Nick Clark · Published July 28, 2026 · PDF

The Common-Execution-Node Relation Type: Charging Co-Hosted Asserting Parties as One Origin

A semantic agent that charges its refusal counter per source of assertion rather than per assertion bounds the contribution of a single origin irrespective of the number of identities that origin presents. Which identities collapse into one source depends on the enumeration of declared relation types the agent evaluates, and the base derivation of Section 2.2 recites three: a shared dispatch lineage, a co-signature, and a common introduction path. The common-execution-node relation type, disclosed in U.S. Provisional Application No. 64/117,812 at Section 10.11, paragraph [0487], supplies a further declared type. It is evidenced where the counterparty identity records (114) of two asserting parties (118) each record a common execution node as having hosted both, whereby parties resident upon a common substrate are assigned to a common origin-equivalence class (200) and their assertions are thereafter aggregated as a single source under the per-source charging of the refusal counter (304).

Cheap Identities, Expensive Refusals

No step of the metering procedure of Section 3 evaluates whether a refusal was well founded. A false conduct evaluation artifact (116) produces the rejected determination (124) and accumulates an increment; an unresolvable one produces the not-determinable determination (126), which increments only where the append-only lineage field (104) records at least one prior such determination for the same asserted conduct event. Volume alone therefore drives the authorization gate (300) of the semantic agent (100) to the withheld state (310), the volume available to an adverse party being limited only by the cost of presenting further identities.

Section 2 forecloses that condition by charging the refusal counter (304) per source of assertion rather than per assertion, a source being a set of asserting parties (118) between which the agent demonstrates a relation from its own records. The contribution of a single origin is thereby bounded irrespective of the number of identities that origin presents.

Which identities collapse into one source depends on the enumeration of declared relation types the agent evaluates. Section 2.2 recites three, each requiring a particular recorded fact: a shared dispatch lineage (202), where two dispatch entries record a common parent dispatch entry as their immediate antecedent; a co-signature (204), where a single lineage entry bears a signature verifiable against an identity primitive of each party; and a common introduction path (206), where the counterparty identity record (114) of each party records an identical introducing party. Where no declared type is evidenced against any already-assigned party, the present asserting party (118) is assigned to a new origin-equivalence class (200); Section 2.5 states that identities not previously encountered and presented as mutually unrelated are each so assigned and each contribute an increment. The enumeration comprises at least one type, an empty one restoring the condition of Section 2.1; paragraph [0487] adds a further entry to it.

A Fourth Way Two Identities Resolve to One Source

Paragraph [0487] recites a further declared relation type for origin-equivalence class (200) derivation, evidenced where the counterparty identity records (114) of two asserting parties (118) each record a common execution node as having hosted both. Where the type is so evidenced, parties resident upon a common substrate are assigned to a common class and their assertions are thereafter aggregated as a single source under the per-source charging of the refusal counter (304).

Position in the ordered procedure. Upon receipt of a conduct evaluation artifact (116) the agent retrieves the counterparty identity record (114) of the asserting party (118), or instantiates one where none exists, retrieves the enumeration of declared relation types from the signed policy object (112) in force, and evaluates each declared type against every asserting party (118) already assigned to a class, whether within the current window or in a preceding one. Each declared relation type is independently sufficient for assignment.

Where the evidence is read. A shared dispatch lineage (202) and a co-signature (204) are evidenced upon entries of the append-only lineage field (104). A common introduction path (206) and the common-execution-node type are evidenced upon counterparty identity records (114) the agent holds, the fact evidencing the latter being that both records name one execution node as having hosted both parties. The evaluation runs, under Section 2.2, upon the agent's own lineage field and the counterparty identity records (114) it holds, 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.

