



THE TRUST LAYER FOR MULTI-AGENT SYSTEMS

Every multi-agent system you build today is a liability.

Agents act, decide, move money, and take action on behalf of people — but nothing underneath them knows who acted, what they were allowed to do, or how to stop them. SYNTHERA is that missing layer: verifiable identity, enforced authority, and a tamper-proof record, as infrastructure rather than code each team rewrites.

WHY THIS IS A CATEGORY, NOT A FEATURE

- 2013** **Docker → Kubernetes.** Apps couldn't run consistently across environments. Containers became the assumed way to ship software.
- 2018** **The service mesh.** Services couldn't talk securely at scale. Istio and Envoy became the assumed way to connect them.
- Now** **The trust layer.** Agents can't be trusted, audited, or contained. SYNTHERA becomes the assumed primitive they're built on.

Whoever owns the trust layer owns the category — the way Docker and the service mesh won, by being assumed.

WHAT IT IS

A foundation beneath every agent framework — framework- and cloud-agnostic, a control plane, not an app or a model. TCP/IP let any computer trust any other on a network; SYNTHERA is that trust layer for autonomous agents.

Framework-agnostic

Cloud-agnostic

A control plane

THE GAP IT FILLS

- 01 No real identity.** Agents identify with a string anyone can claim.
- 02 No enforced limits.** "Only do X" lives in app code, applied inconsistently, bypassable.
- 03 No safe coordination.** Agents call other agents with no verified boundary between them.
- 04 No tamper-proof trail.** No cryptographic record of who did what, under whose authority.

It works in the demo and becomes a security incident in production.

THE CORE PRIMITIVE — VAID

Every agent carries a **Verifiable Agent Identity**

Document: a cryptographically signed proof of who it is, what it may do, and where it came from — and to act, it signs each request with its own key. Around it sit the shared primitives every product inherits: VAID, SYNTHERA, UAB, Statute, Crucible, Sentinel, ShadowStack and AI Factory.

THE MODEL — GIVE AWAY THE STANDARD, RUN THE ENGINE

OPEN — THE STANDARD

The VAID spec & reference SDKs

The signing format, policy language, and reference implementations — free, public, neutral. The cable everyone plugs into; adoption is the goal, no lock-in by design.

COMMERCIAL — THE ENGINE

The running foundation

The policy engine, authorization mesh, federation, and enforcement — delivered as a service. The signing SDK is free; the engine that verifies, decides, and records is where the value lives.

WHERE IT IS

Built and proven to run: the foundation is hardened, the reference products run on one canonical signing layer, and least-privilege delegation is enforced fail-closed. The signing path runs in your browser today against a frozen conformance vector — real cryptography, no trust required. **The open standard and its reference signer: release in preparation.**