

DP20 – Community Ownership

1. Purpose of This Draft

This draft articulates Desirable Property 20 (DP20) as the condition under which communities meaningfully own the digital environments, surplus, and governance artifacts they co-create, not as marketing language about “community,” but as enforceable rights and responsibilities: stake, voice, upside, and continuity that persist across tools and seasons.

DP20 completes a loop with DP18 (feedback and reputation), DP19 (presence and engagement), and DP12 (community governance of AI): participation must be able to mature into ownership, otherwise “community” remains a rented audience for platforms.

DP20 connects to DP4 (data commons and collective data rights), DP6 (fair distribution of commercial surplus), DP9 (incentives that return value to contributors), DP3 (adaptive governance at scale), DP17 (financial sustainability), and DP1–DP2 (accountability and agency for collective actors).

If DP20 is weak, predictable failures follow: extractive growth on volunteer labor, token claims without decision rights, capture of governance by insiders, inability to fork when values diverge, and surplus flowing outward while risks stay local.

DP20 does not mandate a single cooperative legal form or chain. It defines minimum legitimacy conditions for what “ownership” may claim in the meta-layer.

2. Problem Statement

In today’s web, “community” is often a label applied to users whose labor, attention, and data build value they cannot capture, audit, or exit with. Governance surfaces are optional; economic upside is asymmetric; and forking social reality is impractical.

In practice, this produces recurring failures:

- moderators and maintainers burn out while platform valuation rises
- creators depend on distribution they do not control
- tokens grant speculation liquidity without durable rights
- data generated collectively is sold without collective consent (DP4)
- communities cannot carry norms, memory, or economic terms across apps (DP7)

These failures are structural: ownership is confused with usage; participation is confused with consent to extraction.

DP20 reframes ownership as operational: rights you can see, exercise, and defend, including pathways to fork, exit, and recapture surplus when legitimacy breaks.

3. Threats and Failure Modes

3.1 Ownership theater

Marketing claims “community-owned” while decisions and keys remain centralized.

Example: A DAO where a multisig admin can override any vote without published criteria.

Why this matters: DP20 requires honest mapping between claims and controls.

3.2 Token liquidity without duties or rights

Tokens trade while governance is inactive or non-binding.

Example: Voter apathy plus whale dominance makes governance a periodic headline, not a constraint on operators.

Why this matters: Ownership instruments must bind to decisions and surplus, or be renamed.

3.3 Commons maintenance without reciprocity

Value is extracted from a space while moderation, safety, and infrastructure remain underfunded.

Example: A viral forum hosts commerce ads but does not fund anti-abuse capacity.

Why this matters: Connects DP6 reciprocity and DP17 sustainability.

3.4 Capture by concentrated stakeholders

Early contributors or large holders lock rules that favor continuation of privilege.

Example: Fee switches require supermajorities that insiders can block indefinitely.

Why this matters: DP3 adaptive governance and DP12 dialectic must resist ossified capture.

3.5 Forklessness

When values diverge, participants cannot exit with continuity of identity, memory, or economic position.

Example: A community schism loses all history because export is blocked and namespaces are not portable (DP5, DP7).

Why this matters: Forking is a safety valve for legitimacy.

3.6 Collective data sale without governance

Aggregated or derived data monetizes communities without authorization pathways.

Why this matters: DP4 sovereignty requires collective decision surfaces, not only individual toggles.

3.7 Illegible membership and shadow elites

Informal power structures make decisions without published accountability.

Example: Moderators coordinate bans off-platform without records or appeals.

Why this matters: DP1 contestability and DP12 governance memory must extend to collective actors.

3.8 Philanthropic dependency

Communities rely on unsustainable grants with no path to owned revenue or shared surplus.

Example: A civic space runs on a yearly sponsor who can pull funding overnight.

Why this matters: DP17 sustainability and DP6 commerce reciprocity must appear in ownership design.