Assignment, merge, and what is written down. Where any declared type is evidenced, the present party is assigned to the origin-equivalence class (200) of the compared party; where more than one such class is identified, the classes are merged and the merged class recorded under Section 2.3. The merged class is recorded in the per-class increment register as having contributed within the current window where any constituent class is so recorded; no merge decrements the rate accumulator or the weighted contribution register; and the merged class contributes, within one window, the minimum of one increment and the number of constituent classes that contributed. The assignment, the declared types evaluated, the entries relied upon, and the resulting class identifier are appended to the lineage field, and that identifier is written into the counterparty identity record (114) of each party assigned to the class. Under Section 2.2, the identifier so written records the class to which the party is assigned and is not a determination concerning that party's conduct.

How aggregation is effected. The per-class increment register of the refusal counter (304) maps a class identifier to a Boolean recording whether that class has contributed an increment within the current window. At the third step of the increment procedure the class is computed and the register consulted. Where it records a contribution already made within the window, the procedure terminates without incrementing the rate accumulator; otherwise the rate accumulator is incremented by one, the register updated, and the run accumulator incremented by one. Normalization is performed upon receipt of the artifact and before the register is consulted, so presentation of a further identity within a window is incapable of yielding a further increment where the relation is evidenced from records the agent already holds. An artifact from a party of a class already recorded still produces a determination and is appended to the lineage field, only the contribution being aggregated: normalization suppresses no determination and withholds no adjudication.

Persistence and locality. Assignment is persistent and is not window-scoped: the identifier survives elapse of the window, and a party assigned in one window is a compared party in each succeeding window without re-derivation. The per-class increment register alone is window-scoped and is reset. The class is local to the agent that derived it. A second agent holding different records derives, from the same population, a partition that need not agree with the first; no procedure reconciles the two, no identifier is transmitted between them, and neither agent admits one derived by the other. The identifier is therefore not an identity attested by a third party and confers no portable standing.

Declared Quantities and Policy-Set Bounds

The filing places the following in the signed policy object (112), declaring no numeric value for any in connection with this relation type.

- The enumeration of declared relation types, each specifying a class of entry of the append-only lineage field (104) and a matching condition over such entries, comprising at least one type.
- The window identifier, identifying a policy-declared window over which the rate counter (306) accumulates, expressed as an interval of wall-clock time, as a count of conduct evaluation artifacts (116) received, or as a count of successor epochs of the agent's dynamic agent hash chain.
- The rate threshold and the run threshold, each an integer. In accordance with an embodiment, the rate threshold is greater than one, and the run threshold is greater than one and not greater than the rate threshold.
- The cost multiplier of Section 2.5, greater than zero and less than unity, applied to the contribution of an untested class whose recorded introduction paths converge upon a common ancestor entry.
- The introduction-path depth of Section 2.5, at least one, counting traversals performed beyond the immediate introducing party.

A worked trace at Section 2.6 declares, by way of illustration, a rate threshold of five, a cost multiplier of one fifth, and an introduction-path depth of two; that trace concerns the common introduction path (206) rather than co-hosting. No count of parties a co-hosted class may hold is declared, nor any interval limiting how long a recorded hosting fact evidences the type.

Composition With the Rest of the Filing

Each mechanism of Section 10 composes on the persistent semantic agent (100) and the substrate elements of Sections 1 through 9, is disclosed as a distinct embodiment, and may be practiced independently or in combination. Paragraph [0487] appears in Section 10.11 alongside further mechanisms operating on the origin-equivalence class (200).

Under the deferred merge of [0478], an asserting party (118) evidencing a declared relation type against members of two classes is recorded as a bridging party without merging them, the classes remaining distinct and the bridging relation being retained for later evaluation; the bridging party contributes at most one increment within the current window, charged against each class it bridges. The multi-window carryover of [0485] retains the per-class increment register across a declared count of windows rather than resetting it each window, so a class contributing an increment in a first window contributes at a reduced weight in an immediately succeeding one.

