DP2 – Participant Agency and Empowerment

Power to the Participant — the Meta-Layer puts participants, not platforms, in control of how they show up, interact, and shape their online experience.

Purpose of This Draft

This ML-Draft articulates Desirable Property 2 (DP2) as the Meta-Layer's commitment that participants can meaningfully steer their digital lives. Beyond authentication (DP1) and governance (DP3), DP2 establishes that people and accountable agents hold real, usable power over presence, data flows, automation, and the conditions under which they are seen, acted upon, and counted.

DP2 responds to recurring failures of the contemporary Web:

- **Agency theater**: settings and consent flows that do not change outcomes
- **Asymmetric literacy**: systems legible only to specialists while obligations bind everyone
- **Structural dependency**: exit and portability exist nominally but are costly or illusory
- **Delegated opacity**: agents act on a participant's behalf without durable, legible control

This draft guides implementation, governance design, and future ML-RFC development. It is exploratory scaffolding, not a finalized specification.

Problem Statement

Why "control" without capability fails.

For decades, platforms have described participants as "in control" while reserving decisive power for operators, opaque ranking systems, and unbounded automation. The result is not merely dissatisfaction; it is predictable harm: manipulation, lock-in, surveillance-by-default, and governance that responds to scale by narrowing what ordinary people can do or understand.

DP2 begins from a different premise: **agency is not a feeling; it is a property of systems**. A Meta-Layer earns the label human-first only if participants can **observe, redirect, and withdraw** from the forces that shape their experience—within the same zones where accountability (DP1) is enforced.

Agency vs. Authorization

Authorization answers what a token allows. Agency answers whether a participant can **shape outcomes**: defaults, reach, automation, data use, and the rules that allocate visibility and risk.

Systems that conflate "logged in" with "empowered" routinely:

- Bundle consequential defaults behind "agree to continue"
- Hide material changes behind versioned policies
- Route impactful decisions to models or pipelines participants cannot inspect

DP2 separates authentication and authorization (DP1) from **participant-directed configuration of the lived interface**.

Empowerment as Distributed Capability

Empowerment is **capability + legibility + recourse**:

- **Capability**: participants can change outcomes (not just preferences)
- **Legibility**: participants can see how systems act on their behalf
- **Recourse**: participants can contest, reverse, or exit

A system lacking any one of these is not empowering, regardless of interface polish.

Threats and Failure Modes

Agency fails in patterned ways. The failure modes below are named throughout this draft as consequences of specific mechanism gaps; collected here, they form the adversarial model against which a DP2 implementation should be tested. None of them requires a malicious operator. Most emerge from ordinary optimization pressure, interface convenience, or the accumulation of integrations over time.

Agency theater

Interfaces expose settings that do not alter execution. The participant experiences choice; the system behaves identically.

Example: A "limit data use for personalization" toggle changes a display preference but not the ranking pipeline that consumes the same signals.

Why this matters: Simulated control is worse than absent control, because it suppresses the demand for real control.

Delegation drift

Agents act beyond the scope they were granted, either by accumulating permissions incrementally or by interpreting a mandate expansively.

Example: An assistant granted read access to a calendar begins composing and sending replies on the participant's behalf after a capability update.

Why this matters: Scope that is not continuously enforced is not scope; it is a suggestion.

Consent decay

Permissions persist beyond the participant's awareness, becoming "zombie consent" that no one remembers granting and no interface surfaces.

Example: A third-party integration authorized years earlier retains ongoing access with no expiry, review prompt, or visible record.

Why this matters: Consent that cannot be recalled cannot be revoked in any meaningful sense.

Coercive configuration

Defaults, flow design, and asymmetric friction steer participants toward choices that disadvantage them.

Example: Signup completes in one tap; cancellation requires locating a buried page, confirming twice, and waiting for an email.

Why this matters: Where entry and exit have unequal friction, the system has taken a position against the participant.

Automation overrun

Agent activity exceeds human capacity to observe, intervene, or revoke, nullifying agency without ever formally removing it.

Example: Delegated agents transact hundreds of times per hour; the participant's kill switch works, but only after the consequential actions have already settled.

Why this matters: Oversight that cannot keep pace with execution is not oversight.

