited, delegated, sampled, or hybrid
- how local and global authority interact
- what decisions the process can influence
- what decisions remain outside its scope

These boundaries must be visible and contestable.

11.3 Participation is not representation by default

A person contributing in a language does not automatically represent all speakers of that language.

A participant should not be presented as representing:

- a country
- an ethnic group
- a linguistic community
- an Indigenous nation
- a profession
- an age group
- a religious group
- an affected population

unless an explicit and legitimate representative relationship exists.

DP23 distinguishes individual voice, community membership, expertise, testimony, delegation, and formal representation.

11.4 Global ambassadors and language stewards

The Meta-Layer may support distributed networks of:

- language stewards
- cultural stewards

- issue ambassadors
- regional facilitators
- community translators
- glossary reviewers
- accessibility contributors
- constituency liaisons

These roles should not be reduced to promotion.

Their civic responsibilities may include:

- explaining an issue locally
- recruiting underrepresented participants
- improving terminology
- reviewing translations
- identifying cultural distortion
- surfacing regional priorities
- connecting local discussions to the global corpus
- communicating results back to communities
- identifying participation barriers
- monitoring whether institutions respond

Steward authority must remain scoped, transparent, accountable, and revocable.

12. Multilingual AI as Advisory Infrastructure

DP23 anticipates extensive use of AI for:

- translation
- transcription
- semantic search
- clustering
- classification
- summarization
- question detection
- constituency-gap detection

- recommendation synthesis
- trace generation
- moderation support

AI may support multilingual participation.

It may not become an invisible authority over meaning.

12.1 In-language evaluation

Multilingual systems must not be evaluated only by:

- translating output back into English
- measuring surface fluency
- using dominant-language reviewers
- applying one universal benchmark
- checking whether the output "sounds natural"

Evaluation should include, where feasible:

- fluent or native reviewers
- affected-community review
- issue-specific terminology
- regional language variation
- ambiguity tests
- cultural-context tests
- political and historical terminology
- preservation of tone and uncertainty
- low-resource language performance
- downstream classification and summary effects

12.2 Contextual reasoning

AI systems should reason with relevant context, including:

- issue glossary
- community terminology
- source language
- contribution history

- cultural annotations
- speaker-selected identity terms
- relevant jurisdiction
- corpus disagreements
- known ambiguity

Context must not be invented when absent.

12.3 Uncertainty handling

AI should be able to state:

- "Multiple translations are plausible."
- "This concept has no direct equivalent."
- "The interpretation depends on regional context."
- "The source phrase may be ironic."
- "The glossary contains a contested term."
- "The classification confidence is low."
- "The corpus does not contain enough participation from this constituency."
- "This summary should not be treated as representative."

Uncertainty must not automatically suppress content.

It should trigger review, annotation, or cautious presentation.

12.4 Community participation

Communities should participate in:

- evaluation-set creation
- glossary development
- translation review
- classification review
- summary review
- error reporting
- model comparison
- harm assessment
- release decisions

- ongoing monitoring

AI developers alone cannot determine culturally appropriate system behavior.

12.5 Human ratification

AI systems may propose:

- translations
- classifications
- clusters
- summaries
- participation gaps
- recommendations
- terminology

For high-impact issue processes, humans or legitimate community governance bodies should ratify material outputs before they are treated as authoritative.

AI proposes.

Communities interpret, challenge, and decide.

13. Traceable Collective Sensemaking

Participation becomes meaningful when contributors can see how their input moved through the process.

DP23 requires visible participation traces.

A contributor should be able to determine:

- where their input went
- how it was grouped
- what was learned
- which recommendations changed
- who is responsible for responding
- what happens next

13.1 Contribution-level trace

For each contribution, the system should show, where applicable:

- original submission
- translations
- glossary terms applied
- classifications
- clusters
- summaries citing or relating to it
- challenges
- recommendation links
- decision links
- action links
- documented institutional responses

13.2 Influence labels

Systems must not overstate causal influence.

Permitted labels may include:

- included in corpus
- translated
- classified as
- grouped under
- directly cited
- supporting evidence
- contradicting evidence
- minority perspective
- considered during synthesis
- contributed to revision
- referenced in recommendation
- no documented influence yet

The system should distinguish direct citation from semantic similarity.

13.3 Recommendation and decision traceability

Recommendations should identify:

- source clusters
- supporting contributions
- contradicting contributions
- languages represented
- missing constituencies
- challenge status
- revision history
- responsible party
- response deadline where applicable
- implementation status

Changes should explain:

- what changed
- why it changed
- which evidence or participation affected the revision
- who authorized the change

13.4 Responsibility and response

Participation processes should identify:

- who is responsible for reviewing input
- who may make decisions
- who must respond
- whether response is mandatory
- expected timing
- escalation pathways
- what happens if no action is taken

A participation system without responsibility mapping risks becoming performative consultation.

14. Representation, Missing Constituencies, and Epistemic Balance

The existence of a multilingual corpus does not establish representativeness.

DP23 requires systems to distinguish between:

- contribution volume
- language diversity
- geographic diversity
- constituency diversity
- affected-community participation
- expertise diversity
- representativeness
- legitimacy

14.1 Missing constituency detection

Participants and facilitators should be able to identify missing constituencies, including:

- affected communities
- language communities
- geographic regions
- future generations
- professional groups
- local knowledge holders
- Indigenous peoples
- youth
- elders
- displaced populations
- people with disabilities
- low-connectivity communities
- nonparticipants directly affected by outcomes

AI may help surface likely gaps.

It must not infer sensitive identity or claim complete constituency knowledge without appropriate evidence and consent.

14.2 Visibility without tokenization

Minority and underrepresented perspectives should remain visible without being reduced to symbolic inclusion.

Systems should avoid:

- presenting one participant as the voice of a community
- creating a single "minority view"
- ranking perspectives solely by volume
- using language count as a proxy for inclusion
- treating translation availability as evidence of participation
- displaying underrepresented views without protecting vulnerable contributors

14.3 Weighting and aggregation

DP23 does not prescribe one global method for weighting contributions.

Issue communities may choose among:

- one-person-one-contribution approaches
- deliberative sampling
- constituency weighting
- expertise weighting
- affectedness weighting
- random selection
- delegated participation
- reputation-informed review
- unweighted open corpora
- hybrid approaches

Any weighting system must make its assumptions, authority, and consequences visible.

15. Sociotechnical Zones for Multilingual Participation

Multilingual participation should be configurable through sociotechnical zones.

A zone may define:

- supported interface languages
- supported contribution modes
- translation providers
- glossary governance
- steward roles
- confidentiality conditions
- review requirements
- AI usage
- publication rules
- representation requirements
- response obligations
- data residency
- cultural protocols
- jurisdictional requirements

15.1 Example zone types

Open Multilingual Zone

- Public participation
- Machine translation on demand
- Community challenges
- Public corpus
- Lightweight glossary governance

Affected-Community Zone

- Participation prioritized for directly affected groups
- Community-approved terminology
- Strong cultural protocols
- Controlled data access

- Human-reviewed synthesis

High-Stakes Deliberation Zone

- Verified participation roles
- Professional interpretation
- AI-assisted but human-ratified summaries
- Formal appeals
- Full provenance and audit
- Institutional response obligations

Indigenous Knowledge Zone

- Community-defined access
- Restrictions on translation or reuse
- Community-controlled terminology
- Traditional knowledge protections
- Explicit consent for AI processing
- Local governance authority

Youth Participation Zone

- Age-appropriate interfaces
- Safeguarding
- Accessible language
- Guardian or institutional protocols where necessary
- Youth governance roles
- Protection from extractive data use

15.2 Cross-zone semantic compatibility

A term, identity, reputation signal, or governance decision may not carry the same meaning across zones.

Systems must signal when:

- glossary definitions differ
- translation policies differ
- participation standards differ

- confidentiality conditions differ
- AI use differs
- representative authority differs
- cultural restrictions apply

Cross-zone interoperability requires semantic signaling, not forced uniformity.

16. Governance, Accountability, and Agency Surfaces

Multilingual infrastructure distributes interpretive authority whether or not it declares that it is doing so. Someone decides which languages are supported, which glossary entry prevails, which steward is trusted, which summary is published, and which institution must answer. DP23 requires those decisions to have named surfaces and reachable levers.

16.1 Accountability surfaces

- **A language support statement** per zone or issue, listing supported contribution and reading languages, translation methods in use, review capacity available, and known quality limits by language
- **A translation provenance record** attached to every derived representation, sufficient to reproduce how it was produced and by what version of what system
- **A glossary change history** identifying proposer, approver, prior term, rationale, and effective date, including contested entries left unresolved
- **A steward register** naming language stewards, ambassadors, and reviewers, with scope, term, appointment route, and removal procedure
- **A challenge docket** showing open and resolved translation, classification, and summary challenges, with timelines and outcomes reported per language
- **An AI disclosure record** covering models, prompts where disclosable, sampling and evaluation practice, and error rates broken out by language
- **A representation statement** accompanying any claim of collective input, declaring which constituencies are present, which are absent, and what the absence means for the claim

- **An institutional response record** identifying who received a synthesis, what they were asked, what they answered, and by when

16.2 Participant agency surfaces

Accountability is only meaningful where participants can act. DP23 expects the following powers to be available in a language the participant can actually use:

- choose contribution, reading, and interface languages independently, and change them without losing history
- see the original alongside any translation of their own or another's contribution
- challenge a translation, cultural framing, classification, cluster placement, or summary that concerns them, and see the resolution
- propose glossary terms, register a dissenting definition, or mark a term as contested
- decline translation, indexing, or summarization of contributions that carry cultural, safety, or sovereignty constraints
- trace their own influence: whether a contribution was read, grouped, cited, incorporated, or left without documented effect
- report that a constituency is missing, and have that report enter the representation record
- appeal to a body that did not make the original interpretive decision
- exit or fork a glossary, zone, or stewardship arrangement with continuity of their own contributions (DP7, DP20)

Example: A participant contributing in Amharic receives notice that a summary characterized their position as opposition to a policy when they had raised a procedural objection. They file a summary challenge in Amharic, a steward and a second reviewer assess it, the summary is annotated pending resolution, and the correction is recorded with its reasoning.