The same locally derived class is the unit of further mechanisms. Under the attestation origin-equivalence gate of [0452], admission of a carried abstention entry is conditioned upon whether the disclosing agent is assignable, under a declared relation type, to a common class with a party of an entry already held, a bound declared in the signed policy object (112) limiting the count of such entries held from parties of one class. Section 5 gives a determination propagation record effect only where the emitting agent and the asserting party (118) are not assignable to a common class under any declared relation type, that class being computed by the receiving agent from its own memory field (102). The assertion-cost counter (400) carries a per-recipient-class register, its issuance accumulator incrementing where that register records no increment for the class of the receiving party within the current metering window.

Which relation types are in force between two agents can itself be a co-signed term: the mutual admission compact of [0469] enumerates, in one record signed by both parties, declared relation types applicable between them for class derivation. A different unit governs the first-encounter metering of [0391], computed over presented identity material alone.

Adjacent Categories and How They Differ

Several established categories address identity multiplication or co-residency, each by a structure that differs from the one recited here.

Registry-anchored Sybil resistance. Schemes resolving identity against a shared registry or coordinating authority bound identity multiplication at a common point. Section 2.1 draws that distinction expressly: the class (200) is computed by the agent from its own records, without reference to any centralized registry and without coordination with a further execution node.

Network-origin rate limiting and host fingerprinting. Address-based buckets, subnet quotas, and host fingerprints group requests by an inferred common source, then throttle or discard. This structure does neither: an artifact from a party of a class already recorded produces a determination and is appended, only the contribution to the refusal counter (304) being aggregated.

Unfair-rating filtering in reputation systems. Defenses exemplified by the TRAVOS trust model, by beta-reputation filtering, and by the immunization mechanisms surveyed by Dellarocas filter incoming ratings before aggregation to protect a stored score. The filing's background states that the filtering party in such a system is a disinterested aggregator rather than the rated entity, and that filtering imposes no cost on it. Section 3 places the metering structure inside the metered party, written by that party's own determinations.

Remote attestation of a hosting platform. Attestation schemes establish which host an entity runs on by evidence chained to a trust anchor shared with relying parties. The hosting fact relied on in [0487] is already recorded in counterparty identity records

(114) the agent holds, and the class it produces is local and confers no portable standing.

Multi-tenant controls and quorum membership. Controls grouping activity by tenant or instance operate on an identifier the platform itself issues and can reconcile, and quorum schemes bound faulty participants as a fraction of a known set under a shared membership view. Neither appears here: Section 2.2 recites no consensus procedure, and two agents deriving from one population may reach partitions that do not agree.

Nothing above assesses any particular product or implies that any system practices the disclosed subject matter.

Disclosure Scope

The common-execution-node relation type is disclosed in U.S. Provisional Application No. 64/117,812, Section 10.11, at paragraph [0487], with antecedents in Sections 2.1 through 2.4 and in Section 3, which supplies the refusal counter (304) and its increment procedure.

Disclosed: a further declared relation type for origin-equivalence class (200) derivation, evidenced where the counterparty identity records (114) of two asserting parties (118) each record a common execution node as having hosted both; assignment of parties resident upon a common substrate to a common class; aggregation of their assertions as a single source under the per-source charging of the refusal counter (304); and the recording, merge, persistence, and locality consequences Section 2 attaches to an assignment.

Disclaimed: any numeric value for the enumeration, the window, the thresholds, or the multiplier, none being declared for this relation type; any reference numeral for the type itself, none being assigned; any registry, directory, consensus procedure, or coordination with a further execution node in the derivation; any portability of a class identifier between agents; and any reading of that identifier as a determination concerning an asserting party's conduct.

This article is published for defensive purposes and describes subject matter disclosed in the above-identified application. It grants no license and makes no representation as to the scope of any claim.

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)

APPLICATIONS · SPECIFIC

- W3C Decentralized Identifiers and Verifiable Credentials: Verifying an Identifier Versus Metering Its Origin (/articles/origin-equivalence/w3c-decentralized-identifiers)

TERMINOLOGY

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

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