

Report Abuse at Scale: Making an Accusation Cost the Accuser

A coordinated reporting campaign works because pressing the report button is free and answering the report is not. Discord, as publicly described, offers in-app controls for reporting messages, users, and servers, and routes those submissions to a trust and safety function that applies the platform's published rules. That is an operator-mediated design, and it is the right shape for a platform that owns both ends of the exchange. The architecture disclosed in U.S. Provisional Application No. 64/117,812 addresses a different setting, one with no operator in the middle, where two autonomous agents write into each other's governance state directly, and it prices the write against the party making it rather than adjudicating whether the write is true.

A Thousand Complaints Against One Account

An account wakes up to a wall of reports. Each is well-formed, each cites a real category, and none came from anyone the account has interacted with. Somewhere behind them is a group chat that agreed to file. Filing cost one tap per person. Answering is a review obligation, and the cost of being wrong falls on the reported party while the queue drains.

This is the structural fact underneath every report button. It is a write channel into somebody else's standing, free at the point of use, and a report aimed at a thousand parties costs the same per unit as a report aimed at one.

The obvious defenses each run into something. Per-account limits meter accounts, and accounts are cheap to create. Sanctioning false reports is principled, but applying a sanction means first ruling the report meritless, and rulings are the scarcest thing in the pipeline.

Autonomous software raises the stakes rather than changing them. Filing rate used to be bounded by a person at a form. It is now bounded by a request loop, and software instructed to defend its principal's interests will file at machine cadence, because filing costs it nothing.

What Discord's Reporting Path Is Built to Do

Discord operates a communication platform organized around servers and channels, where, as publicly described, a good deal of day-to-day governance sits close to the community itself. Owners and moderators hold tools over their own spaces, and the platform's rules sit on top of that. Also as publicly described, Discord provides in-app controls that let a user report a message, a user, or a server, with categories that direct the submission toward the kind of harm alleged. Submissions reach a trust and safety function that reviews reported content against the platform's published rules and applies enforcement outcomes ranging from measures on a single piece of content up to actions against accounts and spaces. Discord's public materials also describe periodic transparency reporting and a route by which users can appeal enforcement decisions.

Read on its own terms, this is a coherent and well-matched design. The operator holds the identity system, the content, the enforcement surface, and the appeals surface. When one party holds all four, the sensible place for judgment is the middle, where it

can see the whole exchange, weigh context, and correct itself later. That model is what lets enforcement stay consistent at platform scale.

Nothing below is a claim about how well any of that works, or about internal implementation. The comparison is between two architectural settings, not between two products.

Metering the Party Who Asserts

U.S. Provisional Application No. 64/117,812 describes a semantic agent (100) that issues conduct evaluation artifacts (116) concerning conduct performed by other agents, where the issuance is priced against the authorization budget (404) of the issuing agent. The filing frames this as a mirror. An agent that refuses an execution pays a metered cost for the refusal, and an agent that asserts against a counterparty pays a metered cost for the assertion, in each case without any adjudication of the merit of the refusal or of the assertion. Both directions are priced from one quantity held by the party taking the action.

Pricing runs through an assertion-cost counter (400) carried in the agent's memory field (102). Its fields include an issuance accumulator counting increments within a metering window declared in the signed policy object (112), a per-recipient-class register recording whether an origin-equivalence class (200) of receiving parties has already contributed an increment in the current window, a budget floor field, an epoch reference, and a decrement schedule specifying an amount greater than zero.

What matters for flooding is what the register counts. On issuance, the agent computes the origin-equivalence class (200) of the receiving party. Where the register records that the class has already contributed an increment within the current window, no increment is applied. Where it does not, the accumulator increments by one and the register is updated. Responsive to the increment, the authorization budget (404) is decremented by the scheduled amount, and the filing states that the units of the

decrement and the units gating dispatch of actions by the issuing agent are identical. Breadth of accusation is what costs, and it costs in the same currency the agent needs in order to act.

That decrement is not conditioned on merit. Per the filing, an artifact later resolved to the accepted determination (122), the rejected determination (124), the not-determinable determination (126), or the not-applicable determination (128) bears one and the same decrement, and the architecture computes no adjudication of the truth of the conduct alleged as a condition of that decrement.

