

DP7 – Interoperability

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 treated as data transfer rather than continuity of meaning, enforceability, and legitimacy.

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 re-use 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: 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 recentralizes 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.

5. Primary Mechanisms and Structural Conditions

5.0 Interoperability 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.

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.

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.

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.

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.

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.

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.

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.

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.

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 constrains it: external network calls are limited, access to private threads is denied, 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.

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.

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, fallback gracefully, and signal limitations.

A failure mode is silent incompatibility, where objects appear valid but behave incorrectly due to version mismatch.

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.

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 recentralize 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. 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.

10. Minimum Alignment (Non-Normative)

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.

11. 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.

12. 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.

13. 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.

14. 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.

15. 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 recentralize 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.

DP7 ensures the meta-layer is a network, not a set of islands.

Without it, all other properties collapse into silos.

With it, coordination becomes truly global and composable.