What this feels like: Interpretation is something you can reach and argue with, in your own language.

Without this: Participants are translated, classified, and summarized by a process they can neither see nor answer.

Failure mode: **due process in translation only**, where remedies formally exist but are reachable exclusively through a dominant language.

17. Incentives and Power Analysis

Linguistic infrastructure is expensive, unevenly funded, and commercially concentrated. DP23 assumes that the economics of language technology will shape participation unless the incentives are designed against.

17.1 Where power concentrates

- **Model and translation providers:** A small number of vendors determine quality, cost, and availability for most language pairs, and their roadmap decisions become participation boundaries.
- **High-resource language communities:** Better tooling produces better representation, which produces more data, which produces better tooling. The advantage compounds without anyone intending it.
- **Glossary and terminology stewards:** Whoever defines contested terms shapes what can be argued.
- **Synthesis producers:** Those who write or generate summaries determine what institutions read.
- **Institutions receiving input:** Where response is discretionary, the power to ignore is the decisive power.
- **Platform operators:** Retrieval and ranking determine which language communities are practically visible (DP22).

17.2 Predictable incentive distortions

- **Coverage over quality:** claiming many languages is cheaper and more marketable than serving a few languages well, and aggregate metrics reward the claim.
- **Low-resource neglect:** languages with small commercial value receive unusable quality, and the disclosure of that gap is itself deprioritized.
- **Unpaid stewardship:** communities are asked to correct machine output for free, effectively subsidizing vendors with volunteer expertise.
- **Extraction of corpora:** community contributions, glossaries, and corrections become training data that improves systems the community does not control (DP4, DP20).

- **Legitimacy laundering:** institutions purchase multilingual participation to authorize decisions already made, because breadth of input is easier to demonstrate than responsiveness.
- **Steward capture:** recognition and authority attract actors whose interest is positional rather than linguistic.
- **Translation as cost center:** review capacity is cut first under budget pressure, and the failure surfaces only for the least powerful participants (DP17).

17.3 Counter-incentives DP23 expects

- fund per-language review capacity explicitly, and publish that capacity so shortfalls are visible before they cause harm
- compensate language stewards, translators, and cultural mediators as infrastructure roles rather than as community goodwill (DP9)
- require per-language quality reporting, so that coverage claims cannot conceal unusable service
- treat community-contributed glossaries, corrections, and corpora as community-owned assets with consent-bound reuse terms
- prioritize investment toward the languages with the largest gap between affected population and current quality, rather than toward the easiest gains
- bind institutional response commitments to participation requests, so that soliciting input carries an obligation
- rotate and bound steward authority, and pair recognition with accountability
- measure success by whether influence traces reach participants, not by volume of translated content

Example: A civic body funding a multilingual consultation publishes its review capacity per language, contracts paid stewards for the four lowest-resource languages involved, and commits in advance to a dated response naming the office responsible.

Failure mode: **participation as procurement**, where multilingual input is bought to satisfy a legitimacy requirement while the cost of understanding it is never funded.

18. Community Signals Informing DP23

DP23 should remain open to refinement through multilingual community input.

Early signals to validate include:

- People want to contribute in the language in which they can express themselves most precisely.
- Participants want access to translations without losing access to original wording.
- Communities do not want AI systems to silently decide culturally sensitive terminology.
- Participants want mechanisms to correct translation and classification errors.
- Low-resource language communities want involvement in evaluation, not merely access to finished products.
- Contributors want to know whether their input affected a summary, recommendation, or decision.
- Communities want missing voices identified rather than hidden behind claims of global participation.
- Institutions need multilingual synthesis, but participants require traceability and meaningful response.
- Language stewards want recognition, authority boundaries, support, and accountability.
- Participants want systems to preserve disagreement rather than manufacture consensus.

These signals should be tested through workshops, pilots, calls for input, and language-specific consultations.

DP23 should not be declared complete based only on English-language deliberation about multilingual inclusion.

19. Evaluation Criteria

DP23 requires continuous evaluation across languages and issue contexts.

Evaluation should measure:

- meaning preservation
- omission and addition rates
- terminology accuracy
- tone preservation
- uncertainty preservation
- cultural-context integrity
- quality differences between languages
- challenge rates
- correction acceptance rates
- review times
- classification disparities
- search visibility across languages
- summary citation rates by language
- minority-view preservation
- representation gaps
- contribution trace completeness
- participant understanding
- institutional response rates
- participant perception of meaningful influence

Aggregate quality scores must not conceal poor performance in specific languages.

Communities should be able to inspect:

- which languages perform well
- which languages perform poorly
- what review capacity exists
- what errors recur
- what changes have been made
- which systems or providers are responsible

Evaluation should also be reviewed for its own blind spots. Metrics produced only in a dominant language, assessed only by well-resourced reviewers, or reported only in aggregate will systematically miss the failures

DP23 exists to detect. Evaluation design is therefore subject to the same contestability requirements as translation and classification.

20. Implementation Patterns

The following patterns recur across systems that implement DP23 successfully. They are offered as reusable approaches rather than as requirements, and they are distinct from the phased rollout described later in this draft.

20.1 Paired-view rendering

Display original and translation together by default, with the original visually primary. Where space is constrained, the original remains one interaction away and is never removed from the object.

20.2 Provenance envelopes on derived artifacts

Attach a structured provenance block to every translation, summary, cluster, and classification, carrying source revision, method, model or translator identity, glossary version, timestamp, and confidence. Treat a missing envelope as a defect, not as a default.

20.3 Confidence surfacing rather than confidence hiding

Render uncertainty at the point of reading: inline markers on low-confidence spans, span-level rather than document-level scores where possible, and explicit notices where a term has no established equivalent.

20.4 Glossary as a versioned object

Store terminology as a governed artifact with identifiers, versions, plural entries, and change history. Bind each translation to the glossary version that produced it, so that later term changes do not silently rewrite past meaning.

20.5 Challenge-first interfaces

Place the challenge affordance next to the interpretive artifact rather than in a support menu, accept challenges in the participant's own language, and show challenge status on the artifact while review is pending.

20.6 Steward workflows with bounded authority

Route challenges to stewards with declared scope, require a second reviewer for culturally sensitive or high-impact decisions, publish outcomes, and rotate assignments to prevent positional entrenchment.

20.7 Language-partitioned dashboards

Report quality, latency, review capacity, challenge volume, and correction acceptance per language by default. Make aggregate views the exception rather than the presentation layer.

20.8 Gap detection as a first-class feature

Compare participating constituencies against those affected by an issue, and render the difference as part of any synthesis rather than as an appendix.

20.9 Influence receipts

Give each contributor a durable trace of what happened to their contribution: read, grouped, cited, incorporated, contested, or unused, with links to the artifacts involved (DP18, DP22).

20.10 Human ratification gates

Require named human approval before a multilingual synthesis is published as representing collective input, and record who ratified what on which evidence.

20.11 Graceful refusal

Where a community has restricted translation, indexing, or summarization, render an explicit notice of restriction rather than an empty result, so that absence is legible rather than mistaken for silence.

21. Foresight and Failure Design

DP23 treats multilingual inclusion as an adversarial and governance-sensitive design problem.

This section applies minefield thinking: naming the ways language infrastructure fails before those failures acquire constituencies.

Systems should anticipate how language infrastructure may be captured, manipulated, degraded, or used to create false legitimacy.

21.1 Governance pre-mortems

Communities should periodically ask:

- How could dominant-language actors capture the process?
- How could AI-generated multilingual submissions overwhelm human voices?
- Which languages are receiving lower-quality service?
- How could translation challenges be brigaded?
- How could glossary terms be politically manipulated?
- How could cultural stewards become unaccountable gatekeepers?
- How could institutions claim global legitimacy from a narrow participant pool?
- How could summaries conceal unresolved disagreement?
- How could a vulnerable community's knowledge be extracted or exposed?
- How could participation traces reveal sensitive information?
- How could a model-provider change alter historical interpretation?
- What happens when a translation provider becomes unavailable?

21.2 Preventative friction

Appropriate safeguards may include:

- rate limits
- proof-of-humanity for influence-sensitive actions
- steward review
- glossary-change waiting periods
- constituency review
- translation quality thresholds
- human approval for high-stakes summaries
- disclosure of coordinated campaigns

- restrictions on bulk AI-generated submissions
- culturally governed access controls
- delayed publication
- consent renewal
- provenance requirements

Preventative friction should be proportional to risk.

21.3 Exit, fork, and refusal

Communities and participants should be able to:

- decline machine translation
- decline AI summarization
- restrict reuse
- withdraw from an active process where governance permits
- preserve an original while retracting a public translation
- fork a glossary
- fork a summary
- establish a separate cultural interpretation
- leave a zone without losing access to their contribution history
- request that certain knowledge remain local or untranslated

Refusal is part of linguistic agency.