Exit obstruction and degraded export

Leaving is technically permitted but practically infeasible, or the exported artifact is unusable elsewhere.

Example: An archive arrives containing content but no thread structure, permissions history, or stable identifiers.

Why this matters: Exit is the backstop for every other agency guarantee. When it fails, all remaining controls are granted at the operator's discretion.

Interop deception

Portability or integration is advertised, but core agency properties are silently lost in transit.

Example: A migration preserves posts but drops delegation states, so agents authorized in the source system silently gain or lose authority in the destination.

Why this matters: Undisclosed degradation is indistinguishable from misrepresentation.

Agency fragmentation and semantic drift

Participant choices do not persist across systems, or the same permission is interpreted differently in a new context.

Example: A "no automated action without confirmation" preference is honored in one tool and treated as advisory by a downstream integration.

Why this matters: Agency that stops at a system boundary is agency the participant cannot rely on.

Agency opacity

Participants cannot reconstruct what was done on their behalf, when, or under what authority.

Example: An automated decision changed a participant's visibility, but no log, summary, or attribution is available to review or contest.

Why this matters: Without auditability there is no recourse, and without recourse there is no agency.

Capture and collective overreach

Community-level mechanisms either concentrate power in a small group or suppress legitimate individual choices without pathways for challenge.

Example: A stewardship role accumulates permanent authority; or a zone-level setting silently overrides members' individual privacy configurations.

Why this matters: Collective agency must enhance individual agency, not substitute for it.

Core Principle

Agency is the ability to change outcomes, not merely configure preferences. Systems that do not preserve participant intent across automation, delegation, and scale do not provide agency.

Participant agency in the Meta-Layer is the combination of meaningful defaults, legible automation, durable delegation controls, and practical exit—enacted at the interface where people actually live.

Implications:

- Defaults favor the participant where stakes are asymmetric (data, reach, automation, payments), with zone-level calibration
- Automation is always **scoped**: time-bounded, purpose-limited, revocable, and attributable (DP1)

- Core agency paths (privacy, notifications, delegation, export, exit) remain reachable without specialized training
- **Collective agency is first-class**: communities can set norms and enforce them without stripping member autonomy (handoff to DP3, DP18–DP20)

Primary Mechanisms and Structural Conditions

DP2 is enacted through mechanism families that together convert stated control into enforceable control. Each addresses a distinct point at which agency is typically lost.

- **Presence and visibility controls** treat identity plurality and reach as configurable objects rather than fixed profile properties, so that pseudonymous participation is not defeated by accidental correlation or unsolicited amplification.
- **Default and friction design** assigns normative weight to what the system does when the participant does nothing, and distinguishes protective friction from friction engineered to obstruct understanding or exit.
- **The agency system layer** carries continuity, delegation integrity, consent durability, anti-coercion, cross-system semantics, and auditability across environments and time, so that participant intent survives movement.
- **Agency substrates** — purpose-limited processing, the agent delegation graph, and human-in-the-loop gradients — bind data use and automation to declared scope with a reachable interrupt.
- **Portability and exit guarantees** make leaving feasible in human time, with truthful disclosure of what is preserved, what degrades, and what cannot transfer.
- **Collective agency tools** allow communities to set and enforce norms through governed processes without silently nullifying individual controls.

The structural condition uniting these is *enforceability under movement and scale*. A control that holds in a single interface but fails on delegation, integration, or migration is not an agency mechanism; it is a display. DP2 therefore requires that every control either persist across a boundary or signal, at the boundary, that it no longer holds.

Three tensions are inherent to this design and are addressed rather than resolved: usability against configurability, automation against control, and safety against paternalism. These are treated first, because every mechanism that follows is a position taken within them.

Tensions and Tradeoffs

Usability vs. Complexity

Agency introduces configuration surfaces that can overwhelm. Hiding them removes control. DP2 requires **graduated disclosure**: simple defaults that are safe, with deeper controls accessible without specialized expertise.

Automation vs. Control

Automation reduces effort but can displace agency. Participants must be able to answer:

- What did the agent infer?
- What authority does it have?
- How do I stop it?

DP2 requires **visible delegation scopes, renewal, and revocation** aligned with accountable binding (DP1).