3.9 Legal hostility to collective action

Cooperative forms face banking, tax, and liability friction.

Why this matters: DP20 acknowledges law as a design constraint while still demanding honest digital rights bundles.

4. Core Principle

Community ownership in the meta-layer means participants and communities hold defensible rights to govern, benefit from, and responsibly steward the environments and surplus they co-create, including credible paths to fork, exit, and recapture value, with transparency, accountability, and continuity of memory.

Ownership is not a vibe. It is power with responsibility, visible to members and bounded by human-rights baselines and law.

Example: A community charter defines decision rights, surplus split, audit rights, and fork procedures; tools entering the zone must acknowledge the charter's machine-readable profile.

What this feels like: You are not the community in a slogan. You are a member with levers.

Without this: Participation is harvested; dignity is not.

5. Primary Mechanisms and Structural Conditions

5.0 Ownership Layer: Execution, Proof, and Exit

Community ownership requires more than charters and intent. It requires the ability to express, verify, exercise, and, when necessary, exit ownership as part of an operational system.

In many systems, “ownership” fails because it is not bound to decision rights, not provable in practice, or not portable across tools. Rights exist on paper while control remains elsewhere.

DP20 therefore requires a shared ownership layer composed of primitives that make ownership executable rather than symbolic.

Ownership objects

Ownership must be represented as structured, machine-readable objects that bind rights to actors and contexts.

An ownership object includes:

- scope: what environment, assets, or processes are owned
- rights: governance, economic, and data rights
- duties: stewardship and maintenance obligations
- transfer conditions: how ownership can change hands

This binds ownership directly to governance (DP3, DP12).

Ownership receipts

All ownership-relevant actions produce verifiable records.

An ownership receipt includes:

- who holds or exercised a right
- what action was taken (vote, allocation, fork, transfer)
- when it occurred
- what assets or policies were affected

These receipts make ownership auditable and connect to DP15 (provenance).

Surplus and flow proofs

Economic flows must be traceable to ownership claims.

Members must be able to see:

- how value enters the system
- how it is allocated
- how it maps to rights and contributions

Without flow proofs, ownership cannot capture upside in a defensible way (DP6, DP17).

Fork and exit primitives

Ownership must include executable pathways to exit or fork with continuity.

This includes:

- export of data and governance artifacts (DP4, DP7)
- migration of identifiers where honest (DP5)
- continuity of membership and contribution records

Forking is not a failure. It is a safety valve for legitimacy.

Membership and credential objects

Membership and contribution must be represented as portable, verifiable artifacts.

This enables:

- recognition across tools
- resistance to platform lock-in
- continuity of identity and stake

Ownership–governance binding

Ownership must map to decision rights in a visible way.

Participants must be able to see:

- who can decide what
- how ownership affects voting or delegation
- where overrides or special powers exist

Governance that cannot be owned is advisory. Ownership that cannot govern is symbolic.

Ownership memory

Ownership history must persist over time.

This includes:

- prior allocations and changes
- disputes and resolutions
- forks and mergers

Without memory, ownership becomes contestable without evidence.

These primitives do not replace the mechanisms below. They make them enforceable, portable, and auditable across contexts.

5.1 Charter and rights bundle

A written and machine-readable articulation of membership, decision rights, economic rights, data rights, fork rights, and sunset conditions.

This MUST include:

- explicit mapping between roles and powers
- conditions under which rights can change
- linkage to governance execution (DP3, DP12)

Verification:

- participants can inspect current and prior versions
- tools enforce charter constraints at interaction level

Failure modes:

- **charter drift** (rules change without visibility)
- **unenforced rights** (charter exists but does not constrain behavior)

5.2 Transparent surplus accounting

Revenue, fees, grants, and in-kind support are visible to members at useful granularity (DP6, DP17).

