

DP12 – Community-Based AI Governance (V1.1)

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 ensures governance is not abstract or centralized, but participatory, legible, and enforced at the interface where AI behavior is experienced.

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 enforcement 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

5.0 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)

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.

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.

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)

Governance 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.

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.

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

Governance memory

All policy objects, decisions, disputes, and outcomes form a linked, versioned history that supports learning and prevents repetition of past failures.

5.1 Zone-scoped governance

Communities define rules within specific zones of interaction, aligned with context and risk, with clear boundaries and inheritance where applicable.

5.2 Policy as executable objects

Rules are expressed in machine-enforceable formats that bind to runtime behavior and can be tested, simulated, and verified before deployment.

5.3 Governance loops

A continuous cycle of propose → implement → observe → contest → revise, with time bounds and clear state transitions.

5.4 Governance memory

Decisions, rationale, and outcomes are persistently recorded, searchable, and linked to policy versions and receipts.

5.5 Incentive surfaces (DP9 alignment)

Communities can see and influence optimization targets shaping AI behavior, including tradeoffs and red lines that gate unacceptable outcomes.

5.6 Integration with containment (DP13 alignment)

Rules are enforced through containment mechanisms with graduated responses and clear audit trails.

5.7 Auditability and provenance (DP14–DP15 alignment)

Governance actions and outcomes are logged with verifiable evidence and accessible summaries for participants.

5.8 Delegation and representation (DP2, DP3 alignment)

Participants can delegate governance roles with explicit scope, revocability, and accountability, including term limits where appropriate.

5.9 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.

5.10 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.

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. 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**.

10. Minimum Alignment (Non-Normative)

A DP12-aligned system must meet a baseline where governance is not only declared, but operationally binding.

At minimum, systems must:

- 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 governance receipts for all material outcomes (DP15)

- provide appeal, correction, and escalation pathways with defined timelines and outcomes
- couple governance rules to containment and enforcement systems (DP13)
- preserve complete policy history with versioning, rationale, and traceability

If any of these conditions are missing, governance is functionally symbolic, regardless of how comprehensive the written policies appear.

11. 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
-

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. 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
- **cross-system inconsistency:** governance behaves differently across environments, undermining legitimacy

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.

14. 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

Progress should be demonstrated through working systems, not only specifications.

15. 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.