Power-Law Attention Markets

Even fair rules can reproduce inequality when attention is the currency. DP2 does not promise equal outcomes; it guarantees **equal access to the levers** that govern one's participation and visibility within a zone, and **transparent disclosure** when algorithmic allocation is in play (touchpoint DP14).

Safety vs. Patronizing Lockdown

Safety work can slide into infantilizing participants. DP2 pairs with DP1 to require that constraints be **proportionate, explainable, and contestable**, with pathways for competent self-determination inside high-trust zones.

Presence, Identity Plurality, and the Right to Shape Visibility

DP2 treats presence as something participants **sculpt**, not merely a profile object.

Plural Identities, Singular Accountability

Participants may present differently across zones (DP1). Agency requires **per-zone controls** for visibility, linkage, and discoverability so pseudonymous participation is not undermined by accidental correlation.

Reach and Amplification as Explicit Objects

When systems can amplify (boost, recommend, cross-post), amplification settings are **agency-bearing surfaces**: who may amplify me, under what proofs, with what caps? This is where DP2 meets DP1's asymmetric constraints for AI scale.

Defaults, Friction, and "Reasonable Participant" Design

Dangerous Defaults Are a Governance Bug

DP2 assigns normative weight to default selection: the burden of proof lies on whoever proposes a default that increases extraction, surveillance, or irreversible commitment.

Friction as Protection, Not Punishment

Strategic friction (confirmations, cooling-off periods for irreversible acts) protects agency when stakes are high. DP2 distinguishes **protective friction** from **hostile friction** designed to prevent exit or understanding.

Progressive Disclosure Without Burial

Advanced controls may be layered, but never removed from accountability: search, assistive onboarding, and machine-readable policy summaries are part of agency infrastructure.

Agency System Layer: Continuity, Delegation Integrity, and Enforceable Consent

Beyond interface controls and defaults, DP2 requires a coherent agency system layer that persists across environments, interactions, and time. This layer ensures that participant intent, consent, and control remain enforceable under scale, automation, and interoperability.

Agency is not simply the presence of controls. It is the ability to reliably change outcomes across systems without loss of intent, visibility, or recourse.

Agency Continuity Across Systems

Participant choices must persist across tools, zones, and integrations.

This requires:

- Preservation of consent, preferences, and delegation states across environments
- Explicit signaling when agency guarantees degrade or reset
- Protection against silent override of participant intent by downstream systems

A failure mode is **agency fragmentation**, where participant control is lost when moving across systems.

Delegation Integrity and Scope Enforcement

Delegation must remain bounded, legible, and enforceable.

All delegated authority must be:

- Explicitly granted
- Scoped by capability, domain, and time
- Continuously visible to the participant
- Revocable without friction

Systems must prevent delegated agents from expanding scope beyond granted authority.

A failure mode is **delegation drift**, where agents act beyond intended scope without detection.

Consent Durability and Revocability

Consent must persist long enough to be meaningful, but remain revocable at all times.

This requires:

- Clear mapping between consent and system behavior
- Immediate or bounded-time revocation pathways
- Prevention of "zombie consent" where permissions persist beyond participant awareness

A failure mode is **consent decay**, where participants lose track of what they have authorized.

Anti-Coercion and Default Integrity

Defaults must not be used to extract consent or steer behavior against participant interests.

Systems must:

- Prevent dark patterns and coercive flows
- Require explicit confirmation for high-impact or irreversible actions
- Surface when defaults materially affect outcomes

A failure mode is **coercive configuration**, where participants are nudged into decisions that undermine agency.

Cross-System Agency Semantics

Agency signals do not carry identical meaning across all systems.

Systems must:

- Signal when consent or delegation semantics change across contexts

- Prevent misinterpretation of permissions
- Allow participants to re-evaluate choices when entering new environments

A failure mode is **semantic drift**, where participant intent is misapplied across systems.

Agency Memory and Auditability

Participants must be able to reconstruct what they authorized, when, and why.

This includes:

- Logs of delegation, consent, and revocation events
- Visibility into system actions taken on behalf of the participant
- Tools for auditing past decisions and their consequences

A failure mode is **agency opacity**, where participants cannot understand or audit system behavior.