This MUST include:

- inflow sources and categories
- allocation breakdowns tied to functions
- linkage to ownership claims

Verification:

- members can trace value from source → allocation → outcome

Failure modes:

- **surplus opacity** (value cannot be traced)
- **extraction masking** (flows hidden behind aggregation)

5.3 Participatory budgeting and parameter control

Communities set budgets for safety, education (DP10), and incentives (DP9) through governed processes (DP12).

This MUST include:

- binding decision pathways
- budget execution tracking
- feedback loops into future allocations

Failure modes:

- **advisory budgeting** (decisions ignored)
- **capture of allocation power**

5.4 Credible exit and fork

Procedures to export collective artifacts, migrate identifiers where honest (DP5), and continue governance under divergent values without coercion.

This MUST include:

- executable export mechanisms
- continuity of identity, data, and governance artifacts

Failure modes:

- **exit suppression**
- **fork degradation** (loss of continuity)

5.5 Stewardship obligations

Ownership includes maintenance duties: moderation, security, inclusion, and accessibility commitments with accountability (DP1, DP15).

This MUST include:

- defined duties
- enforcement mechanisms
- reporting on fulfillment

Failure modes:

- **duty abandonment**
- **free-rider extraction**

5.6 Anti-capture controls

Rotation, conflict-of-interest rules, minority protections, and emergency pause pathways.

This MUST include:

- detection of concentration
- intervention pathways

Failure modes:

- **ownership capture**
- **governance lock-in**

5.7 Collective data governance

Shared corpuses require collective consent mechanisms and audit of downstream use (DP4).

This MUST include:

- consent pathways
- audit logs of use

Failure modes:

- **data extraction without consent**
- **loss of collective control**

5.8 Interoperable ownership artifacts

Membership and contribution receipts interoperate across tools (DP7).

This MUST include:

- portable formats
- degradation signaling across systems

Failure modes:

- **lock-in via incompatibility**

5.9 Succession and dissolution

Charters define succession for stewards and fair dissolution procedures.

This MUST include:

- transfer conditions
- fair distribution of residual value

Failure modes:

- **power vacuum**
- **unfair dissolution**

5.10 Intersectional inclusion in ownership

Governance design actively counters exclusion patterns through facilitation norms, translation, and accessible participation.

This MUST include:

- measurable inclusion practices
- correction mechanisms

Failure modes:

- **systemic exclusion**
- **ownership concentration through bias**

6. Governance, Accountability, and Agency Surfaces

DP20 requires that ownership is not only visible but **defensible and enforceable**.

Participants MUST be able to:

- verify ownership rights and mappings
- challenge illegitimate control or misallocation
- trigger governance review or dispute processes
- exercise exit and fork pathways without coercion

Communities MUST be able to:

- revoke or reassign ownership under defined conditions
- enforce stewardship obligations
- audit ownership actions and outcomes

Enforcement pathways include:

- rollback or invalidation of illegitimate actions
- governance-triggered reallocation of rights or surplus
- fork as a last-resort enforcement mechanism

Failure modes:

- **non-actionable ownership**
- **illegitimate control persistence**
- **ownership without recourse**

7. Incentives and Power Analysis

Ownership fails when incentives diverge from rights.

Key dynamics:

- **extraction loops:** value flows outward despite community contribution
- **token manipulation:** governance power decouples from responsibility
- **dependency capture:** infrastructure or distribution creates hidden control
- **soft capture:** influence accumulates without formal ownership

Adversarial patterns:

- governance capture through capital or coordination
- speculative ownership without stewardship
- platform rent extraction

Alignment requirements:

- ownership rights must correlate with value capture
- incentives must reward stewardship, not exploitation
- systems must detect divergence between formal ownership and real control

Failure modes:

- **symbolic ownership**
- **incentive inversion**
- **ownership erosion over time**

8. Community Signals Informing DP20

Across platforms and communities, recurring signals point to a shared breakdown between participation and ownership:

- maintainers and moderators demand compensation and decision power proportional to their labor

- skepticism toward “community-owned” claims following repeated governance failures and token collapses
- desire for portable community memory, identity, and governance tools
- frustration with value extraction from spaces built through collective effort
- growing interest in local, cooperative, and community-controlled digital environments

These signals are not isolated. They indicate structural gaps in how ownership is defined, enforced, and sustained.

