SOLARA ASSOCIATES · TECHNICAL BRIEF

VAID — VERIFIABLE AGENT IDENTITY

# Your agents can prove who they are. They can't prove what they did.

Workload identity told us which agent was running. It was never going to tell us what the agent did. This is the layer that was missing.

---

Version 1.0 Date 6 July 2026 Owner Solara Associates Status Public

We spent two years giving AI agents identity. We gave them the wrong kind.

Walk through any serious agent deployment today and you will find workload identity done well. SPIFFE issues an SVID. A service account sits on IAM. Requests route through a gateway that checks, at the door, whether the thing making the call is the thing it claims to be. This is real engineering and it solves a real problem. For a single-cloud, single-framework deployment with a gateway in the path, it is frequently enough. It answers one question: *is this workload who it claims to be, right now, inside my trust domain.*

Here is the question it cannot answer: *did this specific agent author this exact action, and can anyone verify that without trusting me.* Those are not the same question. The gap between them is the category nobody has named yet, and it is about to matter a great deal.

01

## Workload identity and action identity are different things

WORKLOAD IDENTITY

A property of the **runtime**. Established at startup, it lives inside a trust domain and is vouched for by an authority — IAM, an IdP, a SPIFFE control plane. When agent A calls agent B, the gateway confirms A is A. But it confirms it *because A is inside the gateway's domain*. Step outside, and the vouching stops.

ACTION IDENTITY

A property of the **artifact**. It binds to a specific thing an agent did — this request, this tool call, this decision, with these parameters, at this moment — and it travels with that artifact wherever it goes. It does not ask an authority to vouch. It carries its own proof.

The distinction sounds academic until you put two agents from two organisations in a room. Agent A, governed by company X's IAM. Agent B, governed by company Y's. A asks B to do something on the strength of an action A claims a third agent authorised. Whose IAM adjudicates that? Neither. There is no shared trust domain. The workload-identity model, which is entirely built on shared trust domains, has nothing to say. And this is not a hypothetical edge case invented to sell something — it is the direction the whole field is already running.

02

## OAuth was never going to reach this, and neither is SPIFFE

It is worth being precise about why the existing primitives stop where they stop, because it is structural, not an oversight someone will patch.

**OAuth answers delegation:** a user authorises an application to act on their behalf. Three-legged, user-to-app, scoped by consent. A magnificent solution to the problem it was designed for — a human granting software access to their data. It has no concept of an agent authoring an action, no concept of one agent attenuating authority to another, no concept of a signature on the work itself. You can bolt agents onto OAuth, and people do, but you are bending a human-delegation protocol around a machine-authorship problem.

**SPIFFE answers workload attestation:** this process, on this node, in this trust domain, is who it says it is. Issued at startup, single-trust-domain by design, it explicitly defers authorisation to a policy layer above it. That is the correct design for what SPIFFE is — and the reason it cannot be the thing that proves an action: it is bound to workloads, not actions, and it stops at the edge of its domain.

> The industry standardised the floor. The floor is good. The floor is not the building.

03

## Coordination without verifiable action identity is theatre

The agent ecosystem is sprinting toward coordination it cannot actually secure. We are opening cross-organisation agent protocols: agents will negotiate, delegate, sub-contract, and hand work to other agents across company boundaries. That protocol layer is being built in the open and it is genuinely exciting. But a protocol that lets agents from different organisations coordinate, sitting on top of an identity layer that only works inside a single organisation's perimeter, is coordination resting on a fiction. You have built the roads and skipped the signatures on the cargo.

Watch the seam. The cross-organisation protocol is shipping. The cross-organisation identity is left at the IAM perimeter. That tension — protocol for the many-to-many world, identity for the one-to-one world — is not a small gap to be closed later. It *is* the gap. Every demo where two agents "trust" each other across a boundary is, under the hood, either inside one trust domain pretending to be two, or trusting on the strength of nothing a third party could check.

Action identity is what makes the coordination real instead of performed. Without it, an audit log is a list of claims you have to take on faith. With it, every action in the chain carries a signature any party can verify, and the log becomes evidence.

04

## The perimeter vouches. The math vouches.

This is the whole fork in one line. In the workload model, identity is something the **perimeter** vouches for — the gateway, the IAM, the trust domain stands behind the claim. Inside the perimeter, that works beautifully. The proof is only as portable as the perimeter, which is to say: not at all. Leave the domain and the proof evaporates, because the thing that made it true was the authority, and you left the authority behind.

