Nick Clark · Published July 29, 2026 · PDF

SPIFFE and SPIRE: Workload Identity and Origin Equivalence

A party able to present further identities can drive an agent's refusal metering toward a stop without ever presenting a meritorious claim, because volume alone does the work. SPIFFE, and its reference implementation SPIRE, address a different and prior question, as publicly described: how a workload proves cryptographically which workload it is, inside a trust domain that an operator controls. U.S. Provisional Application No. 64/117,812 discloses origin-equivalence normalization, in which a semantic agent derives equivalence classes over asserting parties from its own append-only lineage field and its own counterparty identity records, and charges its refusal counter per source of assertion rather than per assertion. The two sit at different layers, and this article describes how they meet.

Forty faces, one origin

An autonomous agent that accepts conduct evaluation artifacts from outside parties has to decide what happens when those artifacts pile up. Under the metering the filing describes, an artifact that is false produces a rejected determination and accumulates an increment, and an artifact unresolvable against the agent's own append-only lineage field produces a not-determinable determination and accumulates an increment. Both refusal paths therefore move a counter toward a stop.

The failure does not require the adverse party to be right about anything. As the filed disclosure states it, volume alone drives the authorization gate to the withheld state, and the volume available to an adverse party is limited only by the cost of presenting further identities. One party holding forty identities is indistinguishable, to a counter that charges per artifact, from forty parties holding one each. The metering layer measures assertions when the thing it means to measure is sources.

One familiar repair resolves identity against a shared registry or a coordinating authority. The filing treats that as the contrast case, and its stated aim is a procedure in which no registry, directory, or shared scoring service is consulted, so that no shared authority is required. The question becomes whether an agent can bound the contribution of a single origin using only what it already holds.

What SPIFFE and SPIRE set out to do

SPIFFE, the Secure Production Identity Framework For Everyone, is an open specification for giving workloads cryptographically verifiable identities. As publicly described, a SPIFFE identity is expressed as a URI naming a trust domain and a path within it, and is carried in a credential called the SPIFFE Verifiable Identity Document, described publicly in X.509 and JWT forms. Workloads present these documents to each other, and verification proceeds against the trust domain's bundle rather than against a static shared secret held in configuration.

SPIRE is the reference implementation, and it is where the specification becomes operational. As documented publicly, SPIRE separates a server, which holds signing authority for a trust domain and the registration entries defining which identities may be issued, from agents running on nodes that hand credentials to the workloads there. Issuance is described as gated by attestation in two stages: a node attests to the server, and a workload attests to the local agent by properties the agent can observe about the

calling process. Public materials describe issued credentials as short-lived and subject to rotation. SPIFFE also defines federation, so that a workload in one trust domain can verify one in another where the operators have arranged to trust each other's bundles.

The problem that answers is worth naming precisely. It establishes what a workload is, on the authority of an operator entitled to say so, and makes that statement portable and verifiable across the systems that accept the trust domain. That is a well posed problem, and a considered answer to it.

Deriving a class from the agent's own records

The mechanism disclosed in U.S. Provisional Application No. 64/117,812 operates on the metering layer. The refusal counter (304) of the semantic agent (100) is incremented per origin-equivalence class (200) rather than per conduct evaluation artifact (116). A source, in the filing's terms, is a set of asserting parties (118) between which the agent demonstrates a relation from its own records.

Assignment follows an ordered procedure. On receipt of an artifact, the agent retrieves the counterparty identity record (114) of the asserting party, 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. Each declared type is then evaluated between the present party and every party already assigned to a class, whether in the current window or a preceding one.

The filing names three such relation types:

- **Shared dispatch lineage (202)**, evidenced where the lineage field contains an entry recording a dispatch to the present party and an entry recording a dispatch to the compared party, both recording a common parent dispatch entry as immediate antecedent.

- **Co-signature (204)**, evidenced where a single lineage entry bears a signature verifiable against an identity primitive of the present party and a signature verifiable against an identity primitive of the compared party.
- **Common introduction path (206)**, evidenced where the counterparty identity records of both parties each record an introducing party and those introducing parties are identical.

