Nick Clark · Published July 29, 2026 · PDF

Invalid Traffic and Click Fraud: Metering by Source, Not Impression

Invalid traffic works because identities are inexpensive and events are counted one at a time. An operator that can present many devices, profiles, or app installs gets a separate line in the ledger for each. Chapter 2 of a pending provisional application discloses a different accounting basis: the agent derives origin-equivalence classes from its own lineage and identity records, then charges its refusal counter once per class within a window rather than once per assertion. Under that structure the contribution of a single origin is bounded irrespective of how many identities that origin presents, because relatedness is evidenced locally from records the agent already holds, with no registry, directory, or shared scoring service consulted.

One Operator, Many Ledger Lines

A fraud operator does not have to defeat a detection model. It is enough to make the accounting unit cheap. Digital advertising generally counts impressions, clicks, installs, and conversions, attributing each to an identity of some kind: a device identifier, a cookie, an app instance, a supply-path seat. Identities of that sort are commonly described as inexpensive to mint, rotate, and discard, while evaluating, contesting, or refunding each unit costs the defender something real. The precise economics vary and are not the subject here.

That asymmetry is what the attack rents. When protection logic decides to stop bidding on a placement, flag a supply source, or withhold payment, the decision usually rests on an accumulation of individual bad events. The operator's counter-move is not to make the events look better. It is to spread the same behavior across more identities so that no single identity accumulates enough weight to trip anything, or, run in reverse, to flood a source with fabricated bad events until a legitimate publisher's supply is throttled. Both moves exploit one property: volume follows identity supply. Better classifiers do not repair a counter that charges per event while the adversary controls the event count.

Where Filtration and Reputation Systems Run Out

Each of the standard responses stalls for a structural reason rather than an engineering one.

Per-event filtration improves the quality of each judgment without changing the number of judgments. An operator able to present ten times as many identities absorbs ten times as many correct refusals and still pushes the same aggregate through.

Shared blocklists and reputation exchanges change the accounting unit to something more durable, but they do it by importing an outside authority. Someone has to attest that identity A and identity B belong to the same operator, and that attestation becomes an asset: portable, contestable, worth attacking. Whoever consults the list inherits its coverage gaps and incentives, plus a failure mode shared with every other consumer of it.

Device attestation and verified supply chains raise the unit cost of an identity, which helps, though they answer a different question. They speak to whether a signal came from a real device or a declared seller. They do not establish that two real devices are operated by the same party, which is the fact the accounting needs.

Probabilistic clustering on behavioral or network features does try to answer that question. Its weakness is evidentiary rather than statistical: the cluster is an inference the defender cannot readily show its work on, awkward to defend in a billing dispute and open to poisoning. Across all four, the metering basis stays at the level of the assertion.

Origin-Equivalence Normalization

The disclosed procedure moves the metering basis from the assertion to the source. In the filed language, the refusal counter (304) of the semantic agent (100) is incremented per origin-equivalence class (200) rather than per conduct evaluation artifact (116).

The chapter states the condition being foreclosed. Under the metering and gating procedure governed by a separate chapter of the filing, volume alone drives the authorization gate (300) to the withheld state (310), and the volume available to an adverse party is limited only by the cost of presenting further identities. Charging the counter per source instead bounds the contribution of a single origin irrespective of the number of identities that origin presents, and no registry, directory, or shared scoring service is consulted.

A source, in the disclosure, is a set of asserting parties (118) between which the agent demonstrates a relation from its own records. On receipt of an artifact, the agent retrieves or instantiates the counterparty identity record (114) for the asserting party, reads from the signed policy object (112) in force an enumeration of declared relation types, and evaluates each of them between the present party and every party already assigned to a class, whether in the current window or a preceding one. Three such types are set out, each independently sufficient. A shared dispatch lineage (202) is evidenced where two dispatch entries record a common parent dispatch entry as their immediate antecedent. A co-signature (204) is evidenced where a single lineage entry bears

signatures verifiable against identity primitives of both parties. A common introduction path (206) is evidenced where both counterparty identity records name an identical introducing party.

Where a relation is evidenced, the present party joins the compared party's class; where more than one class is identified, they merge; where nothing is evidenced, a new class is formed. The assignment, the relation types evaluated, the entries relied upon, and the class identifier are appended to the lineage field and written into each member's counterparty identity record. The filing is explicit that the identifier so written records only the class assignment and is not a determination concerning that party's conduct.

Two properties matter for what follows. The first is locality: the class is computed by the agent from its own records, without reference to a centralized registry, without query to a directory, without participation in a consensus procedure, and without coordination with a further execution node. A second agent holding a different lineage field derives a partition that need not agree, no procedure reconciles the two, and neither admits a class identifier derived by the other, so the identifier is not an identity attested by a third party and confers no portable standing. The second property is narrowness of effect: an artifact from a party whose class already contributed within the window still produces a determination and is still appended to the lineage field, and applies no further increment. Normalization suppresses no determination and withholds no adjudication.

