

DP9 – Developer and Community Incentives

The Meta-Layer gives developers and community builders the tools and incentives to create shared value across the web.

Purpose of This Draft

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.

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.

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.

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.

Primary Mechanisms and Structural Conditions

5.0 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.

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.

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.

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.

Attribution and lineage

Incentive systems must track contribution lineage across time and systems.

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.

Reward event splitting

Each value-generating event should distribute rewards across contributing layers.

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.

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.

Anti-gaming and containment integration

Incentive systems must integrate with containment mechanisms (DP13).

This includes rate limits on rewardable actions, detection of clustered or synthetic behavior, and discounting or quarantining suspicious contributions.

Incentives must not be exploitable at scale through automation or coordination.

A failure mode is incentive farming, where participants optimize for reward extraction rather than meaningful contribution.

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.

Incentive–ownership binding

Incentives must create pathways to ownership (DP20).

Participants who contribute consistently should accumulate stake, gain governance rights, and participate in long-term value flows.

Without this, incentives produce activity without durable power.

A failure mode is extractive participation, where contributors generate value but do not share in outcomes.

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.

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.

5.1 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.

5.2 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).

5.3 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.

5.4 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.

5.5 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.

5.6 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.

5.7 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.

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.

5.8 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.

5.9 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.

5.10 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.

5.11 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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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).

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).

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.

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.

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).

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.

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.

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.

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).

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 auditable event logs (DP12, DP13, DP15). Openness is at the point of entry; accountability is at the point of execution.

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.

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).

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.

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.

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.

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

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

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

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

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

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

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

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

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

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

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

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.

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.

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.

Minimum DP9 Alignment (Non-Normative)

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.

Partial compliance that omits execution, auditability, containment, or cross-system integrity should not be treated as alignment.

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.

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.

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.