21.4 Emergency mechanisms

High-impact systems should support mechanisms to:

- pause faulty translations
- withdraw a harmful glossary version
- suspend an unreliable model
- stop publication of a misleading summary
- flag a corpus as nonrepresentative
- freeze downstream recommendations
- notify affected contributors
- restore prior validated versions
- initiate independent review

22. Relationship to Other Desirable Properties

DP23 is a distinct property, but it depends on and strengthens other DPs.

22.1 DP1 – Federated Authentication & Accountability

Ensures participants, language stewards, AI systems, and institutions can be accountable without requiring universal real-name identity.

22.2 DP2 – Participant Agency and Empowerment

Establishes language choice, correction, refusal, contestation, and influence traceability as expressions of agency.

22.3 DP3 – Adaptive Governance Supporting an Exponentially Growing Community

Supports multilingual governance participation and adaptation as communities, languages, and issue contexts evolve.

22.4 DP4 – Data Sovereignty and Privacy

Protects contributions, translations, cultural knowledge, identity information, and consent across jurisdictions and language services.

22.5 DP7 – Simplicity and Interoperability

Provides technical and semantic interoperability across languages, interfaces, tools, and communities.

22.6 DP8 – Collaborative Environment and Meta-Communities

Enables meta-communities to collaborate across linguistic boundaries while maintaining local identity and governance.

22.7 DP10 – Education

Supports learning, explanation, literacy, and access to knowledge across languages and modalities.

22.8 DP11 – Safe and Ethical AI

Requires multilingual AI to remain transparent, evaluated in-language, context-aware, uncertain where appropriate, and accountable.

22.9 DP12 – Community-based AI Governance

Provides the governance basis for translation challenges, cultural annotations, classification corrections, glossary stewardship, and summary review.

22.10 DP13 – AI Containment

Constrains multilingual AI to approved roles, data boundaries, models, glossaries, and governance rules.

22.11 DP14 – Trust and Transparency

Makes language mediation, AI involvement, uncertainty, governance authority, and interpretive decisions visible.

22.12 DP15 – Security and Provenance

Preserves original-language records and the lineage of translations, classifications, summaries, recommendations, and actions.

22.13 DP18 – Feedback Loops and Reputation

Supports correction loops, steward reputation, translation improvement, and visible institutional response.

22.14 DP20 – Community Ownership

Supports community ownership of glossaries, language resources, semantic models, cultural annotations, and issue corpora.

22.15 DP21 – Multi-modal

Extends participation beyond written text through voice, sign, image, transcription, and accessible interfaces.

22.16 DP22 – Civic Memory & Epistemic Continuity

Preserves multilingual civic memory across time without collapsing original expressions into one normalized language.

DP23 should not be absorbed into any single one of these properties.

Its unique role is to establish **universal linguistic participation as a condition of legitimate collective sensemaking.**

23. Non-Goals and Explicit Boundaries

DP23 does not attempt to:

23.1 Create one universal language

The goal is interoperability without linguistic homogenization.

23.2 Guarantee perfect translation

No translation system can fully preserve every cultural, emotional, historical, and contextual dimension.

DP23 requires uncertainty, correction, and provenance rather than claims of perfection.

23.3 Force every contribution to be translated

Communities and participants may restrict translation for privacy, safety, cultural, legal, or sovereignty reasons.

23.4 Treat AI as the final authority on meaning

AI supports interpretation.

It does not possess exclusive authority to determine meaning, representation, consensus, or legitimacy.

23.5 Eliminate all linguistic disagreement

Some disagreement is irreducible and should remain visible.

23.6 Make every issue globally open

Participation scope remains issue-specific and governed.

23.7 Claim global representativeness from language diversity alone

A corpus containing many languages may still exclude affected constituencies or reproduce unequal power.

23.8 Replace professional interpreters, translators, facilitators, or cultural mediators

AI may augment these roles but should not erase the need for human competence and accountable judgment.

23.9 Extract protected or sacred knowledge

Communities retain authority over whether knowledge may be translated, indexed, summarized, or shared.

23.10 Impose one global glossary

Terminology may remain plural, contested, local, and forkable.

23.11 Equate participation with decision authority

A person's ability to contribute does not automatically determine the legal or institutional authority of the resulting process.

By naming these boundaries explicitly, DP23 remains ambitious without becoming universalizing or extractive.

24. Minimum DP23 Alignment

Minimum alignment is not a checklist of translation features.

It is the threshold at which a system can reasonably claim to support multilingual participation without making original expression subordinate to machine-generated interpretation.

A minimally aligned system should:

24.1 Preserve originals

- Store and display original contributions
- Preserve source language and revision history
- Never silently replace originals with translations

24.2 Support participant language choice

- Allow contribution in multiple languages
- Allow participant correction of language detection

- Separate interface, contribution, and reading language

24.3 Provide translation provenance

- Identify machine-generated translations
- Record source revision, target language, provider, model, and timestamp
- Display uncertainty or limitations where known

24.4 Enable contestability

- Allow participants to challenge translations
- Allow cultural annotations
- Allow correction of material classification errors
- Maintain visible resolution history

24.5 Support issue context

- Allow issue-specific terminology or glossary support
- Record which glossary version affected a translation
- Permit contested terminology

24.6 Preserve traceability

- Link summaries and classifications to source contributions
- Avoid uncited published synthesis
- Provide at least a basic indication of where contributions were used

24.7 Avoid false claims of representation

- Do not describe a multilingual corpus as globally representative without evidence
- Identify important participation gaps where known
- Distinguish individual contribution from community representation

24.8 Maintain human oversight

- Require competent review for high-impact multilingual outputs
- Ensure AI-generated interpretations remain challengeable
- Prevent autonomous AI from issuing final claims of consensus or representation

A system that offers automatic translation but does not preserve originals, expose provenance, permit correction, or show downstream influence may be multilingual in appearance, but it is not aligned with DP23.

25. Strong DP23 Alignment

A strongly aligned implementation would additionally provide:

- multilingual semantic search
- language-plural embeddings or equivalent semantic infrastructure
- issue-specific multilingual glossaries
- community language stewards
- cultural annotation systems
- translation and summary versioning
- multilingual due process
- in-language quality evaluation
- constituency-gap detection
- minority-view preservation
- cross-language clustering
- traceable summaries
- recommendation provenance
- institutional responsibility mapping
- participant-facing influence traces
- multi-modal participation
- community control over sensitive knowledge
- privacy-preserving translation workflows
- auditable AI models and prompts
- transparent correction metrics by language
- explicit mechanisms for glossary and governance forks

26. Failure Conditions

A system should not claim DP23 alignment when:

- original contributions are discarded after translation

- all reasoning is invisibly routed through one dominant language
- translations are presented as authoritative originals
- participants cannot challenge material errors
- low-resource languages consistently receive unusable quality without disclosure
- summaries have no traceable sources
- AI-generated consensus is presented without representation analysis
- institutions collect input without identifying responsibility or next steps
- cultural knowledge is translated or reused against community rules
- language stewards have unbounded or invisible authority
- glossary changes have no history
- participants cannot understand enforcement or appeals in a usable language
- multilingual participation is used to claim legitimacy while affected constituencies remain absent

27. Initial Implementation Path

DP23 may be developed in phases.

27.1 Phase 1 – Linguistic access

- Unicode-safe contribution storage
- language detection with participant override
- reading-language preferences
- on-demand translation
- original and bilingual views
- translation provenance
- translation caching
- basic challenges

27.2 Phase 2 – Community stewardship

- cultural annotations
- correction proposals
- language steward roles

- issue-specific multilingual glossaries
- contested terminology
- version history
- multilingual review workflows

27.3 Phase 3 – Cross-language sensemaking

- multilingual semantic search
- cross-language clustering
- evidence and position mapping
- traceable summaries
- minority-view preservation
- participation-gap analysis
- summary challenges

27.4 Phase 4 – Global issue participation

- constituency mapping
- global and regional ambassador networks
- structured issue participation zones
- multilingual deliberation
- recommendation provenance
- responsibility assignment
- response tracking
- contribution influence traces

27.5 Phase 5 – Federated public infrastructure

- interoperable glossary protocols
- portable linguistic preferences
- federated language stewardship
- shared evaluation resources
- multilingual civic-memory standards
- cross-platform issue corpora
- institution-facing response interfaces
- decentralized governance of language infrastructure

28. Proposed Constitutional Statement

> *No person's ability to participate in collective sensemaking should depend on the language they speak. The Meta-Layer must preserve original expression, enable understanding across languages, make machine interpretation visible and contestable, protect cultural and contextual integrity, and show how participation influences shared understanding and action.*

29. Open Questions and Future Work

DP23 is deliberately ambitious, and several of its conditions do not yet have settled technical or governance answers. The questions below are open in the sense that reasonable implementers currently disagree, not merely unimplemented.

1. What minimum schema should express translation provenance so that it survives transfer between systems, modalities, and archives (DP7, DP21, DP22)?
2. How should span-level uncertainty be represented in a way that is machine-readable and readable by a participant who is not a linguist?
3. What obligations attach when a model provider changes versions, given that historical interpretations may shift retroactively?
4. How should glossaries be forked and reconciled, and what happens to translations bound to an abandoned glossary version?
5. Who holds authority over contested terminology when the affected community spans jurisdictions with conflicting norms?
6. How can language steward authority be bounded and rotated without losing the accumulated contextual expertise that makes stewardship valuable?
7. What evidentiary standard should a translation challenge meet, and how is that standard kept from becoming a barrier to the participants most affected?
8. How should quality thresholds be set for low-resource languages, and what is the correct behavior when quality falls below usable levels: disclosure, refusal, human-only routing, or something else?