The chapter also handles identities the agent has never encountered, which the base procedure would assign to distinct classes. A severance-survival test resolves a reason-type for each constituent edge from recorded severance events of a payment class and an obligation class; an edge resolves not-typeable where no such severance has been recorded against it and survived within a declared continuation interval, and a class whose every constituent edge so resolves is designated an untested class. Where a class is designated untested **and** its recorded introduction paths converge upon a common ancestor entry within a depth declared in the signed policy object, a cost multiplier

declared in that object, greater than zero and less than unity, is applied to the class contribution. The filing sets no value for that multiplier, as it sets none for the rate threshold, the window, the continuation interval, or the declared depth.

What Changes in the Traffic-Quality Workflow

Take, by way of illustration, a buyer-side or exchange-side agent that meters refusal contributions across a window before an authorization gate moves to the withheld state, whether that means it stops bidding into a supply path, suspends a placement, or holds settlement.

Under per-assertion metering, the operator's playbook is arithmetic. The counter has to reach a threshold or stay under one, and the lever the operator controls is how many identities carry the load. Origin-equivalence normalization moves that lever. Suppose the agent can evidence, from records it already holds, that a large set of app instances descend from a common parent dispatch entry. Those instances are then one class, and that class contributes at most one increment in the window. The filing puts the mechanics directly: normalization is performed on receipt of the artifact and before the per-class increment register is consulted, so a party newly presented within a window is assigned to the class of its related parties before any increment attributable to that artifact is applied, and presentation of a further identity is incapable of yielding a further increment where the relation is evidenced from records the agent already holds. Where no declared relation type is evidenced, the further identity forms its own class and contributes.

The untested-class multiplier addresses the reverse failure, which is the treatment of supply the agent has not transacted with before. Such a party has no recorded severance events of either class, so its edges resolve not-typeable and its class is designated untested. Where classes so designated also share introduction paths converging within the declared depth, the declared multiplier reduces their contribution below that of a single class whose edges resolve want-sustained. The chapter's worked trace shows the

shape. With a declared threshold of five, a multiplier of one fifth, and a depth of two, forty newly encountered parties sharing no declared relation type form forty untested classes. Thirty-five of them converge on a common ancestor entry and each adds one fifth to a weighted contribution register, which reaches seven; the remaining five do not converge, contribute at full weight, and bring the rate accumulator to five. The rate of the window is twelve, and the threshold is satisfied. Had all forty instead recorded a common introducing party, they would have formed one class contributing one increment, and the gate would have remained in the granting state.

The conditioning here is easy to overstate. The multiplier applies only where the class is designated untested **and** its introduction paths converge; an untested class whose paths do not converge contributes at full weight notwithstanding the designation. The reduction is also stated to be temporary: the multiplier does not exclude the class, does not suppress the determinations produced for its artifacts, and does not prevent the class from contributing at full weight where the class later accumulates recorded severance events.

Two further consequences follow from the recordkeeping. Because the class assignment, the relation types evaluated, and the entries relied upon are appended to the lineage field, a later question about why a class was formed has a record to examine rather than only a score. And mid-window discovery is stabilized by rule: on a merge, the merged class is recorded as having contributed where any constituent did, neither the rate accumulator nor the weighted contribution register is decremented, and no further increment is applied where the register records a contribution, so a party evidencing a relation mid-window neither inflates nor deflates that window's rate. Assignments are persistent and not window-scoped, while the per-class increment register alone is reset, so an operator does not face a fresh identity graph each window.

Integration Reality and the Limits of the Approach

Record quality governs everything here. If dispatches are not recorded with their antecedent parent entry, if introducing parties are not captured, or if signatures are not retained in a form verifiable against identity primitives, then no relation type is evidenced and every party lands in its own class. The filing is blunt about the degenerate case: an empty enumeration of relation types assigns every asserting party to a distinct class and restores the original condition. Instrumentation, not the metering change, is where the work sits.

Locality is a tradeoff and not only a benefit. Because the class is derived from one agent's records, one agent's linkage does not help another, and an operator well linked in one party's records may be entirely unlinked in another's. No class identifier is transmitted, and no procedure reconciles two agents' partitions. That is what removes the shared authority and the shared failure mode, and it is also why an ecosystem-wide effect is not something this mechanism produces on its own.