This agency system layer ensures that participant control is not an illusion created by interface design, but a durable property that persists under real-world conditions.

Data, Automation, and Delegation: Agency Substrates

Purpose-Limited Processing

Collection and use are tied to **stated purposes with granular switches**, not monolithic "privacy" toggles (deep coupling to DP4).

Agent Delegation Graph

For any automated or AI-mediated actor operating with participant intent, the system exposes:

- **Scope** (read/write domains, rate, spend limits where relevant)
- **TTL and renewal**
- **Attribution** to a responsible entity (see DP1, *Binding AI Outputs to Responsible Entities*)
- A **kill switch** reachable from the primary interface layer

Human-in-the-Loop Gradients

Not every action needs a click, but **material actions** (payments, legal commitments, public attributions, irreversible posts) require explicit human confirmation unless a community zone defines a higher-automation norm with informed opt-in.

Systems MUST remain safe under automated delegation at scale. This includes resisting coordinated agent behavior, preventing silent escalation of authority, and ensuring that human override remains effective even under high-volume automated activity.

A failure mode is **automation overrun**, where agent activity exceeds human capacity to observe, intervene, or revoke, effectively nullifying participant agency.

Portability, Exit, and Interoperability as Agency Guarantees

Agency must survive movement. If a participant's control disappears at boundaries, the system is coercive by design.

Practical Exit

Exit must be feasible in **human time** (hours or days for standard data classes). Stalling tactics, hidden dependencies, or degrading exports constitute agency violations.

Systems MUST:

- Provide complete, machine-readable exports for user-held data and artifacts
- Disclose exclusions (e.g., third-party licensed data) with clear rationale
- Preserve identity continuity signals where technically honest (DP1), or explicitly signal loss

Failure modes:

- **Exit obstruction**: artificial friction prevents leaving
- **Degraded export**: data is technically exported but unusable

Forking and Continuity

Where communities fork norms or stacks (see DP1, *Exit, Fork, and Kill Switches*), participants retain identity continuity and portable artifacts where technically honest, avoiding punishment for disagreement.

Systems SHOULD support:

- Portable credentials and attestations with provenance
- Migration guides and compatibility layers for common formats

Failure mode: **fork penalty**, where dissent results in loss of history or access.

Interoperability Honesty

Interoperability claims MUST be truthful. If a system advertises portability or integration, it MUST specify:

- What is preserved (data, credentials, delegation states)
- What degrades (semantics, guarantees, rate limits)
- What is not transferable and why

Failure mode: **interop deception**, where portability is claimed but core agency properties are lost in transit.

Collective Agency and Community Tools

Individuals act within communities. DP2 requires that collective mechanisms enhance, rather than erase, individual agency.

Communities MUST be able to:

- Define and publish mandates for stewardship roles (moderators, curators, treasurers)
- Set and revise community-level switches (e.g., human proof for governance votes) via governed processes (DP3, DP12)
- Audit decisions and reverse them through defined procedures

Systems MUST ensure:

- **Mandate transparency**: who can act, within what scope, for how long
- **Revocation pathways**: members can challenge or replace stewards
- **Proportional constraints**: community rules do not silently nullify individual controls

Failure modes:

- **Capture**: small groups entrench power and override member agency
- **Collective overreach**: community settings suppress legitimate individual choices without recourse

---

Governance, Accountability, and Agency Surfaces

DP2 is unusual among the Desirable Properties in that agency surfaces are not a supporting requirement but the property itself. A backend that faithfully enforces scoped delegation while exposing no way to see or change that scope has satisfied nothing. The condition DP2 imposes is therefore that the levers exist, that they are reachable without specialized expertise, and that pulling them demonstrably changes system behavior.