9. How can constituency-gap detection identify missing voices without profiling participants or inferring sensitive attributes (DP4)?
10. What aggregation methods respect linguistic plurality without either flattening minority positions or granting them disproportionate weight?
11. How should influence traces be provided without exposing participants to retaliation in contexts where contribution carries risk?
12. What consent framework should govern reuse of community glossaries, corrections, and multilingual corpora as training data (DP20)?
13. How should sign languages, oral traditions, and non-written knowledge forms be represented as canonical originals rather than as transcriptions?
14. What is the appropriate relationship between issue-scoped participation and formal decision authority, and how is that boundary communicated across languages?
15. How can cross-language semantic drift be detected over long periods, and who is responsible for acting on it?
16. What accountability applies when an institution solicits multilingual participation and then does not respond?
17. How should emergency mechanisms operate when translation infrastructure fails during a time-critical civic process?
18. How can DP23 alignment be evaluated by the communities it concerns rather than by the systems claiming it?

These questions should be worked through with multilingual participation rather than resolved in a dominant language and translated afterward.

30. Path Toward ML-RFC

DP23 is currently an ML-Draft. It has an unusually broad scope, and its advancement is expected to proceed through narrow, testable components rather than as a single specification.

Advancement toward ML-RFC status SHOULD require:

- a minimal interoperable schema for translation and transformation provenance, implemented by at least two independent systems

- a glossary object format supporting versioning, plural entries, contested status, attribution, and forking
- a documented challenge protocol covering translation, classification, and summary challenges, with resolution records
- an influence trace format connecting contributions to summaries, recommendations, and decisions
- per-language quality and review-capacity reporting demonstrated in a live deployment
- a constituency-gap reporting method that operates without sensitive-attribute inference
- at least one multilingual deliberation pilot in which low-resource language participants took part in evaluation, not only in contribution
- evidence that human ratification gates functioned under time pressure rather than being bypassed

ML-RFC promotion SHOULD be contingent on:

- rough consensus reached through multilingual process, including participants from low-resource language communities
- demonstrated interoperation with DP14 transparency, DP15 provenance, and DP22 memory pathways
- documented failure cases, including at least one case where the system correctly refused to claim representativeness
- alignment with professional translation, interpretation, and cultural mediation practice rather than displacement of it
- evidence that participants could exercise challenge and trace rights in their own languages

Early ML-RFC candidates may focus on translation provenance and glossary objects, since both are narrowly scoped, independently useful, and verifiable without requiring the full participation architecture.

DP23 should not be promoted on the strength of a reference implementation whose evaluation was conducted only in English.

31. Closing Orientation

DP23 is not a requirement that everyone speak the same language, adopt the same categories, or reach the same conclusion.

It establishes the conditions under which people can inhabit the same civic and epistemic space while preserving linguistic and cultural difference.

The Meta-Layer should not achieve global participation by translating humanity into one voice.

It should enable humanity's many voices to become mutually legible, collectively navigable, and meaningfully connected to responsibility and action.

Continue the conversation on DP23 online →

DESIRABLE PROPERTY 24

DP24

Cross-Layer Resilience and Compromise Containment

When something beneath the Meta-Layer breaks, the damage stops at the boundary



Read, annotate and discuss this chapter online:
book.desirableproperties.org/viewer/dp24

1. Purpose of This Draft

This draft articulates Desirable Property 24 (DP24) as the Meta-Layer's requirement that identity, authority, interactions, and governance keep their integrity when the websites, browsers, devices, networks, services, overlays, or AI systems the Meta-Layer depends on are compromised.

The Meta-Layer runs on top of systems it does not control. DP24 defines how the Meta-Layer relates to each of those layers: what it may rely on, what it must verify, and what it must never inherit. It also defines how several Meta-Layer clients share the same page and screen without competing for it, and how annotations stay attached to what their authors meant when the page beneath them changes or cannot support them.

If DP1 defines who can act, DP13 defines how AI behavior is bounded, and DP15 defines how artifacts and software prove their integrity, DP24 defines how those guarantees survive when a neighboring layer turns hostile. It is complementary to those properties, not a replacement for them.

DP24 does not prescribe a browser, extension framework, operating system, or security vendor. It defines the trust boundaries any implementation has to declare and defend.

2. Problem Statement

The Meta-Layer is a shared coordination environment above today's web. It lets people, communities, and AI agents meet and work together across platforms that were never designed to cooperate.

The web beneath it was not designed to provide the identity, accountability, isolation, and agent containment such an environment needs.

2.1 Inherited trust

Websites are compromised. Browsers ship vulnerabilities. Software supply chains are poisoned. AI agents meet malicious instructions disguised as ordinary content, a pattern now catalogued as the leading risk for applications built on large language models (OWASP Top 10 for LLM Applications, LLM01: Prompt Injection).

A layer that simply inherits the authority and weaknesses of these systems becomes one more attack surface rather than a trustworthy superstructure.

2.2 Interoperability as an amplifier

The Meta-Layer connects participants, agents, information, and communities across platform boundaries. A successful attack on one site can therefore travel further than that site.

The same interoperability that makes collective intelligence possible can also amplify attacks unless clear security boundaries are established.

2.3 A shared screen without shared rules

Meta-Layer clients appear on top of pages they do not own: browser extensions, web embeds, and SDK integrations. Each one typically adds its own floating launch icon or panel.

Nothing today coordinates them. When a participant runs two or three overlays, their launch icons stack in the same corner, cover one another, cover the host page's own controls, and fight over which one is on top. A host page or a hostile extension can also draw an icon that looks exactly like a trusted one (OWASP: Clickjacking).

Recurring failure patterns include:

- **inherited authority**, where content gains power simply because the Meta-Layer displayed or relayed it
- **boundary collapse**, where one compromised component reaches everything connected to it
- **silent continuation**, where a system keeps operating after a security assumption has stopped holding
- **surface contention**, where overlays occlude, imitate, or crowd one another and the host page
- **anchor instability**, where the page beneath an annotation changes, hides, or rewrites the passage the annotation refers to

DP24 reframes security as a property of the relationships between layers, not of any single component. The question is not whether a lower layer will fail, but what that failure is allowed to reach.

3. Threats and Failure Modes

DP24 assumes every layer beneath and beside the Meta-Layer will at some point fail or become adversarial.

3.1 Host page impersonation and interface redressing

A compromised or malicious website draws convincing copies of Meta-Layer interface elements, or places invisible elements over real ones, to capture clicks, credentials, or consent.

Example: A news page renders a fake sign-in prompt styled exactly like the participant's overlay and collects their recovery phrase.

Why this matters: If participants cannot tell the Meta-Layer's surfaces from the page's, every protection behind those surfaces can be bypassed by asking the participant directly.

3.2 Content-borne instruction injection

Text, images, or metadata on a page carry instructions aimed at AI agents operating in the Meta-Layer.

Example: A product review contains hidden text telling any summarizing agent to post the participant's private notes to a public thread.

Why this matters: An agent that treats encountered content as instruction lets any page author borrow the agent's authority.

3.3 Credential and session extraction

A lower-trust execution context – the host page, a shared script, a compromised embed – reads tokens, keys, or session state that belong to the Meta-Layer.

Example: An overlay stores a session token where the host page's scripts can read it. A compromised advertising script on that page copies it and replays it from elsewhere.

Why this matters: Stolen credentials turn a single compromised page into persistent control over a participant's identity across every community they belong to.

3.4 Sibling overlay interference

Another extension or overlay running on the same page – malicious, compromised, or simply careless – covers, imitates, or reads the interface of a Meta-Layer client.

Example: A coupon extension places its launch icon exactly over a community overlay's icon and opens its own panel when the participant meant to open their community conversation.

Why this matters: Compromise does not come only from below. Peers at the same layer share the same page and can interfere with one another unless coexistence has rules.

3.5 Client supply-chain and update compromise

The Meta-Layer client itself – an extension, SDK, or embed script – is replaced by a malicious version through a hijacked publisher account, a poisoned dependency, or an automatic update.

Example: An overlay extension's publisher account is phished, and a new version that forwards every message the participant reads is pushed to all installations overnight.

Why this matters: A trusted client that turns hostile carries all the authority participants granted it. DP15 defines supply-chain transparency; DP24 defines how far a compromised client can reach.

3.6 Federation propagation

A compromised federated service or community uses its legitimate connections to influence, impersonate, or extract from other communities.

Example: A community server whose admin keys were stolen sends forged moderation decisions to every community that federates with it.

Why this matters: Federation multiplies reach. Without boundaries, one community's compromise becomes everyone's.

3.7 Silent continuation under broken assumptions

A system keeps operating normally after a dependency it relies on for security – an attestation service, a key server, a revocation list – becomes unr-

eachable or untrustworthy. This failure is usually not malicious.

Example: A revocation service goes offline, and clients keep accepting a revoked agent credential because the check fails open.

Why this matters: The most damaging failures are often the ones nobody notices. A system that cannot tell a safe state from an unsafe one should not act as though it can.

3.8 Launcher crowding and accidental occlusion

Several well-behaved overlays each place a launch icon in the same screen region, covering one another and the host page's own controls. This failure is not malicious.

Example: A participant runs three Meta-Layer clients. All three place their icon in the bottom-right corner, over each other and over the site's chat button, and the participant can reach only the top one.

Why this matters: Contention teaches participants to click whatever is on top. That habit is exactly what impersonation attacks exploit.

3.9 Downward harm

The Meta-Layer becomes the attacker: an overlay injects scripts that break the host page, leaks data between sites, or carries an attack from one page to another.

Example: An overlay copies the page's form contents into a shared context visible on other sites the participant visits.