Consequences at the floor are stated conditionally in the filing. Responsive to the authorization budget (404) satisfying the floor held in the budget floor field, the authorization gate (300) is written to the withheld state (310) for an enumerated set of action classes, and the agent transitions into the non-executing cognitive mode (302) with respect to those classes. The filing is specific that the faculty withheld is that of executing actions of the enumerated classes, and that the capacity to issue conduct evaluation artifacts (116) is not itself withheld by the write. While the budget stands at or below the floor, the agent attaches no attestation of the counter state to artifacts it issues.

Attestation is the receiving side of the design. On issuance the agent advances its dynamic agent hash chain to a successor epoch and attests, within the artifact, the accumulator state and that epoch identifier. A receiving agent verifies, as a precondition to admitting the artifact to its admission evaluator (120), that the attested epoch identifier is a valid successor of one it previously recorded for that issuer in its counterparty identity record (114), and that the attested accumulator state is not less than a state previously recorded there. An artifact with no attestation, with a non-successor epoch, or with a regressed accumulator state is appended to the receiver's append-only lineage field (104) and is not admitted. It produces no determination, moves no value of the scoped integrity vector (106), and increments no counter.

Two further guards close the obvious evasions. An issuer that over-splits its classification of receiving parties, so reducing its own increments, is detected when the receiver recomputes the assignment and appends a class-splitting divergence record naming both identifiers, after which the attestation is treated as one that does not verify. Receivers are protected from the inverse load by a receipt-verification register holding a per-class count of verifications within a declared window. Once a declared verification cap is reached, further artifacts from that class within the window are appended to lineage, not verified, not admitted, and append nothing adverse to the issuing party.

Replenishment is the last piece. The budget is replenished by receipt of an admitted conduct evaluation artifact (116) originating from an origin-equivalence class (200) absent from the replenishment register (402), by an amount from the budget replenishment schedule (406), bounded above by the authorization budget ceiling (408) with any excess discarded. Per the filing, that amount is less than the per-increment decrement, whereby an agent is incapable of financing a volume of issuance by the receipts its own issuance provokes, and the budget is not replenished by elapsed time, by expiry of a metering window, or by issuing further artifacts.

Two Layers, Not Two Answers

Both designs address the same failure, cheap accusations consuming expensive attention, and they place the cost in different places.

An operator-mediated path puts judgment in the middle. Someone weighs the report against published rules and context, decides, and can be appealed to. That produces outcomes a human can defend, and it is the shape that fits when one party owns identity, content, and enforcement.

The filed architecture presumes no such party. Its cost lands at issuance, on the issuer, before anyone has decided anything, and the receiving side's test is arithmetic on an attestation rather than an assessment of the allegation. The filing states that the mechanism does not depend upon voluntary compliance by the issuing agent and does not depend upon any party other than the two parties to the exchange.

Name the trade honestly. Merit-blind metering prices a true accusation exactly as it prices a false one. That is a deliberate property of the filed design, and a platform that has review capacity may reasonably decline the trade. An agent mesh with no operator has no review capacity to decline it with. The positioning is complementary rather than competitive.

Where Both Would Sit in One System

A plausible deployment keeps the human-facing report path exactly as it is. People reporting people is a judgment problem, and judgment belongs with the operator.

The metered layer suits traffic the disclosed architecture is built for: submissions from software acting on a principal's behalf, and agent-to-agent exchanges that never touch a user interface. There, an unattested submission, or one whose attestation fails verification, is filtered before it reaches a queue, and the filtering costs the receiver a register lookup rather than a reviewer.

The limits of the disclosed architecture deserve equal billing. It does not decide whether a report is true, since the filing is explicit that no adjudication of truth is computed as a condition of the decrement. It removes no content, and it produces no determination on the merits for anyone to appeal; the only restraint it describes falls on the issuing agent's own execution faculty. Quantities that would govern behavior in practice, among them the decrement schedule, the metering window, the budget floor, and the verification cap, are declared in a signed policy object, and the filing supplies values

only by way of illustration. It also requires participants that maintain a memory field, an append-only lineage field, and a dynamic agent hash chain, which describes a governed agent rather than a person with a phone.

These solve adjacent halves. One decides what should happen to a report. The other bounds how many reports one party can make matter.