Participants must be able to:

- see, in one place, every active delegation: which agent, what scope, granted when, expiring when, acting on whose responsibility
- revoke any delegation and observe the effect within a bounded, stated time
- see what has been done on their behalf, with enough context to contest it
- see which defaults currently apply to them and which of those defaults materially affect data use, reach, or cost

- distinguish a protective confirmation from an obstruction, because the system states which stakes triggered the friction
- initiate export and exit, see what the export will and will not contain before requesting it, and receive it within human time
- understand, at a boundary, which of their choices will persist into the new context and which will not

Communities must be able to:

- publish stewardship mandates with scope and duration, and revise them through governed process rather than accretion
- set zone-level agency floors — for example, requiring confirmation for irreversible acts — that individual settings may strengthen but not silently weaken
- audit whether community-level switches are nullifying individual controls, and reverse them where they are
- preserve memory of why an agency rule was adopted and what it was responding to (DP3)
- retain the ability to fork or exit collectively, carrying member artifacts and continuity where technically honest

Example: A participant opens a single delegation view, sees that a shopping agent holds a spend limit expiring in nine days and a summarization agent with indefinite read access, revokes the second, and receives confirmation that the change took effect along with a log of what that agent had previously done.

Without these surfaces, the failure mode is **agency by assertion**, where a system's claims about participant control cannot be verified, adjusted, or contested by the participants those claims concern.

Incentives and Power Analysis

Agency is expensive to provide and profitable to withhold. Most agency failures are not the product of hostile design decisions but of ordinary optimization: the default that converts better ships, the export that no one is measured on degrades, the delegation scope that reduces friction expands.

The recurring pressures are:

- **Defaults as revenue.** Default selection is among the highest-leverage economic decisions a system makes. This is why DP2 places the burden of proof on whoever proposes a default that increases extraction, surveillance, or irreversible commitment, rather than treating defaults as neutral engineering choices.

- **Friction asymmetry as retention.** Where growth is measured and churn is penalized, signup friction is optimized away and cancellation friction is not. Parity between entry and exit is an incentive correction, not a courtesy.
- **Export as an unowned cost.** Portability generates no revenue and creates competitive risk, so exports degrade by neglect rather than intent. DP2 therefore requires that export usability be demonstrated in an independent system, not merely offered.
- **Automation as scope accumulation.** Broader agent authority produces better outcomes on most metrics, so delegation scope drifts outward absent enforcement. Time-bounding and continuous visibility exist to make drift visible rather than trusting restraint.
- **Legibility as competitive exposure.** Explaining how a system acts on a participant's behalf reveals mechanisms operators would prefer to keep private. DP2 resolves this by requiring participant-legible explanation and audit paths rather than full internal disclosure.
- **Attention markets as concentration.** Where reach is the scarce good, fair rules still produce unequal outcomes. DP2 does not promise equal reach; it insists that access to the levers governing one's own visibility be equal within a zone.

Example: A system honors revocation faithfully but places the delegation view four levels deep, unlinked from any surface where agents actually act. No rule was broken; the lever was priced out of reach.

Why this matters: Power in agency systems accrues to whoever controls placement, default, and pace. DP2 treats those three as governed surfaces (DP3, DP12) precisely because they determine whether the formal guarantees are usable.

Community Signals Informing DP2

Recurring themes from public discourse (non-exhaustive) include both desires and tensions:

- Fatigue with consent theater and unreadable policies
- Demand for "show me what you're doing with my data right now" views
- Preference for AI copilots that **ask before acting** on the user's behalf
- Interest in portable reputation without panopticon scoring (tension with DP1)
- Desire for simplicity alongside real control (tension with complexity)

These signals reveal a core contradiction: participants want **power without overload**. DP2 addresses this through progressive disclosure, safe defaults, and auditability, rather than removing control.

Foresight and Failure Design

DP2 assumes that agency will erode rather than break. Systems rarely remove controls; they let controls fall out of correspondence with behavior as pipelines are added, agents gain capability, and integrations multiply. The characteristic DP2 failure is a system whose settings page is accurate on the day it shipped and increasingly fictional thereafter.

Predictable degradation paths include:

- **Control drift**, where new processing pathways are added without being bound to existing participant switches, so the same toggle governs a shrinking share of actual behavior
- **Scope creep through capability updates**, where an agent's authority effectively expands because its tools expanded, without any new grant
- **Consent accumulation**, where authorizations pile up faster than any review cadence, until the delegation surface is too large to audit
- **Export decay**, where schema changes gradually strip meaning from an export that no one tests against an independent importer
- **Automation outpacing oversight**, where the interrupt still works but the actions it interrupts are already consequential
- **Collective substitution**, where community-level settings progressively absorb decisions that were previously individual, with no moment at which the change was contested

These paths compound. Scope creep raises the volume of automated action, which raises the audit burden, which reduces the probability that drift is noticed at all.