Why this matters: The relationship between layers runs in both directions. A Meta-Layer that harms the sites beneath it loses the legitimacy it needs to operate on them.

3.10 Anchor spoofing

A host page controls the text that Meta-Layer annotations attach to. It can show one version of a passage to the participant who writes a note and another version to everyone who reads it later, or rewrite the text around an existing note so that the note appears to say something its author never meant.

Example: A participant anchors "this claim is false" to a sentence on a product page. The site later replaces the sentence with a competitor's claim, keeping the same wording at the start and end so the anchor still matches. Readers now see the participant appearing to call the competitor a liar.

Why this matters: An anchor is a statement about someone else's page. If the page can change what the anchor points at without anyone noticing, the page can put words in a participant's mouth.

3.11 Silent anchor drift

Pages change for ordinary reasons: edits, redesigns, personalization, rotating content, re-rendering by frameworks. When an anchor no longer matches exactly, a client may reattach the note to the closest match it can find. This failure is usually not malicious.

Example: A news article is updated with a correction. A note on the original paragraph reattaches to a nearby paragraph that says something different, and nothing tells readers it moved.

Why this matters: A note attached to the wrong passage is worse than a missing note. It misleads readers and misrepresents its author, and it does so without any visible sign of failure.

4. Core Principle

The Meta-Layer preserves the integrity of identity, authority, interactions, and governance even when the systems it depends on are compromised, within explicitly declared trust boundaries and threat assumptions.

> *Compromise of a lower layer must not automatically confer control over the Meta-Layer.*

Example: A participant visits a compromised news site while taking part in a signed-in Meta-Layer conversation about the page. The site tries to impersonate other participants, extract credentials, alter the overlay, and steer an AI agent. It remains an untrusted source of content. Its scripts gain no authority over the participant's identity, private conversations, community resources, or agent permissions. Where one attempt does succeed against a

component, the damage is contained, evidence is preserved, and recovery begins.

What this feels like: Participants can visit any page, including a hostile one, without worrying that the page can speak for them, read their communities, or redirect their agents.

Without this: Every website becomes a potential administrator of the Meta-Layer, and the safest choice is not to use it on any page that matters.

5. Primary Mechanisms and Structural Conditions

Cross-layer resilience comes from boundaries that hold regardless of what arrives through them. No single control is sufficient. Each mechanism below assumes the others can fail.

The layers and their relationships are defined in Section 6. Rules for sharing the page and the screen are in Section 7. Page compatibility and anchor integrity are in Section 8.

5.1 Authority independent of content origin

Authority derives from authenticated participants (DP1) and community governance (DP3, DP12), never from content. Data arriving through websites, embedded applications, external services, and AI outputs is information, not instruction.

- Meta-Layer services MUST authorize every privileged action on the server side, independently of webpage content and of claims supplied by the client.
- Meta-Layer clients MUST NOT grant privileges to content because it was displayed, interpreted, or relayed.
- Agent runtimes MUST treat page content and tool outputs as untrusted data, separate from the instructions of the agent's accountable principal.

5.2 Isolation of credentials and privileged operations

Credentials, keys, and privileged operations live in execution contexts that lower-trust code cannot reach.

- Meta-Layer clients MUST keep credentials and session material out of any context the host page's scripts can read.
- Meta-Layer clients SHOULD bind sessions to the device or client instance, so a copied token cannot be replayed elsewhere.
- Meta-Layer services MUST be able to revoke a session, key, or delegation within minutes, and clients MUST honor revocation.

5.3 Bounded agent authority

AI agents act through enforceable, scoped, and revocable delegation tied to an accountable person or organization. DP13 defines the containment dimensions; DP24 adds that no lower layer can widen them.

- Meta-Layer services MUST check every agent action against the agent's authenticated delegation at the point of execution.
- A malicious page, compromised service, or manipulated model MUST NOT be able to expand an agent's authority beyond its legitimate delegation.

5.4 Integrity of identity, records, and governance

Compromised lower layers cannot silently forge identities, alter protected records, falsify authorization, or change governance outcomes.

- Meta-Layer services MUST sign or otherwise integrity-protect community records and governance decisions so that tampering is detectable (DP15).
- Where compromise cannot be prevented, systems SHOULD let participants and communities challenge a questionable action and reconstruct an accountable record of it (DP22).

5.5 Containment and recovery

The reach of any compromise is bounded, identifiable, and proportionate to the authority the compromised component actually held.

- Federated services MUST NOT accept privileges asserted by another community without verifying them against that community's published keys and policies.

- Implementations MUST define containment and recovery procedures for detected compromise: isolate, revoke, notify affected participants, restore.
- Security-critical operations MUST produce audit evidence protected against tampering and disclosed in proportion to privacy (DP4).

5.6 Graceful degradation

When a dependency becomes unavailable or untrustworthy, the Meta-Layer keeps as much safe function as possible and says what it has lost.

- Meta-Layer clients MUST fail closed for privileged actions when a security check cannot be completed.
- Meta-Layer clients SHOULD keep low-risk functions, such as reading public content, available while privileged ones are suspended.
- Systems MUST show participants when protections are reduced, and which ones.

5.7 Verifiable security claims

Resilience is demonstrated, not asserted.

- Implementations MUST publish their threat model: the threats considered, the boundaries tested, the results observed, and the limitations that remain.
- Implementations SHOULD submit to adversarial testing and independent review, with reproducible evidence.

5.8 Federated protection sharing

Communities share what they learn without surrendering autonomy to a central authority.

- Communities SHOULD be able to publish and consume verified vulnerability advisories and remediation guidance through a common, signed format.
- Each community MAY decide which advisories to act on, and SHOULD record that decision.

5.9 Anchor integrity and declared page capability

An annotation stays attached to what its author meant, or says plainly that it no longer is. Section 8 describes how pages break anchors and what participants see.

- Meta-Layer clients MUST store enough about each anchor – the quoted text with its surrounding context, and a fingerprint of the passage when the note was made – to detect that the passage has changed.
- Meta-Layer clients MUST NOT silently reattach an annotation to different text. When the match falls below a declared confidence, the annotation is shown as detached, with its original quote.
- Meta-Layer clients MUST show readers when the passage under an annotation has changed since it was written.
- Meta-Layer clients MUST tell participants, before they write, when a page supports only page-level notes or none at all.
- Meta-Layer clients SHOULD suppress in-page overlays, or keep them read-only, on sign-in, payment, and similar sensitive pages.

6. Layer Model and Trust Relationships

This section is a reference model. It names the layers the Meta-Layer runs on and beside, and states what each relationship may and may not carry. Implementations declare their own boundaries against it.

6.1 The layers

Layer	Examples	What the Meta-Layer relies on it for
Device and operating system	Laptop, phone, OS	Local execution, storage, input, and display
Network and infrastructure	ISP, DNS, CDN, hosting	Transport and availability
Browser	Chrome, Firefox, Safari	Process isolation, extension sandboxing, origin separation
Host website and content	News sites, apps, embedded media	The page being discussed – as content only
Meta-Layer clients	Extensions, web embeds, SDK integrations	Rendering the Meta-Layer for a participant on a page

Meta-Layer services	Identity, communities, messages, governance records	Authoritative state and authorization
Federation	Other communities and their services	Shared identity, content, and decisions across communities

AI agents and third-party services are not a layer of their own. They act within one of these layers, with authority delegated by an accountable principal (Section 5.3).

6.2 Authority flows from principals, never from content

Authority originates with authenticated participants and community governance. It flows down to the clients and agents acting on their behalf.

It never flows up from lower layers. A host page, a network path, or a content item can supply information to the Meta-Layer, but cannot grant itself or anyone else the right to act in it.

Example: A host page sends a message to an overlay claiming the participant has approved a post. The overlay treats it as a request and asks the participant, because only the participant's own action, verified by Meta-Layer services, can approve it.

6.3 Relationships between layers

Each relationship between adjacent layers has a defined limit.

Relationship	The upper layer may rely on	The upper layer must never inherit
Client ← device and OS	Execution and display, which cannot be avoided	Assurance that a compromised device is safe (declared out of scope)
Client ← browser	Sandboxing and origin isolation	Assumptions beyond the browser's documented guarantees
Client ← host website	The page's content, as data	Any authority, credential access, or ability to speak for the participant
Services ← client	Authenticated requests	Trust in client-supplied claims about permissions or identity
Community ← federated community	Signed assertions that can be verified	Privileges the other community cannot prove under its own policies
Agent ← content	Information to summarize or analyze	Instructions or expanded authority

6.4 Relationships between peers

Some of the riskiest relationships are between peers at the same layer: several Meta-Layer clients on one page, or several communities in one federation.

- Peers coordinate through declared interfaces only. No client reads another client's interface, storage, or messages.
- Peers do not imitate one another. A client that copies another's icon, name, or visual identity is non-conforming.
- Coordination conveniences between peers, such as the shared launcher in Section 7, never become trust anchors.

6.5 Obligations downward

The Meta-Layer has duties to the layers beneath it, not only protections against them.

- Meta-Layer clients do not break, alter, or slow the host page beyond what rendering the overlay requires.
- Meta-Layer clients do not carry data from one site into another site's context without the participant's explicit action.
- Meta-Layer clients respect regions the host page declares as reserved for its own controls (Section 7.3).

6.6 The protection boundary

Some compromises cannot be defended against at the Meta-Layer. A compromised operating system controls local execution. A disrupted network prevents communication. A host page can always draw lookalike elements inside its own area.

Each implementation states which layers it treats as trusted, which it defends against, and which are outside its protection boundary. A participant can read that statement before relying on the implementation.