Disclosure Scope

The architecture described here is disclosed in U.S. Provisional Application No. 64/117,812. Statements about the assertion-cost counter (400), the authorization budget (404), the replenishment register (402), the budget replenishment schedule (406), the authorization budget ceiling (408), and conduct evaluation artifacts (116) reflect that filing and no implementation.

References to large-platform abuse reporting systems are to public materials and are used for comparison only; no relationship, endorsement, or infringement is asserted. Descriptions of that product are qualitative, address purpose and category rather than internal implementation, and should be checked against the vendor's own current documentation. Nothing here characterizes any third-party system as practicing anything disclosed in the application, and nothing here is legal advice.

Assertion-Cost Symmetry (</assertion-cost-symmetry>) [All 49 steps → \(/inventive-steps\)](#)

Making an accusation costs the accuser too — symmetric, metered, replay-proof.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

- [Assertion-Cost Symmetry: Making an Accusation Cost the Accuser \(/articles/assertion-cost-symmetry/making-an-accusation-cost-the-accuser\)](/articles/assertion-cost-symmetry/making-an-accusation-cost-the-accuser)

SECONDARY TECHNICAL

- [Withheld Delegating Faculty, Preserved Direct Execution \(/articles/assertion-cost-symmetry/withheld-delegating-faculty\)](/articles/assertion-cost-symmetry/withheld-delegating-faculty)
- [Depth-Propagated Harm Inheritance Across a Delegation Chain \(/articles/assertion-cost-symmetry/depth-propagated-harm-inheritance\)](/articles/assertion-cost-symmetry/depth-propagated-harm-inheritance)
- [Co-Signature Conditioning of Restoration Authority \(/articles/assertion-cost-symmetry/co-signature-conditioned-restoration\)](/articles/assertion-cost-symmetry/co-signature-conditioned-restoration)
- [The Anti-Evasion Charge on Governance-Address Removal \(/articles/assertion-cost-symmetry/anti-evasion-governance-address-removal\)](/articles/assertion-cost-symmetry/anti-evasion-governance-address-removal)
- [The Progressive Issuance Decrement \(/articles/assertion-cost-symmetry/progressive-issuance-decrement\)](/articles/assertion-cost-symmetry/progressive-issuance-decrement)
- [Non-Attenuable Forward Restraint Inheritance: A Delegation Record That Cannot Strip an Agent's Abstention State \(/articles/assertion-cost-symmetry/non-attenuable-forward-restraint-inheritance\)](/articles/assertion-cost-symmetry/non-attenuable-forward-restraint-inheritance)
- [Holding Authorization When the Renewal Register Cannot Be Read: Re-Read Cadence, Held-Value Resume, and the Max-Hold Inquiry \(/articles/assertion-cost-symmetry/register-unavailable-re-read-cadence-held\)](/articles/assertion-cost-symmetry/register-unavailable-re-read-cadence-held)

APPLICATIONS • GENERAL

- [Report Abuse and Takedown Flooding: Making the Accusation Cost the Accuser \(/articles/assertion-cost-symmetry/takedown-and-report-abuse\)](/articles/assertion-cost-symmetry/takedown-and-report-abuse)
- [The Economics of Agent-to-Agent Disputes \(/articles/assertion-cost-symmetry/agent-to-agent-dispute-economics\)](/articles/assertion-cost-symmetry/agent-to-agent-dispute-economics)

APPLICATIONS • SPECIFIC

- [DMCA Notice and Takedown: Where the Cost of an Assertion Falls \(/articles/assertion-cost-symmetry/dmca-notice-and-takedown\)](/articles/assertion-cost-symmetry/dmca-notice-and-takedown)
- [Report Abuse at Scale: Making an Accusation Cost the Accuser \(/articles/assertion-cost-symmetry/platform-abuse-reporting\)](/articles/assertion-cost-symmetry/platform-abuse-reporting)

TERMINOLOGY

- [acknowledgment artifact \(/glossary#acknowledgment-artifact\)](/glossary#acknowledgment-artifact)

- [assertion-cost counter \(/glossary#assertion-cost-counter\)](/glossary#assertion-cost-counter)
- [authorization budget \(/glossary#authorization-budget\)](/glossary#authorization-budget)
- [dynamic agent hash chain \(/glossary#dynamic-agent-hash-chain\)](/glossary#dynamic-agent-hash-chain)

[Assertion-Cost Symmetry overview → \(/assertion-cost-symmetry\)](/assertion-cost-symmetry)