DP2 therefore requires safeguards designed in advance:

- **Delegation expiry by default**, so that inaction shrinks authority rather than preserving it
- **Consent review cadence**, surfacing accumulated grants at intervals rather than only on request
- **Boundary signaling**, so that degradation of agency guarantees is announced at the crossing rather than discovered afterward
- **Reachable interrupts** whose latency is stated and tested under load, not assumed
- **Export conformance testing** against at least one independent system, treated as a recurring obligation rather than a launch milestone
- **Postmortems for agency incidents** — coerced consent, automation overrun, failed revocation — linked to the default or scope change that produced them

Failure is expected. Silent drift is not. A DP2-aligned system is one where the gap between what the controls claim and what the system does is observable while it is still small.

Relationship to Other Desirable Properties

- **DP1** supplies accountable actors; **DP2** supplies the levers those actors hold. Without DP1, agency collapses into anonymity games; without DP2, accountability becomes surveillance.
- **DP3** scales governance; DP2 ensures scale does not erase participatory steering.
- **DP4–DP6** ground agency in data, namespace, and commerce realities.
- **DP7–DP10, DP21** carry agency into experience, education, and modality.
- **DP11–DP13** bound AI so delegation does not swallow human steering.
- **DP14–DP17** make power auditable and sustainable.
- **DP18–DP20** turn agency into shared evolution of the stack.

Several of these couplings are load-bearing rather than thematic:

- **DP4** and DP2 share the consent stack. DP4 governs what may be collected, inferred, and retained; DP2 governs whether the participant can actually change those terms and observe the change. Purpose-limited processing is a DP4 mechanism exercised through a DP2 surface.
- **DP5** determines whether plural presence is possible at all. Per-zone visibility controls depend on identifiers that can remain distinct without being correlated by shared infrastructure.
- **DP6** makes economic surfaces agency-bearing: spend limits, subscription reversal, and agent budget mandates are DP2 controls enforced in a DP6 context.
- **DP7** determines whether agency survives movement. Interoperability honesty is the DP2-facing obligation that DP7 portability claims must satisfy.
- **DP13** provides the containment that makes delegation safe at scale. A kill switch is a DP2 affordance; the guarantee that it halts execution is a DP13 property.
- **DP20** is the final backstop: if the stack itself cannot be forked or collectively owned, every agency guarantee remains a grant that can be withdrawn.

Non-Goals and Explicit Boundaries

DP2 defines the conditions for agency; it does not promise universal outcomes.

DP2 does not:

- Guarantee equal outcomes or neutralize attention economics by fiat
- Eliminate all paternalistic protections in safety-critical zones (these must be labeled, bounded, and contestable)
- Replace DP1 accountability with unchecked "freedom to harm"

- Mandate a single UX globally; pluralism across zones is expected

DP2 also does not:

- Require full transparency of all system internals where doing so would enable exploitation; instead, it requires **participant-legible explanations and audit paths**
- Allow delegation to obscure responsibility; all automated action remains attributable (DP1)

Failure mode: **overreach**, where DP2 is interpreted to justify unsafe or unaccountable behavior.

Minimum DP2 Alignment (Non-Normative)

Minimum alignment is not a UX checklist. It is the threshold at which participant agency is **real, enforceable, and resistant to coercion, drift, and automation capture**.

A system that does not meet these conditions may expose controls, but it does not provide agency.

At minimum, a system claiming DP2 alignment MUST satisfy the following **irreducible conditions**:

Outcome-Level Control (Not Preference Simulation)

- Participants MUST be able to change meaningful outcomes, not only surface preferences
- Core levers (visibility, data use, automation authority, exit) MUST directly affect system behavior
- Systems MUST NOT simulate control through settings that do not alter execution

Failure mode: **agency theater**, where interfaces imply control without changing outcomes.

Delegation Visibility and Revocation

- All automated or AI-mediated actions MUST be attributable and visible to the participant
- Delegation MUST include scope, duration, and authority limits
- Participants MUST be able to revoke delegation in real time or bounded time

Failure mode: **delegation opacity**, where systems act without legible authority or revocation.

