Nick Clark · Published July 28, 2026 · PDF

The Bridging Party and the Deferred Merge: Charging One Windowed Increment Against Every Origin-Equivalence Class a Single Asserting Party Bridges

A semantic agent that meters refusals per source rather than per assertion depends on a partition of asserting parties into origin-equivalence classes, and where a newly presented party evidences a declared relation type against members of more than one class, those classes are merged. The bridging party and deferred merge mechanism, disclosed in U.S. Provisional Application No. 64/117,812 at Section 10.11, paragraph [0478], sets out an embodiment in which the merge is deferred instead. The two origin-equivalence classes (200) remain distinct, the asserting party (118) evidencing a declared relation type against members of both is recorded as a bridging party, and the bridging relation is retained for later evaluation. That party contributes at most one increment within the current window, charged against each class it bridges, and its contribution is not applied a second time where the classes are thereafter merged.

The Partition Changes While the Counter Is Still Running

Section 2.1 records the condition the normalization addresses. Both refusal paths accumulate increments: the rejected determination (124) upon a false conduct evaluation artifact (116), and the not-determinable determination (126) upon one unresolvable against the append-only lineage field (104). Volume alone therefore drives the authorization gate (300) of the semantic agent (100) to the withheld state (310), and the volume available to an adverse party is 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 semantic agent (100) demonstrates a relation from its own records. That set is the origin-equivalence class (200), and the per-class increment register (312) records, for each class, whether it has contributed an increment within the current window.

The partition of parties into classes is itself revised as artifacts arrive. Under the fourth step of the derivation, where more than one class is identified for a newly presented asserting party (118), the classes are merged and the merged class is recorded under Section 2.3. Paragraph [0478] sets out an embodiment in which that merge is deferred instead.

Recording the Party Instead of Collapsing the Classes

The derivation runs upon receipt of a conduct evaluation artifact (116). The semantic agent (100) retrieves the counterparty identity record (114) of the asserting party (118), or instantiates one where none exists, then retrieves from the signed policy object (112) in force an 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. It evaluates each declared relation type against every asserting party (118) already assigned to a class (200), whether in the current window or a preceding one.

Three relation types are recited. A shared dispatch lineage (202) is evidenced where the append-only lineage field (104) holds a dispatch entry for each party, both recording a common parent dispatch entry as their immediate antecedent. A co-signature (204) is evidenced where a single entry bears a signature verifiable against an identity primitive of each party. A common introduction path (206) is evidenced where each party's counterparty identity record (114) records an introducing party and the two are identical. Each relation type is independently sufficient for assignment, and the evaluation is performed upon the agent's own entries and records alone.

Where the fourth step resolves multiplicity by merging, Section 2.3 supplies the reconciliation: the agent appends a merge reconciliation record enumerating the constituent class identifiers, the merged class identifier, and the register values recorded within the current window. The merged class is recorded as having contributed where any constituent class is so recorded, and neither the rate accumulator nor the weighted contribution register is decremented by the merge.

The deferral. Under paragraph [0478], the merge of two origin-equivalence classes (200) is deferred, and an asserting party (118) evidencing a declared relation type against members of two classes (200) is recorded as a bridging party without merging the classes. The filing states two consequences: the classes remain distinct, and the bridging relation is retained for later evaluation.

One increment, marked against every class it bridges. The bridging party contributes at most one increment within the current window. That contribution is charged against each origin-equivalence class (200) it bridges, and the per-class increment register (312) of the refusal counter (304) records each bridged class as contributed upon the contribution, whereby a bridging party neither contributes a further increment per bridged class nor escapes contribution by bridging.

Where the register records each bridged class as already contributed within the window, the bridging party applies no increment.

A later merge does not re-charge. Where the classes are thereafter merged, the reconciliation rule for a merge applies to the merged class, and the bridging party's contribution is not applied a second time.

Consulting the register is not the first step an artifact reaches. The admission evaluator (120) produces a determination and appends it, and the class of that determination is examined before the origin-equivalence class (200) is computed: the accepted determination (122) resets the run accumulator for the action class identified by the conduct descriptor and applies no increment, the not-applicable determination (128) applies no increment and does not reset it, the rejected determination (124) applies an increment, and the not-determinable determination (126) applies an increment only where the append-only lineage field (104) records at least one prior determination of that class for the same asserted conduct event.

Time-scope divides between the two structures. Assignment of an asserting party (118) to a class is persistent and not window-scoped: the class identifier written into the counterparty identity record (114) survives elapse of the window, and a party assigned in one window is a compared party in each succeeding window without re-derivation of the assignment. The register, a mapping from class identifier to Boolean value, alone is window-scoped.

Marking a class as contributed does not, by itself, withhold anything. The rate accumulator holds an integer count of increments applied within the window at full weight; the run accumulator holds a count of consecutive increments applied without an intervening accepted determination (122) for the same action class. The two are compared against the rate threshold and the run threshold respectively, and only upon satisfaction of both comparisons does the refusal counter (304) write the authorization gate (300) to the withheld state (310) for an enumerated set of action classes, the semantic agent (100) transitioning concurrently into the non-executing cognitive mode (302) with respect to those classes.

That enumeration comprises the action classes recorded as implicated by the conduct descriptors of the artifacts that produced increments within the window. Where an increment was withheld by the register, the action class implicated by that artifact's conduct descriptor is not enumerated by reason of that artifact, and the determination produced for it is nonetheless produced and appended. Normalization suppresses no determination and withholds no adjudication; it governs the metering alone.

Declared Quantities and Structural Caps

Each quantity governing the mechanism is retrieved from the signed policy object (112) in force.