Several things fall outside the chapter and remain the job of adjacent tooling. Whether an event is fraudulent is not decided here; normalization governs the metering alone. Device and supply-path authenticity are not attested. An operator with genuinely disjoint provenance, one that never co-signs, never shares a dispatch antecedent, and enters through distinct introducing parties, evidences no declared relation type and is partitioned accordingly; raising the cost of that posture is the objective, and eliminating the attack is not claimed. The quantities that determine whether any gate moves, among them the rate threshold, the window, the multiplier, the continuation interval, and the introduction-path depth, are policy-declared, with no values fixed by the filing, and setting them badly reopens the volume path the procedure exists to close. What deploys cleanly is narrow: an agent that already keeps lineage and counterparty records, already produces determinations, and already meters over a window can change the basis of that meter from the assertion to the derived source class without adding an external dependency.

Disclosure Scope

The mechanism described here is disclosed in Chapter 2, Origin-Equivalence Normalization, of U.S. Provisional Application No. 64/117,812, filed by Adaptive Query. That chapter covers derivation of an origin-equivalence class from an agent's own append-only lineage field and counterparty identity records, class merge and reconciliation of the per-class increment register, aggregation of refusal contributions per class, and the untested-class cost multiplier with its severance-survival and introduction-path convergence conditions. Separate chapters govern other parts of the architecture, including the metering and gating procedure this normalization feeds; those chapters are not described here, and no architectural statement above draws on them.

This article is published as a general applications discussion and as a timestamped public disclosure. It does not name, assess, or characterize any company, product, or service, does not assert that any party practices the disclosed subject matter, and does not state that a license is required. The application referenced is pending.

Origin-Equivalence Normalization (/origin-equivalence)

All 49 steps → (/inventive-steps)

Metering that can't be gamed by splitting one accuser into many.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

- Origin-Equivalence Normalization: Metering Refusal Per Source, Not Per Assertion (/articles/origin-equivalence/origin-equivalence-normalization)

SECONDARY TECHNICAL

- Content-Blind Encounter Commitment and the Paired Encounter Receipt (/articles/origin-equivalence/content-blind-encounter-commitment)
- Cross-Receipt Equivocation Detection: Catching an Agent That Tells Two Stories (/articles/origin-equivalence/cross-receipt-equivocation-detection)
- The Untested-Origin Saturation Bound: Pricing a Flood of Freshly Minted Identities (/articles/origin-equivalence/untested-origin-weighted-saturation-bound)
- Counterparty-Side Fork Detection: Recognizing a Split Identity From Outside (/articles/origin-equivalence/counterparty-side-fork-detection)
- The Attestation Origin-Equivalence Gate: An Anti-Collusion Bound (/articles/origin-equivalence/attestation-anti-collusion-bound)
- The Bridging Party and the Deferred Merge: Charging One Windowed Increment Against Every Origin-Equivalence Class a Single Asserting Party Bridges (/articles/origin-equivalence/bridging-party-deferred-merge)
- The Common-Execution-Node Relation Type: Charging Co-Hosted Asserting Parties as One Origin (/articles/origin-equivalence/common-execution-node-relation-type)
- Receiver-Side First-Encounter Budget Decrement: Pricing a Stranger's Arrival Per Presented-Material Origin Class (/articles/origin-equivalence/receiver-side-first-encounter-budget-decrement)
- Mutual First Encounter Without Cross-Budget Coordination: Two Strangers, Two Matched Pairs, Two Separate Budget Decrements (/articles/origin-equivalence/mutual-first-encounter-no-cross-budget)

APPLICATIONS · GENERAL

- Sybil-Resistant Agent Networks: When One Accuser Becomes Many (/articles/origin-equivalence/sybil-resistant-agent-networks)
- Fraud Rings in Autonomous Commerce: Metering Per Source, Not Per Claim (/articles/origin-equivalence/fraud-rings-in-autonomous-commerce)
- **Invalid Traffic and Click Fraud: Metering by Source, Not Impression (/articles/origin-equivalence/adtech-invalid-traffic)**

APPLICATIONS · SPECIFIC

- W3C Decentralized Identifiers and Verifiable Credentials: Verifying an Identifier Versus Metering Its Origin (/articles/origin-equivalence/w3c-decentralized-identifiers)
- Okta Auth for GenAI: Agent Identity and Origin Equivalence (/articles/origin-equivalence/okta-auth-for-genai)
- WorkOS: Agent Identity Infrastructure and Source Metering (/articles/origin-equivalence/workos-agent-identity)

- Entra Agent ID: Directory Identity and Derived Origin Classes (/articles/origin-equivalence/microsoft-entra-agent-id)
- SPIFFE and SPIRE: Workload Identity and Origin Equivalence (/articles/origin-equivalence/spiffe-spire)
- Cloudflare Bot Management: Blocking Agents and Metering Sources (/articles/origin-equivalence/cloudflare-bot-management)

TERMINOLOGY

- asserting party (/glossary#asserting-party)
- origin-equivalence class (/glossary#origin-equivalence-class)

Origin-Equivalence Normalization overview → (/origin-equivalence)