Consent Binding and Enforcement

- Consent MUST be explicitly tied to system behavior
- Systems MUST enforce consent boundaries consistently across components and integrations
- Revocation MUST propagate across systems or clearly signal where it does not

Failure mode: **consent bypass**, where downstream systems ignore or reinterpret user intent.

Anti-Coercion Defaults

- Defaults MUST NOT materially disadvantage participants without explicit opt-in
- High-impact actions MUST require clear confirmation
- Systems MUST NOT use dark patterns to obtain or retain consent

Failure mode: **coerced consent**, where participants are steered into decisions against their interest.

Practical Exit and Portability

- Participants MUST be able to export their data and exit systems in reasonable human time
- Exit MUST NOT result in silent loss of identity continuity without explicit signaling (DP1)
- Systems MUST NOT impose artificial friction to prevent exit

Failure mode: **exit obstruction**, where users are technically allowed but practically unable to leave.

Cross-System Agency Integrity

- Participant choices MUST persist across integrations where technically feasible
- Systems MUST signal when agency guarantees degrade across contexts
- Downstream systems MUST NOT silently override upstream participant intent

Failure mode: **agency fragmentation**, where control is lost across system boundaries.

Auditability of System Behavior

- Participants MUST be able to inspect what actions were taken on their behalf
- Systems MUST provide logs or summaries of automated decisions and their effects
- Critical actions MUST be reconstructable for dispute or review

Failure mode: **agency opacity**, where participants cannot understand or challenge system behavior.

These conditions define the **minimum viable agency layer** of the Meta-Layer.

Partial implementations that omit outcome control, delegation integrity, consent enforcement, or exit MUST NOT be considered aligned with DP2.

Open Questions and Future Work

- Portability vs. abuse: preventing weaponized export while honoring exit (interfaces with DP1 memory models)
- Legibility budgets: how much system behavior can be made comprehensible without overload; role of machine summaries vs. audits (DP14–DP15)
- Collective overrides: when may a community limit individual agency for safety without capture?
- Cross-zone identity correlation: separation vs. pressure for unified reputation
- Economic agency: tipping, subscriptions, and paid reach as agency-bearing surfaces (touchpoint DP6)

Further questions concern how agency is expressed, timed, and measured at the edges of a system:

- how to express delegation scope in terms participants can evaluate, given that capability boundaries are natural to engineers and opaque to nearly everyone else
- what revocation latency is acceptable for which action classes, and how that latency should be stated rather than left implicit
- how to keep an agency audit trail useful without turning it into a surveillance record of the participant's own behavior (tension with DP4)
- how to detect control drift automatically, so that a new processing pathway unbound to any existing switch is flagged rather than silently shipped
- how agency guarantees should degrade when a delegated agent operates in a system that does not implement DP2 at all
- how to measure agency health at the ecosystem level — revocation success rates, export usability, default asymmetry — without reducing it to a compliance score

---

Path Toward ML-RFC

Advancement from ML-Draft to ML-RFC should demonstrate that agency is not only described but **operationally verified**.

Key steps:

- Community review with builder and civil-society lenses
- Reference implementations of delegation graphs, revocation, and exit flows
- Conformance tests for **minimum alignment** (see *Minimum DP2 Alignment*) including adversarial scenarios (automation overrun, consent bypass, interop deception)
- Clear separation of **normative invariants** vs **design space**
- Publication of audit reports demonstrating real-world behavior under load

Graduation criteria SHOULD include:

- Evidence that participants can revoke delegation and observe effect within bounded time
- Evidence that exports are usable in at least one independent system
- Evidence that automated actions are attributable and auditable end-to-end

---

Closing Orientation

DP2 asserts that participants are not merely subjects of systems, but operators within them.

When agency is real, people can:

- change outcomes
- understand system behavior
- delegate safely
- exit without penalty

When agency is simulated, systems accumulate hidden power: defaults decide, automation acts, and participants absorb the consequences.

DP2 is the commitment that control is not inferred, but **demonstrable**.

It is the difference between having settings and having a steering wheel.

DP1 asks who may act with integrity. DP2 asks whether participants hold the wheel—or only the liability.