- **The window.** 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. In the successor-epoch embodiment the window is not advanced by a party other than the semantic agent (100), nor by manipulation of a clock available to the execution node hosting it. A window expressed as wall-clock time does not carry that property.
- **Rate threshold and run threshold.** Each is expressed as an integer. The rate threshold is an integer greater than one, and the run threshold is an integer greater than one and not greater than the rate threshold.
- **Declared relation types.** The enumeration comprises at least one relation type. An empty enumeration assigns every asserting party (118) to a distinct class (200) and restores the condition of Section 2.1.
- **The increment cap.** Expressed structurally rather than numerically: at most one increment within the current window, that contribution charged against each origin-equivalence class (200) the bridging party bridges.
- **Reset behavior.** The rate accumulator, the weighted contribution register, and the per-class increment register (312) are reset for the succeeding window. The run accumulator is not reset by elapse of the window; it is reset by an accepted determination (122) for the action class identified by the conduct descriptor and by no other event.

Where the Deferral Sits in the Filed Architecture

Section 10 states that each additional mechanism composes on the persistent semantic agent (100) and the substrate of Sections 1 through 9, is a distinct embodiment, and may be practiced independently or in combination with any other. Paragraph [0478] falls in Section 10.11, Further foundational embodiments.

The structures it operates upon are those of Section 2. The derivation of Section 2.2 supplies the classes and the declared relation types; Section 2.3 supplies the reconciliation rule a deferred merge later invokes; Section 2.4 supplies the rule under which an artifact from an already-contributed class produces a determination, is appended, and applies no increment. Under Section 2.5, where a class is designated an untested class and its introduction paths converge upon a common ancestor entry, a declared cost multiplier greater than zero and less than unity applies to its contribution.

Downstream sits Section 3. The register a bridging charge writes is a field of the refusal counter (304), which comprises a rate counter (306) and a run counter (308); the gate write of Section 3.5 is performed upon satisfaction of both thresholds. Under the variant of Section 3.6 replacing the rate accumulator with an exponentially weighted accumulator, the register keeps its Boolean semantics and continues to withhold a second contribution of one class (200) within the window, the decay acting upon the accumulated value and not upon the register.

Locality carries through. An origin-equivalence class (200) is local to the semantic agent (100) that derived it; a second agent holding different records derives a partition that need not agree with the first, no class identifier is transmitted between them, and neither admits one derived by the other. The identifier is accordingly not an identity attested by a third party and confers no portable standing. Two later sections reach back into the derivation: under Section 10.9 a co-signed mutual admission compact record enumerates declared relation types applicable between the parties for class

(200) derivation, and under Section 10.8 the attestation origin-equivalence gate conditions admission of a carried abstention entry on whether the disclosing agent is assignable to a common class with a party of an entry already held.

Distinctions From Registry-Resolved Identity and Rating Filters

The background of the filing enumerates the categories from which the architecture is distinguished. Each distinction set out below is structural.

Sybil-resistance schemes resolve identity against a shared registry or a coordinating authority. Section 2.1 recites the contrast: the origin-equivalence class (200) is computed by the semantic agent (100) from its own records, without reference to any centralized registry and without coordination with a further execution node. Deferring a merge and recording a bridging party are operations over that one agent's own records.

Reputation and trust systems compute a score for an entity from ratings supplied by others and locate it at a registry, a scoring authority, or a shared ledger, the scored entity neither holding the score nor participating in its computation. Defenses against unfair ratings, exemplified by the TRAVOS trust model, by beta-reputation filtering, and by the immunization mechanisms surveyed by Dellarocas, filter incoming ratings before aggregation; the filtering party is a disinterested aggregator rather than the rated entity, and the exercise of filtering imposes no cost upon it. By contrast, the metering structure a bridging contribution writes into is resident inside the metered party and is written by that party's own determinations.

Behavioral integrity systems compute a conformity measure against a baseline model of expected behavior, compare it against a threshold, and reduce capability upon a deficient comparison. The background names U.S. Patent No. 12,563,045 (Daon) and U.S. Patent No. 12,526,244 (Citibank) as conformity detectors, in which divergence from the expected baseline is itself the fault signal and each system has exactly one direction, divergence narrowing capability. Neither, as the background records, admits an assertion by an external party concerning the agent's conduct, provides any procedure by which the agent tests such an assertion against its own record, or meters the agent's refusals of such assertions. Metering refusals is what the refusal counter (304) performs upon the agent itself, and the per-class increment register (312) is the field that bounds the contribution of one class (200) to it.

Policy-enforcement systems implementing the XACML standard return a decision drawn from permit, deny, not-applicable, and indeterminate, a result directed at the request that produces no persistent state in the deciding entity. Accountability protocols exemplified by PeerReview hold the verdict at auditors, and remote attestation architectures exemplified by RFC 9334 locate the appraisal of evidence in a verifier and the consequent grant or denial of authority in a relying party. In each, appraisal is held by a party other than the entity appraised.

These categories are described as the filing describes them, and nothing here characterizes any product, service, or party as practicing the disclosed mechanism.

Disclosure Scope

The mechanism described here is disclosed in U.S. Provisional Application No. 64/117,812 at Section 10.11, Further foundational embodiments, paragraph [0478], captioned "Bridging party and deferred merge." The base architecture on which it composes appears in the same application at Section 2, Origin-Equivalence Normalization, paragraphs [0051] through [0080], and Section 3, The Refusal Counter and Merit-Independent Metering, paragraphs [0081] through [0110]. Composing mechanisms noted above appear at Section 10.8, paragraph [0452], and Section 10.9, paragraph [0469].

Reference numerals used here are those bound to the corresponding elements in the filed specification. This article is a technical summary published for disclosure purposes; the filed text governs, and no parameter value or bound is stated beyond what the filing itself declares.

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)