Each declared relation type is independently sufficient. Where one is evidenced, the present party joins the compared party's class; where several classes are identified, they merge; where none is evidenced, a new class is formed. The assignment, the relation types evaluated, the entries relied upon, and the resulting class identifier are appended to the lineage field, and the identifier is written into the counterparty identity record of each party in the class. The filing is explicit that this identifier records class membership and is not a determination concerning the conduct of the party.

Two properties matter for the comparison that follows. First, the class is local. A second semantic agent holding a different lineage field and different counterparty identity records derives, from the same population, a partition that need not agree, no procedure reconciles the two, no class identifier is transmitted, and neither agent admits a class identifier derived by the other. The identifier is therefore not an identity attested by a third party and confers no portable standing. Second, the derivation consults no centralized registry, no directory, no consensus procedure, and no further execution node.

Aggregation follows from the class. A per-class increment register records whether a class has contributed within the window; an artifact from a party in a class already recorded produces a determination and is appended to the lineage field, applying no increment. Normalization suppresses no determination and withholds no adjudication, and governs the metering alone. Because assignment runs before the register is consulted, a further identity presented mid-window cannot yield a further increment where the relation is evidenced from records already held.

The filing also addresses the party arriving with many previously unencountered identities. Where every constituent edge of a class resolves not-typeable under the severance-survival test, the class is designated an untested class; where such a class is so designated 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 policy object is applied to the class's contribution. Both conditions must hold: a designated class whose introduction paths do not converge contributes at full weight notwithstanding the designation. The multiplier does not exclude the class and does not suppress the determinations produced for its artifacts.

Different questions, adjacent answers

Put the two side by side and the divergence is structural rather than competitive. A workload identity framework answers who a peer is, on the authority of an operator entitled to make that statement, and produces a credential built to be portable across everything that trusts the domain. The disclosed architecture answers a later question, how much a set of peers may jointly count against one agent's own refusal metering, and produces a class identifier the filing states is not portable.

Portability is the first axis. Federation, as publicly described, is a designed feature of workload identity: an identity is meant to mean the same thing to every relying party in the domain, and across domains where operators have arranged it. The filed class identifier is specified in the opposite direction. Two agents may partition the same population differently, no procedure reconciles the two, no class identifier is transmitted between them, and neither agent admits a class identifier derived by the other. The filing states that the identifier is therefore not an identity attested by a third party and confers no portable standing.

Scope of derivation is the second axis. An operator-anchored identity is authoritative inside the boundary its operator controls, which is what a fleet operator needs. What the disclosed architecture requires is different in kind: its relation types are evaluated

over entries the agent itself wrote, and the filing specifies that this evaluation proceeds 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.

Convergence is real too. Neither design accepts a self-asserted claim on its face; each builds from evidence instead. They draw that evidence from different places, and key material of the sort a workload identity system produces is the kind of material against which the filing's co-signature test would check an identity primitive.

Coexistence in one deployment

Where both are present, the layering is complementary. A workload identity system establishes, on its operator's authority, that a connecting party controls the key material it claims. The filed procedure takes identity primitives as inputs to its co-signature test. Origin-equivalence normalization then operates over the population of asserting parties and charges the refusal counter per source of assertion rather than per assertion. The two questions are sequenced, not in tension.

The boundaries the filing itself draws are worth restating. Assignment to a class is persistent and is not window-scoped: the identifier written into the counterparty identity record 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 being reset. On a merge, the filing reconciles that register by an ordered rule and appends a merge reconciliation record, so that a contribution already applied within the window is not withdrawn by a later merge. Because every declared relation type is evaluated over the agent's own lineage entries and its own counterparty identity records, the derivation reaches no further than those records. And where the enumeration of declared relation types is empty, every asserting party is assigned to a distinct class, restoring the very condition the section was written to foreclose.

Quantities are policy-declared. The filing specifies the cost multiplier as greater than zero and less than unity, and the declared introduction-path depth as at least one, and locates the values themselves in the signed policy object rather than in the mechanism. The threshold, multiplier, and depth in its worked trace are given by way of illustration.

Disclosure Scope

This article describes subject matter disclosed in U.S. Provisional Application No. 64/117,812, Chapter 2, origin-equivalence normalization. A provisional application is a pending filing and confers no issued rights. Statements about the disclosed architecture describe the filed embodiments, are not claim constructions, and are not legal advice.

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

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)
- 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)