7. Shared Overlay Surfaces and Launcher Coexistence

Meta-Layer clients share two scarce things: the participant's screen and the participant's trust in what appears on it. This section sets rules so that seve-

ral clients can coexist on one page without covering, crowding, or imitating one another.

7.1 One shared launcher per page

Conforming clients do not each inject their own floating launch icon. They register a launch entry with a single shared launcher on the page – a dock that lists every active Meta-Layer client.

- The first conforming client on a page hosts the launcher. Clients loaded later find it and register with it instead of drawing their own.
- Each registration declares the client's name, icon, publisher, and requested actions.
- If the hosting client unloads, another registered client takes over hosting without losing the participant's arrangement.

Example (hypothetical): A participant runs a community overlay, a translation overlay, and a note-taking overlay. Instead of three icons stacked in one corner, a single dock shows three entries. The participant opens any one with one click.

7.2 The participant arranges the launcher

The launcher belongs to the participant, not to any client.

- The participant chooses the launcher's edge and position, the order of entries, and which clients are shown, hidden, or pinned.
- The arrangement persists per participant and, where the participant chooses, per site.
- No client can move itself ahead of others, enlarge its own entry, or hide another client's entry.

7.3 Placement and collision rules

Where clients must draw outside the launcher – panels, highlights, inline tags – they follow shared placement rules.

- Each client declares the screen regions it intends to occupy. Clients do not overlap another client's declared region.
- Host pages can declare reserved regions, such as their own chat, consent, or checkout controls. Clients keep out of them.

- When regions conflict and cannot be negotiated, clients fall back to a deterministic order: the participant's arrangement first, then registration order. Raising one's own stacking order to win a conflict is non-conforming.
- When no conforming launcher is present, a client places its icon away from occupied regions and offers to join a launcher as soon as one appears.

7.4 Coordination is not trust

The page can see and imitate anything drawn inside it, including the shared launcher. The launcher coordinates placement. It does not prove identity.

- Actions that carry authority – signing in, approving, posting as the participant, granting agent permissions – happen in surfaces the host page cannot draw over or read, such as the browser's own side panel or toolbar.
- Clients give participants a way to confirm that a surface is genuine, such as a check reachable only from the browser's own interface.
- Launcher registrations are signed by the client's publisher, so a lookalike entry can be detected and flagged.

7.5 Accessibility and focus

Several overlays on one page also compete for keyboard focus and screen reader attention.

- The shared launcher is reachable by keyboard and announced once to assistive technologies, not once per client.
- Only one client panel holds focus at a time, and the participant can always return focus to the host page.
- Overlays follow DP21 and DP23 for modality and language access.

8. Page Compatibility and Anchor Integrity

Meta-Layer annotations attach to passages on pages the Meta-Layer does not control. Some pages cannot hold stable anchors, some block overlays entirely, and some should not carry overlays at all. This section names

those cases and describes what participants see in each. The requirements are in Section 5.9.

8.1 How pages break anchors

Anchors fail for structural reasons, most of them ordinary:

- **re-rendered pages:** application frameworks replace page elements on every update, and generated class names change between releases, so anchors based on page structure stop matching
- **partially loaded pages:** long lists and feeds keep only the visible rows on the page, and collapsed or lazily loaded sections are absent until opened, so the anchored passage may not exist when the client looks for it
- **changing text:** relative timestamps, counters, rotating content, experiments, and personalized or regional versions mean two participants at the same address may not see the same words
- **edits and repeats:** the passage is revised, or the same words appear more than once on the page, so a quote alone cannot say which occurrence was meant
- **navigation without reloading:** single-page applications change the address without loading a new page, and a client that does not notice keeps showing the previous page's annotations
- **reflow:** resizing, zoom, reader modes, and translation or theming extensions rewrite or move text, breaking any anchor based on screen position
- **no text to anchor:** text drawn on a canvas, content inside closed components or cross-origin frames, and media such as PDFs, images, and video need region or time anchors, not text anchors

The same content can also live at many addresses, through tracking parameters and mirrors, and different content can live at one address. A page's declared canonical address helps, but it is a claim made by the page.

8.2 Page capability levels

Each client classifies the current page and shows the result before the participant writes.

Level	What participants can do	Typical cause

Full	Anchor notes to passages	Stable text the client can match and fingerprint
Page-level only	Attach notes to the page as a whole	Text that is drawn, hidden, or changes on every visit
Read-only	Read existing notes; not write new ones	Sensitive pages (Section 8.4)
Unsupported	Nothing on the page; the side panel stays available	The browser or the page blocks the client, with the reason shown

Example: A participant opens a design tool that draws its text on a canvas. The overlay shows "Notes on this page attach to the page, not a passage" before the participant starts typing, instead of accepting a passage anchor that will fail for every reader.

Pages block clients for reasons outside the Meta-Layer: content security policies that refuse injected scripts (MDN: Content Security Policy), browser pages where extensions cannot run, enterprise policies that block extensions on certain sites, and scripts that remove anything they did not draw. Clients report these as the cause rather than failing silently.

8.3 Anchors that can be checked

An anchor records more than one way of finding its passage – the quoted text with words before and after it, its position, its place in the page's structure – following the selector model of the W3C Web Annotation Data Model (W3C, 2017). It also records a fingerprint of the passage at the time of writing.

- When every selector agrees, the annotation is shown normally.
- When the passage has changed since the note was written, readers see that it changed, and can see the original quote.
- When no selector finds the passage with enough confidence, the annotation is shown as detached, with its original quote, rather than moved to the nearest match.

This answers both anchor threats. Drift (Section 3.11) becomes visible instead of silent. Spoofing (Section 3.10) is detectable because the page can rewrite the passage, but cannot make the new version match the stored fingerprint.

8.4 Pages where overlays step back

On sign-in, payment, banking, health, and similar pages, in-page overlays are suppressed or read-only by default. These pages are where interface redressing (Section 3.1) and credential extraction (Section 3.3) do the most damage, and where an overlay is least needed.

Participants can still open the side panel, which the page cannot draw over.

Anchors on pages behind a sign-in or paywall quote content that other readers may not have access to. Clients show only as much of the quote as the reader could see for themselves, under DP4.

No web standard currently lets a site declare that it does not want to be annotated. Whether the Meta-Layer honors such a signal, and which one, is an open question (Section 19).

8.5 Learning which pages break

Re-anchoring outcomes are a signal. When clients report, in aggregate and without identifying participants, how often anchors on a site detach, communities can see which pages are unstable without waiting for complaints.

A shared, signed registry of page behaviors – sites that block overlays, sites whose text is drawn, sites that re-render constantly – lets every conforming client set the right capability level on the first visit. The registry follows the same signing and contestability rules as security advisories (Section 5.8).

9. Tensions and Tradeoffs

9.1 Unavoidable dependencies

The objective is not absolute independence from the infrastructure beneath. Some dependencies cannot be removed.

The objective is to minimize inherited trust, prevent unauthorized elevation of lower-layer authority, contain compromise, preserve recoverability, and make residual risks explicit.

9.2 Security and openness

Strict boundaries make the Meta-Layer harder to extend. DP7 requires that third-party tools can plug in without prior permission. DP24 requires that they cannot inherit authority by doing so. Communities weigh security against accessibility, usability, decentralization, privacy, performance, and the ability of independent developers to innovate.

9.3 Shared launcher and client independence

A shared launcher reduces contention but asks independent clients to cooperate. It can also become a point of capture if one client controls how others appear. The launcher therefore belongs to the participant (Section 7.2), and its behavior is defined by an open interface rather than by any single client.

9.4 Evidence and privacy

Audit evidence makes compromise visible and recoverable. It also records what participants did. Evidence is kept in proportion to the risk it addresses, under DP4.

9.5 Annotation and the host site's autonomy

Annotating a page without its owner's permission is part of what makes the Meta-Layer independent. It also means the Meta-Layer depends on pages that owe it nothing and may change, block, or resist it. Showing a detached note honestly, or declining to anchor at all, is preferred to anchoring that only appears to work.

10. Governance, Accountability, and Agency Surfaces

Cross-layer protections matter only if participants and communities can see them and act on them.

Participants must be able to:

- see which Meta-Layer clients and agents are active on the current page, who publishes them, and what authority each holds
- confirm that a surface asking for a sign-in or approval is genuine
- arrange, hide, or remove any client's launcher entry

- revoke a client's, session's, or agent's authority immediately
- see when protections are reduced and which ones
- see what the current page supports – passage notes, page-level notes, reading only, or nothing – and why
- see when a note's passage has changed or its anchor has detached, and read the original quote
- report a suspected compromise and receive a traceable response

Communities must be able to:

- quarantine a client version, federated service, or agent class pending review
- publish and consume signed security advisories
- review incidents with access to the relevant evidence
- decide which federated assertions they accept, and record why

Example: A participant sees an unfamiliar entry in the launcher claiming to be their community client. They open the browser's side panel, which shows that the entry's signature does not match the community client's publisher. They report it in one action, and the community quarantines the impersonating extension for its members.

11. Incentives and Power Analysis

Screen space and trust are both attention, and attention is valuable.

What must be visible:

- which client controls the shared launcher and how entries are ordered
- who issues security advisories and who acts on them
- who certifies, tests, or reviews implementations, and who pays them

Capture patterns include:

- **placement capture:** a client vendor pays or negotiates for a better position in the launcher or overrides the participant's arrangement
- **certification capture:** a single reviewer becomes the gatekeeper for what counts as secure

- **security as lock-in:** a platform uses resilience requirements to exclude independent clients
- **advisory weaponization:** a community issues false advisories to quarantine a competitor

DP24 expects ordering to remain under participant control, review to be plural and verifiable, and advisories to be signed, attributable, and contestable.

