DP12 – Community-based AI Governance

AI systems in the Meta-Layer are governed not by corporations — but by the communities that use them.

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.

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.

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).

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.

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.

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.

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

---

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.

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.

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.

Proposal and standing

Communities 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
- standing to propose is stated explicitly, including whether affected non-members may petition
- 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.

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 (5.10)
- 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.

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.

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 (5.10)
- adoption produces a signed policy object with version, rationale, and effective date
- notice periods before enforcement begins, proportional to the change's impact

Failure mode: silent adoption, where a rule takes effect before those subject to it can observe that it changed.

Delegation and representation

Participants may delegate governance capacity, and delegation must remain accountable.

- scopes are explicit and revocable at any time (5.8)
- 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

Failure mode: representation drift, where delegation is durable and revocation is theoretically available but practically inert.

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.

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

Failure mode: appeal as absorption, where objections are collected, resolved individually, and never change the rule that produced them.

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.

Failure mode: coordination capture, where cross-community structures become the venue through which local autonomy is overridden.

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 (5.4)

Failure mode: rule sediment, where obsolete constraints persist because no process retires them and enforcement becomes selective by necessity.

Why this matters: The execution layer determines whether rules bind. Process determines whether they deserve to.

Policy-Bound Verification

Governance that binds policy to behavior must be able to demonstrate that it did so. Without verification, a governance 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.

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.

Receipts as verifiable artifacts

Governance receipts (5.0) 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.

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 AI Governance Processes)

Failure mode: blind adoption, where rules are enacted without knowing what they will do.

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 (**Foresight and Failure Design**)

Failure mode: specification gaming, where the audit passes and the harm continues.

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.

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 (5.0)
- 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.

Verification across boundaries

When policies move between systems, verification must move with them or the loss must be declared (5.9, 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.

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.

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.

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**.

Minimum DP12 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.

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

---

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.

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.
