Nick Clark · Published July 29, 2026 · PDF

Sift: Abuse Decisioning and the Cost of an Assertion

Abuse decisioning platforms exist because online marketplaces, payment flows, and user-generated content systems generate more reports, signals, and risk events than any human review team can resolve. Sift is publicly described as a machine learning platform for digital trust and safety, scoring risk for the party under evaluation so an operator can decide what to allow. Chapter 4 of U.S. Provisional Application No. 64/117,812 addresses an adjacent question with a different subject: what the act of asserting misconduct costs the agent that makes the assertion. The filed chapter meters the asserting party through an assertion-cost counter and an authorization budget, clocked in hash chain epochs, barred against replay, verified at the point of receipt, and replenished only by exposure to counterparties not yet counted.

When the complaint is free and the defense is not

A seller on a marketplace wakes up to nine reports of counterfeit goods filed overnight against her listings. Each is text in a form field, and each enters a review queue, consumes analyst attention, and applies a provisional restriction while pending. Several come from competitors, and none cost the filer anything measurable.

That asymmetry is a structural property of most conduct systems rather than a flaw in any product: a write into another party's standing is cheap to originate and expensive to absorb. Scale it to software agents transacting at machine rates and it becomes an attack surface, because the receiving side pays more to evaluate an allegation than the asserter paid to produce it.

Abuse decisioning platforms address one side of this directly and well: determining, quickly and at volume, whether the party under scrutiny is likely doing something harmful. A separate design question sits on the asserting side, and the filed chapter below takes it up. What would an architecture have to do for the act of asserting to draw down something the asserter itself holds?

Sift in the trust and safety category

Sift is publicly described as a digital trust and safety platform applying machine learning to fraud and abuse decisioning for online businesses. Its publicly stated scope covers familiar families of abuse: payment fraud, account takeover, fake accounts, promotion and incentive abuse, and content abuse such as spam and scam postings. Public materials position the product for operators who need decisions in the flow of a transaction rather than after the fact.

At the level Sift describes publicly, the approach is signal-driven and model-driven. Signals drawn from user activity are scored by models and surfaced as risk assessments an operator acts on through configurable rules and workflows, with review tooling for cases that call for a human. Building coverage at that scope, keeping models current against adaptive adversaries, and exposing the result through operable decision logic is difficult engineering.

The resulting frame is worth stating precisely. A platform of this kind is engaged by the operator of a service, evaluates parties and events within that operator's environment, and returns a judgment the operator applies. The subject of the assessment is the party

being evaluated, which is the right framing for the problem the category was built to solve. Nothing below is a criticism of it.

Metering the party that writes into someone else's record

Chapter 4 of U.S. Provisional Application No. 64/117,812 takes the other subject: a semantic agent (100) that issues a conduct evaluation artifact (116) about conduct performed by another agent prices that issuance against its own authorization budget (404). Separate chapters govern the evaluated-party side and are not described here.

The metering instrument is an assertion-cost counter (400), carried in the issuing agent's memory field (102). An issuance accumulator counts increments within a metering window declared in the signed policy object (112) and expressed as a count of successor epochs of the agent's hash chain. A per-recipient-class register records whether a given origin-equivalence class (200) of receiving parties has already contributed an increment in that window. A budget floor field, an epoch reference, a budget reference, and a decrement schedule complete the structure.

Issuance follows an ordered procedure. The agent builds the artifact from its own identifier, a recorded assertion time, and a conduct descriptor drawn from its append-only lineage field (104). It computes the origin-equivalence class (200) of the receiving party and consults the register: a class already counted adds no increment, an uncounted class increments the accumulator by one. Responsive to that increment, the authorization budget (404) is decremented by the scheduled amount, in units identical to those gating the agent's own action dispatch, and no conversion is computed between an assertion-denominated quantity and an execution-denominated one. The agent then advances its hash chain to a successor epoch, attests the accumulator state and epoch identifier inside the artifact, and appends the issuance, the increment, the decrement, and the attested state to lineage.

The decrement is not conditioned on merit. 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 no adjudication of the truth of the allegation conditions the charge.

The metering interval is not a timer. Metering runs in successor epochs of the agent's hash chain, each generated from a prior epoch, an unpredictability contribution, and a volatile salt, so a successor epoch is not computable in advance by the issuing agent and is not computable at all by any other party. An attestation binds an accumulator state to a specific epoch, so artifacts attesting a common state carry a common epoch identifier, and a receiving agent holding a prior artifact bearing that identifier detects the repetition from its counterparty identity record (114). The filed text states the epoch identifier is not a wall-clock timestamp and is not drawn from a clock available to the hosting execution node, so advancement of the interval is not accelerated by manipulation of a clock.

Verification happens at the point of receipt. Before admitting an artifact to its admission evaluator (120), a receiving semantic agent (100) confirms that the attested epoch identifier is a valid successor of one previously recorded for that issuer, and that the attested accumulator state is not less than a state previously attested. An artifact failing either check, or arriving without an attestation, is appended to lineage and admitted to nothing: no determination, no movement of the scoped integrity vector (106), no counter incremented. The issuer also attests which class it assigned the recipient to, and a recipient whose recomputation finds one class split across distinct attested identifiers records a class-splitting divergence and treats the attestation as failing to verify.