In the action model, identity is something the **math** vouches for. A full-document cryptographic signature — Ed25519 over the canonical bytes of the action — is true because the signature verifies, full stop. No authority is consulted. No trust domain is required. Any party, anywhere, online or offline, now or in ten years, can check it against a public key and get a yes or a no. The proof travels with the artifact because the proof *is* the artifact.

WHAT A VAID IS

A **Verifiable Agent Identity Document**: the signed record of an agent's action, carrying a per-action signature that anyone can verify with no shared trust domain. The name says *document* because that is what it is — a portable, self-proving artifact — but the substance is the per-action signature inside it. It is not a passport checked at a gate. It is a notarised statement that stays notarised no matter where it goes.

Count the cost of each model at scale. Trust by perimeter means federation: every pair of domains that needs to trust each other has to establish and maintain a relationship — $O(n^2)$ relationships, brokered, rotated, and audited, growing quadratically as the ecosystem grows. Trust by signature means verification: any party checks any signature against a public key — $O(1)$, one operation, no relationship, no broker. The first model gets more expensive every time the world gets bigger. The second does not notice.

05

## Framework-neutrality is the moat, and it is the one that holds

A fair question: if this layer is real and useful, why won't the platforms simply absorb it? They will absorb the parts they can. Cloud-neutrality — running across AWS, Azure, GCP — is a real near-term advantage and an entirely copyable one. If the only moat were "works on more than one cloud," it would be a matter of time.

The moat that does not yield is **framework-neutrality**, and the reason is structural. A platform's agent framework is its funnel. The entire commercial logic of a first-party agent stack is to make that stack the path of least resistance and the privileged citizen of the governance layer. A platform *can*, in principle, treat an agent built on a competitor's framework, on a competitor's cloud, as a first-class governed entity. It will never *want* to, because doing so dismantles the funnel that justifies building the platform in the first place.

> So the standard has to come from outside the funnel — neutral, cryptographic, portable. It is deliberately the layer no platform is incentivised to build, which is exactly why it needs to be built in the open and given away.

06

## Complement, don't fight

None of this is a war on the floor. The floor is good and the floor is necessary. SPIFFE workload attestation is a real input: a VAID can consume an SVID as one attestation among others and sign over the action on top of it. A VAID verifier can run inside a platform's own gateway as an extension, checking the signature at the same chokepoint that already checks the workload. The honest framing is not "replace your gateway." It is: here is the portable identity layer that survives outside the gateway, and can also ride inside it.

The depth only earns its keep where you actually hit the boundary the floor stops at: multiple clouds, multiple frameworks, multiple organisations, or any setting where two parties must trust each other's actions without trusting a shared authority. If you are single-cloud, single-framework, gateway-in-path, the floor may be all you need — and you should use it. Action identity is for the world the floor cannot reach, and that world is arriving on the back of the same cross-organisation protocols everyone is racing to ship.

07

## Verify it yourself

A standard that asks you to take its word for it would be its own kind of theatre. So the specification and two independent reference implementations — one in Rust, one in Python — are being prepared for open release under Apache-2.0. The Rust implementation, `vaid-pop`, is the proof-of-possession primitive: the canonical signing path — JCS canonicalization, SHA-256, Ed25519 over the digest — with `vaid-client` as the reference SDK on top. The Python implementation is its own `vaid-pop`, written independently. Both let you sign an action and verify the signature standalone, with no server and no service in the loop.

Here is what makes “neutral standard” more than a phrase. Both implementations are pinned by the **same frozen conformance vector**, and they reproduce it byte for byte. The same canonical input produces the same SHA-256 digest and the same Ed25519 signature in Rust and in Python — down to the last byte, with no shared code between them and no runtime in common. That is the difference between a standard you are asked to believe in and a standard you can check.

CHECK IT NOW

You do not have to wait for the release to see the signing path run. The live in-browser demo on `solara.associates/vaid` executes the exact canonical path — JCS → SHA-256 → Ed25519 — against that same frozen conformance vector, in your own browser, with real Web Crypto. Any third implementation that hits the same vector is byte-compatible with both, by construction. That is what a standard is.

> **Workload identity told us which agent was running. It was never going to tell us what the agent did. That layer was missing — and now it is opening.**

---