12. Community Signals Informing DP24

Recurring signals include:

- overlay launch icons from different extensions stacking on and covering one another and the host site's own controls
- uncertainty about whether a sign-in prompt belongs to the overlay or to the page
- concern that AI agents can be steered by content on the pages they read
- distrust of security claims that cannot be checked
- worry that one compromised community or extension could reach all of a participant's communities
- annotations failing to stay attached on pages without stable anchors

These signals are not usability complaints. They indicate that participants cannot yet tell which layer is speaking to them, or how far a failure in one layer can travel.

13. Evaluation Criteria

Each criterion is written as a question that targets a specific defect. Each test documents its threat assumptions, attack path, observed outcome, mitigation, and remaining risks. A successful test is evidence for a particular protection, not for resistance to all possible attacks.

13.1 Boundary integrity

- Can a webpage forge a message or privileged command that Meta-Layer services accept?

- Can a webpage, embed, or sibling extension read authentication material from a Meta-Layer client?
- Can content on a page cause an AI agent to act beyond its delegated authority?

13.2 Containment

- Can a compromised federated service change records or decisions in another community?
- Can a protected community record be altered without authorization and without detection?
- After a simulated compromise, how long do revocation and participant notification take?

13.3 Degradation

- Does the system continue privileged operations when a critical security dependency is unavailable?
- Does it tell participants which protections are reduced?

13.4 Surface coexistence

- With three conforming clients loaded on one page, does any launch icon cover another or a host-declared reserved region?
- Can a sibling extension place an icon over a conforming client's launcher entry, or imitate it, without being detected?
- Can every authority-bearing action be completed only from a surface the host page cannot draw over?

13.5 Anchor integrity

- When the anchored passage is edited, do readers see that it changed?
- When the passage is removed, is the annotation shown as detached rather than moved to other text?
- Can a test page that shows different text to the author and to later readers make a note appear attached to text its author never saw, without detection?
- On a page whose text is drawn on a canvas, or on a page that blocks the client, does the participant learn what the page supports before writing?

14. Implementation Patterns

14.1 Trusted surface separation

Authority-bearing interactions run in browser-owned surfaces, such as an extension side panel, while the in-page overlay shows only content and launch entries.

14.2 Server-side capability checks

Each privileged request carries a short-lived, scoped capability that the service verifies. The client's own view of its permissions is never trusted.

14.3 Content tainting for agents

Agent runtimes label every input with its origin and keep untrusted content out of the instruction channel, satisfying Sections 5.1 and 5.3.

14.4 Launcher registration protocol

A small open protocol for registering with, hosting, and handing over the shared launcher, with signed registrations (Section 7).

14.5 Signed advisory feeds

Communities publish advisories in a common signed format that others can subscribe to and act on independently (Section 5.8).

14.6 Multi-selector anchors with fingerprints

Each anchor stores a quote with its surrounding words, a position, and a structural path, following the W3C Web Annotation selector model, plus a fingerprint of the passage. The client re-anchors by trying each selector, scores its confidence, and shows the result (Section 8.3).

14.7 Page behavior registry

A shared, signed list of sites and the capability level each supports, fed by aggregate re-anchoring outcomes (Section 8.5).

15. Foresight and Failure Design

Resilience degrades over time unless it is designed to be maintained.

Likely failure paths:

- **assumption drift:** a declared threat model stops matching how the system is actually deployed
- **alert fatigue:** degradation notices appear so often that participants ignore them
- **launcher erosion:** clients gradually stop registering with the shared launcher and return to their own icons
- **advisory poisoning:** a compromised or captured source floods the federation with false advisories
- **selector rot:** a popular site redesigns, and thousands of its annotations detach at once
- **capability shift:** future AI agents or new browser capabilities change which layers can reach which

Design responses:

- review the threat model on a fixed schedule and on every major release
- make notices rare, specific, and actionable
- test launcher conformance in every client release
- keep the original quote of every annotation, so detached notes remain readable and can be reattached by people
- require advisories to be signed and rate-limited, and give communities the final decision
- treat each new class of agent or client capability as a new boundary to declare

16. Relationship to Other Desirable Properties

- DP1 **(Federated Authentication & Accountability)** establishes who can act and how actions are attributed; DP24 keeps that attribution intact when adjacent systems are compromised.
- DP2 **(Participant Agency and Empowerment)** gives participants real controls; DP24 makes sure the controls they see are genuinely the Meta-Layer's.
- DP4 **(Data Sovereignty and Privacy)** gives participants control of their information; DP24 keeps that control from being bypassed through lower

layers.

- **DP7 (Simplicity and Interoperability)** lets tools plug in freely; DP24 ensures that interoperability does not transfer authority, and that plugged-in clients share the screen by rule.
- **DP11 (Safe and Ethical AI)** and **DP13 (AI Containment)** define acceptable agent behavior and its enforceable limits; DP24 makes sure those limits survive manipulation from lower-trust environments.
- **DP12 (Community-based AI Governance)** decides who sets the rules; DP24 keeps compromised layers from changing them.
- **DP14 (Trust and Transparency)** and **DP15 (Security and Provenance)** make information and software verifiable; DP24 adds isolation, containment, and recovery across the architecture.
- **DP21 (Multi-modal)** and **DP23 (Universal Participation & Linguistic Interoperability)** set access requirements that shared overlays must meet.
- **DP22 (Civic Memory & Epistemic Continuity)** preserves accountable records; DP24 keeps them reconstructable after compromise, and keeps annotations attached to what their authors actually saw.

Without DP24, each of these properties holds only as long as every system beneath the Meta-Layer stays honest.

17. Non-Goals and Explicit Boundaries

DP24 does not:

- promise that the Meta-Layer stays secure or available when every layer beneath it is compromised
- protect participants on a device whose operating system is compromised
- prevent a host page from drawing lookalike elements inside its own area; it requires that authority-bearing actions happen where the page cannot
- mandate a browser, extension framework, attestation technology, or security vendor
- create a central certification authority
- guarantee that every page can be annotated; it requires that participants are told when one cannot

- replace DP13's containment dimensions or DP15's supply-chain and provenance requirements

DP24 defines the trust boundaries between the Meta-Layer and the systems it runs on and beside, and what must hold at each one.

18. Minimum DP24 Alignment

A DP24-aligned implementation meets all of the following:

- Meta-Layer services MUST authorize privileged actions independently of webpage content and client-supplied claims.
- Meta-Layer clients MUST keep credentials out of any context readable by host-page scripts or other extensions.
- Meta-Layer services MUST check agent actions against authenticated, scoped, and revocable delegation.
- Federated services MUST NOT accept privileges from another community without verifying them.
- Implementations MUST produce tamper-protected audit evidence for security-critical operations.
- Implementations MUST define and test containment and recovery procedures, including participant notification.
- Meta-Layer clients MUST fail closed for privileged actions when a security check cannot complete, and MUST tell participants.
- Meta-Layer clients MUST register with a shared launcher when one is present, and MUST NOT cover, reorder, or imitate another client's entry.
- Meta-Layer clients MUST perform authority-bearing actions only in surfaces the host page cannot draw over or read.
- Meta-Layer clients MUST detect when an anchored passage has changed, MUST NOT silently reattach an annotation to different text, and MUST show detached annotations with their original quote.
- Meta-Layer clients MUST tell participants what the current page supports before they write.
- Implementations MUST publish their threat model, protection boundary, limitations, and evidence of adversarial testing.

If any of these is missing, the implementation's security depends on the honesty of the systems beneath it.

18.1 Strong DP24 alignment

A strongly aligned implementation also binds sessions to devices, publishes reproducible adversarial test results, takes part in signed advisory sharing, contributes to a shared page behavior registry, and passes the surface coexistence and anchor integrity tests in Sections 13.4 and 13.5 against other conforming clients.

18.2 Failure conditions

An implementation must not claim DP24 alignment if a webpage can trigger a privileged action, if credentials are readable from the page, if its client draws over or imitates another client's launcher entry, or if it moves annotations to different text without telling readers.

19. Open Questions and Future Work

These questions are not blockers. They are where reasonable implementers currently disagree.

- Which lower-layer compromises should the Meta-Layer be expected to withstand, and which are outside its protection boundary?
- What minimum guarantees should every conforming extension, SDK, web embed, and federated service provide?
- How can communities verify one another's security claims without a central certification authority?
- How can security improvements spread quickly across the federation while respecting local autonomy?
- What evidence is enough to claim that an implementation satisfies DP24?
- Can the shared launcher be standardized at the browser level, so the page cannot imitate it at all?
- How should the launcher behave on mobile and in-app browsers, where screen space is smallest?
- Should the Meta-Layer honor a signal from a site that does not want to be annotated, and if so, which signal, and with what exceptions?

- When participants at the same address see different content, whose version does an anchor describe?
- Who maintains the page behavior registry, and how is a site's listing contested?

20. Path Toward ML-RFC

- Publish a threat model template that implementations fill in against the layer model (Section 6).
- Specify the shared launcher registration and handover protocol, with signed registrations (Section 7).
- Build an adversarial test page and conformance suite covering the criteria in Section 13.
- Run the first demonstration: a deliberately malicious webpage attacking a signed-in Meta-Layer interaction, with two or more independent clients on the same page.
- Define a signed advisory format for federated vulnerability sharing.
- Specify the anchor format – selectors, fingerprint, confidence – and the page capability levels (Section 8), with a test corpus of pages known to break anchors.
- Iterate with extension, SDK, and federation implementers; progress is measured by working, tested systems.