DP20 treats these signals as design requirements, not feedback to be addressed after failure.

9. Non-Goals and Explicit Boundaries

DP20 does not:

- require all systems to be collectively owned
- eliminate private enterprise or hybrid ownership models
- guarantee equal distribution of value or influence
- remove the need for legal structures and compliance

DP20 defines the conditions under which ownership claims are legitimate and enforceable. It does not prescribe a single model.

10. Minimum Alignment (Non-Normative)

Minimum alignment defines the threshold where ownership is **real, enforceable, and not misleading**.

A DP20-aligned system **MUST**:

- bind ownership to executable governance and economic rights
- provide verifiable ownership records and history (DP15)
- ensure visibility into value flows and surplus allocation (DP17)
- enable credible exit and fork with continuity (DP4, DP7)
- maintain enforceable stewardship obligations

Failure modes to avoid:

- **ownership theater** (claims without control)
- **fork suppression** (no exit pathway)
- **surplus extraction without rights**
- **historical erasure of ownership changes**

Systems that omit execution, auditability, or exit **MUST NOT** be considered aligned.

11. Open Questions and Future Work

Key open questions include:

- how to design ownership models that function across jurisdictions
- how to balance tokenized and non-tokenized ownership structures
- how to maintain privacy while supporting interoperable ownership credentials (DP4)
- how to protect minority voices without blocking collective action
- how to measure contribution across visible and invisible labor (care, moderation, coordination)
- how to handle liability for collective decisions across legal systems
- how AI agents participate in ownership structures, if at all (DP11–DP13)

These questions reflect the boundary between social legitimacy and technical implementation.

12. Relationship to Other Desirable Properties

DP20 anchors ownership within the broader meta-layer system.

- DP3 defines how governance evolves under ownership
- DP4 constrains how collectively generated data can be used
- DP5–DP7 enable portability of identity, assets, and ownership artifacts
- DP6 and DP9 determine how value flows and incentives interact with ownership
- DP12 ensures ownership-linked governance can execute in practice
- DP15 ensures ownership claims and actions are provable
- DP17 ensures ownership structures can sustain themselves financially
- DP18–DP19 provide the participation and reputation inputs that mature into ownership

DP20 binds these properties into a coherent model of collective power.

13. Foresight and Failure Design

DP20 assumes ownership systems will be tested by capture attempts, economic pressure, and governance failure.

Common failure paths include:

- concentration of ownership among early or privileged actors
- erosion of rights through informal overrides or hidden control layers
- suppression of exit or fork pathways to preserve centralized power
- divergence between formal ownership and actual value capture

DP20 requires designing safeguards in advance, including:

- transparent allocation and reallocation mechanisms
- enforceable limits on concentrated control
- clear and executable fork and exit procedures
- public postmortems linking failures to structural changes

Failure is expected. Illegible failure is not.

14. Path Toward ML-RFC

Advancing DP20 toward ML-RFC requires:

- standardizing ownership object and receipt formats
- developing reference implementations of community-owned systems with open accounting
- aligning ownership models with interoperability and identity standards
- testing fork and exit mechanisms in real communities
- integrating legal and cooperative expertise into design processes

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

15. Closing Orientation

DP20 is where the meta-layer turns participation into power.

It rejects the model where communities create value but do not control it.

When ownership is real, communities can govern, sustain, and evolve the systems they depend on.

When it is not, “community” remains a narrative layered over extraction.

Ownership is the difference between presence and permanence, between contribution and control.

DP20 is where the meta-layer stops confusing audience with citizenship.

Ownership is the difference between a group that is used and a group that endures, with memory, upside, and responsibility held together.

When DP20 is strong, people can build commons that last because the commons can belong to those who care for them.