Refill runs by one procedure and no other. The authorization budget (404) is replenished on receipt of an artifact admitted by the admission evaluator (120) and originating from an origin-equivalence class (200) absent from the agent's replenishment register (402). Elapsed time does not replenish it, expiry of a metering

window does not, and issuing further artifacts does not. The replenished value is bounded above by an authorization budget ceiling (408) declared in the signed policy object (112), excess is discarded rather than carried forward, and the replenishment amount is specified to be less than the per-increment decrement, whereby an agent is incapable of financing a volume of issuance out of the receipts its own issuance provokes.

Where the budget satisfies 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 enters the non-executing cognitive mode (302) for those classes. The withheld faculty is execution of those classes, not the capacity to issue conduct evaluation artifacts (116). While the budget stands at or below the floor the agent attaches no attestation to what it issues, and unattested artifacts are not admitted downstream. A later replenishment raising the budget above the floor returns the gate to the granting state for those enumerated classes and no others.

One illustration in the filing, offered as illustration and not as a recommended setting: with a declared decrement of one unit and a budget of forty units, an agent issuing to sixty parties assigned to sixty distinct origin-equivalence classes (200) in one window exhausts the budget at the fortieth increment, and the remaining twenty issuances carry no valid attestation. Had those sixty parties been assigned to three classes, the accumulator would have reached three.

Two different subjects of assessment

The difference comes down to whose record an accusation debits.

- **Subject.** A decisioning platform in Sift's category assesses the party under evaluation, from signals about that party's behavior. The filed chapter assesses the asserting party, from a counter that party carries itself.

- **Trigger.** The disclosed decrement applies at issuance, before adjudication, and attaches whichever of the four determinations the artifact resolves to.
- **Enforcement locus.** Attestation is verified at receipt against the recipient's own counterparty identity record (114). The filing states the mechanism depends on no party other than the two in the exchange.
- **Refill.** The budget refills only on exposure to an uncounted origin-equivalence class (200), capped by a declared ceiling, at an amount smaller than the charge.

These are complementary questions rather than competing answers. An operator still needs to know whether an account is committing payment fraud, and model-driven risk scoring of the kind Sift describes is aimed at that. The filed chapter addresses a different case, in which the accusing party is itself an agent whose capacity to generate accusations is not bounded by the effort of writing them.

Reading this in a live trust and safety stack

Agent-to-agent commerce is the setting where both matter. When the parties filing conduct reports are software agents, assertion volume is bounded only by compute, identity is cheap to multiply, and the receiving side pays the evaluation cost every time.

A decisioning platform of Sift's kind addresses the content of the allegation, scoring whether the alleged pattern looks like fraud or abuse. The filed mechanism addresses rate and reach, charging each newly counted class of recipient against the same budget that gates the issuer's ability to act. The origin-equivalence class (200) construction cuts both ways: asserting against a hundred parties within one class costs one increment, while an issuer that over-splits its classification to reduce its increments is what recipient-side recomputation is disclosed to detect.

Four questions are worth asking of any design in this area. Does an assertion debit anything the asserter holds? Does that debit survive the assertion turning out to be correct? Can the metering interval be advanced by anything the asserter controls? Does

the capacity to assert refill from elapsed time, or from something harder to manufacture? The filed chapter states an answer to each.

Disclosure Scope

This article describes subject matter disclosed in Chapter 4 of U.S. Provisional Application No. 64/117,812: the assertion-cost counter, the authorization budget and its ceiling, epoch metering, recipient-side verification of attested state, and replenishment on exposure to uncounted origin-equivalence classes. Other chapters govern other subjects and are not described here. The application is pending, and nothing here states that any claim has been allowed or that any scope of protection has been determined.

References to Sift are to public materials and are used for comparison only; no relationship, endorsement, or infringement is asserted.

Statements about Sift describe product purpose and category qualitatively and are not representations about internal implementation, performance, or roadmap. Readers should consult the vendor's current materials and the filed application text rather than this summary.

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)

SECONDARY TECHNICAL

- Withheld Delegating Faculty, Preserved Direct Execution (/articles/assertion-cost-symmetry/withheld-delegating-faculty)
- Depth-Propagated Harm Inheritance Across a Delegation Chain (/articles/assertion-cost-symmetry/depth-propagated-harm-inheritance)
- Co-Signature Conditioning of Restoration Authority (/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)
- The 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)
- 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)

APPLICATIONS · GENERAL

- Report Abuse and Takedown Flooding: Making the Accusation Cost the Accuser (/articles/assertion-cost-symmetry/takedown-and-report-abuse)
- The Economics of Agent-to-Agent Disputes (/articles/assertion-cost-symmetry/agent-to-agent-dispute-economics)
- Cloud SLA Credit Claims and the Cost of Asserting Downtime (/articles/assertion-cost-symmetry/cloud-sla-credit-claims)

APPLICATIONS · SPECIFIC

- DMCA Notice and Takedown: Where the Cost of an Assertion Falls (/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)
- Arkose Labs: Attack Cost and Symmetric Assertion Pricing (/articles/assertion-cost-symmetry/arkose-labs)
- **Sift: Abuse Decisioning and the Cost of an Assertion (/articles/assertion-cost-symmetry/sift)**

TERMINOLOGY

- acknowledgment artifact (/glossary#acknowledgment-artifact)
- assertion-cost counter (/glossary#assertion-cost-counter)
- authorization budget (/glossary#authorization-budget)
- dynamic agent hash chain (/glossary#dynamic-agent-hash-chain)

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