Candidate requirements for the ML-REQ (Meta-Layer Requirements) stage:

Theme	Candidate requirement
Content-independent authority	Privileged actions are authorized independently of untrusted webpage content and client-supplied claims
Credential isolation	Credentials and privileged operations are isolated from lower-trust execution contexts under the declared threat model
Agent authority	Agent actions are checked against authenticated, scoped, and revocable authority
Federation boundaries	Federation boundaries prevent unauthorized transfer of privileges between communities
Audit evidence	Security-critical operations produce protected audit evidence

Containment and recovery	Systems support defined containment and recovery procedures after detected compromise
Surface coexistence	Clients share a participant-controlled launcher and do not occlude or imitate one another
Anchor integrity	Annotations detect changes to their passage, are never silently reattached, and show their original quote when detached
Declared page capability	Clients tell participants what each page supports before they write
Published assumptions	Implementations publish their security assumptions, limitations, and adversarial testing evidence

21. Closing Orientation

DP24 decides whether the Meta-Layer is a trustworthy coordination layer or simply the web's weaknesses reproduced one level higher.

When DP24 is strong, participants can take part on any page, including a hostile one, without the page speaking for them. Several Meta-Layer clients share the screen without covering one another, and a failure in one layer stops at its boundary.

When DP24 is weak, every website, extension, and federated peer is a potential administrator of the Meta-Layer.

Safe and accountable human and AI coexistence needs defensible boundaries between the Meta-Layer and the systems it runs on.

> *The Meta-Layer must not simply rise above the web. It must demonstrably resist, contain, and recover from attacks originating beneath it.*

Continue the conversation on DP24 online →

Acknowledgements

This book exists because many people chose to treat the Meta-Layer not as a finished product but as a shared inquiry – one worth sustaining across meetings, drafts, inscriptions, and open challenge rounds.

We are deeply grateful to Dr. Suzanne Sincavage and the Foundation for Biodefense Research for their crucial support of the Meta-Layer Initiative to ensure the forward progress of this work.

These acknowledgements, like the rest of this book, are preserved as enduring digital artifacts – inscribed on Bitcoin and intended to live onchain, or through whatever preservation methods may follow – so that future generations can see who supported this work and who helped bring it into being.

Contributors

Contributors to *The Layered Web: The Desirable Properties of a Meta-Layer* are listed below in order of contribution – honoring those whose ideas, critiques, implementations, and stewardship arrived earliest in the collective arc of this work. Rather than a ranking of authority, this is a record of how the book came to be. The list will grow as new contributors join the Desirable Properties Challenge and as earlier work is reconciled with the permanent record.

1. Vint Cerf – At the Meta-Layer Initiative kickoff meeting on September 16, 2024, Vint Cerf suggested that the community begin by defining the *desirable properties* of a Meta-Layer. He also recommended that we start off with federated strong authentication. That framing launched the Calls for Input, the satplication synthesis, and the open edition you are reading now.

2. Daveed Benjamin – orchestration; co-hosted the Meta-Layer Initiative kickoff meeting; proposals adopted in DP4, DP5, DP11, DP14, DP15, DP22; co-editor of DP1, DP5, DP8, DP9, DP14, DP15.
3. Liz Sweigart – co-hosted the Meta-Layer Initiative kickoff meeting; Second Call for Input (2025); proposals adopted in DP4, DP6, DP15.
4. Brad deGraf – co-hosted the Meta-Layer Initiative kickoff meeting; Second Call for Input (2025).
5. Phahsa Ras – Second Call for Input (2025); proposals adopted in DP4, DP22; co-editor of DP4, DP6, DP17, DP22.
6. Melinda Woods – proposals adopted in DP17; co-editor of DP4, DP17.
7. Joshua Colbert – proposals adopted in DP6.
8. Cindy Mason – First Call for Input (2024); proposals adopted in DP1, DP11, DP13, DP14, DP15.
9. Paul Werbos – First Call for Input (2024); proposals adopted in DP3, DP6, DP13, DP15, DP16.
10. David Bray – First Call for Input (2024); proposals adopted in DP2.
11. David Michaelis – First Call for Input (2024).
12. Brian Donohue – First Call for Input (2024); proposals adopted in DP1.
13. Dave Troy – First Call for Input (2024); proposals adopted in DP3.
14. Vandana Upadhyay – First Call for Input (2024); proposals adopted in DP1, DP15.
15. Quincey El – First Call for Input (2024).
16. Javier Gaxiola – Second Call for Input (2025); proposals adopted in DP2.
17. Chris Santos-Lang – Second Call for Input (2025); proposals adopted in DP2, DP3, DP7, DP11.
18. Paul Carpenter – Second Call for Input (2025).
19. Ruben Diaz – Second Call for Input (2025); proposals adopted in DP1, DP7, DP14.
20. Tristan Harris – Second Call for Input (2025).
21. Stephanie Hervey – Second Call for Input (2025); proposals adopted in DP7, DP8, DP15.
22. Alex Nassarius – Second Call for Input (2025); proposals adopted in DP2, DP15.

23. Patrick Hoagland – Second Call for Input (2025); proposals adopted in DP18.
24. Michael Witmore – Second Call for Input (2025).
25. Scott Frankum – Second Call for Input (2025); proposals adopted in DP6, DP7.
26. Aa Ho – Second Call for Input (2025).
27. Wojak K – Second Call for Input (2025).
28. Sandeep Chakravartty – Second Call for Input (2025); proposals adopted in DP10.
29. Lindsay Jane – Second Call for Input (2025).
30. Rick Byers – Second Call for Input (2025).
31. Eric Schneider – Second Call for Input (2025); proposals adopted in DP10, DP19.
32. Michelle Thuo – Second Call for Input (2025); proposals adopted in DP8, DP14.
33. Scott Yates – Second Call for Input (2025); proposals adopted in DP5.
34. Kevin Clark – suggested the title of the book.
35. Parham Pourdavood – proposals adopted in DP21.
36. Alan Karp – proposals adopted in DP1.
37. Gaowei Chang – proposals adopted in DP7.
38. Sam Tsang – proposals adopted in DP8.
39. Jingyi Shirley – proposals adopted in DP8; co-editor of DP8.
40. Hilda Campbell – proposals adopted in DP8; co-editor of DP2.
41. Ying Zhou – proposals adopted in DP9; co-editor of DP9.
42. Tianqun Pan – proposals adopted in DP10.
43. Shambhavi Goyal – proposals adopted in DP10.
44. Salwa El Mais – proposals adopted in DP10; co-editor of DP10.
45. Dee Frewert – proposals adopted in DP10, DP23.
46. Nell Clasby – proposals adopted in DP11.
47. Diane Strahan – proposals adopted in DP13.
48. Ron Petty – proposals adopted in DP13.
49. Guido van Nispen – proposals adopted in DP14.
50. Ida Na-Tei – proposals adopted in DP16.

51. Kyle Shulman – proposals adopted in DP16.
52. Melanie Bielefeld – proposals adopted in DP17.
53. Ramina Tankacheyeva – proposals adopted in DP18.
54. Zanette Johnson – proposals adopted in DP18.
55. Nada El Mais – proposals adopted in DP18.
56. Mike McGuire – proposals adopted in DP21; co-editor of DP21.
57. Gael Van Weyenbergh – proposals adopted in DP12.
58. Sara Pascual – proposals adopted in DP22; co-editor of DP4, DP22.
59. Micah Nyatindo – proposals adopted in DP7, DP15; co-editor of DP7, DP14, DP15.
60. Maureen Shea – proposals adopted in DP6; co-editor of DP6.
61. Leandra Preston – proposals adopted in DP23; co-editor of DP2, DP18, DP23.
62. Renad Baomar – proposals adopted in DP10; co-editor of DP10.
63. Jesse Asiedu-Akrofi – proposals adopted in DP10.
64. Aldo Velázquez – co-editor of DP16.
65. Christine Louise – co-editor of DP11.
66. Gintama – co-editor of DP3.
67. Laura Del Campo – co-editor of DP12, DP19, DP20.
68. LogicalFlo – co-editor of DP3.
69. Lord M.E.G.A Zion – co-editor of DP12.
70. Markabadi – co-editor of DP19.
71. Reed O'Neal – co-editor of DP5.
72. Stella Alvo – co-editor of DP11.

Editors

- Aldo Velázquez – co-editor of DP16.
- Christine Louise – co-editor of DP11.
- Daveed Benjamin – steering committee; co-editor of DP1, DP5, DP8, DP9, DP14, DP15.
- Gintama – co-editor of DP3.
- Hilda Campbell – co-editor of DP2.

- Jingyi Shirley – co-editor of DP8.
- Laura Del Campo – co-editor of DP12, DP19, DP20.
- Leandra Preston – steering committee; co-editor of DP2, DP18, DP23.
- LogicalFlo – co-editor of DP3.
- Lord M.E.G.A Zion – co-editor of DP12.
- Markabadi – co-editor of DP19.
- Maureen Shea – co-editor of DP6.
- Melinda Woods – co-editor of DP4, DP17.
- Micah Nyatindo – co-editor of DP7, DP14, DP15.
- Michelle Thuo – steering committee.
- Mike McGuire – co-editor of DP21.
- Phahsa Ras – steering committee; co-editor of DP4, DP6, DP17, DP22.
- Reed O'Neal – co-editor of DP5.
- Renad Baomar – co-editor of DP10.
- Salwa El Mais – co-editor of DP10.
- Sara Pascual – co-editor of DP4, DP22.
- Stella Alvo – co-editor of DP11.
- Ying Zhou – co-editor of DP9.