SOLARA ASSOCIATES

SYNTHERA

The trust layer for multi-agent systems. Verifiable identity, enforced authority, and a tamper-proof record — as infrastructure, not code each team rewrites.

Mission deck Version 1.0 6 July 2026 solara.associates

THE THESIS

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

Agents act. Agents decide. Agents move money, change state, and take action on behalf of people. But there is no layer underneath them that knows who acted, what they were allowed to do, or how to stop them when they go wrong.

WHY THIS IS A CATEGORY, NOT A FEATURE

Each era’s missing layer becomes the next era’s assumption

2013 · Container era

Docker → Kubernetes

Apps couldn’t run consistently across environments. Containers became the assumed way to ship software.

2018 · Cloud-native era

The service mesh

Services couldn’t talk securely at scale. Istio and Envoy became the assumed way to connect them.

Now · Agentic era

The trust layer — SYNTHERA

Agents can’t be trusted, audited, or contained. A trust layer becomes the assumed primitive they’re built on.

> Docker won by being assumed. Service meshes won by being assumed. Whoever owns the trust layer owns the category.

WHAT IT IS

A foundation that sits beneath every agent framework

SYNTHERA gives AI agents a **verifiable identity**, enforces **what each one is allowed to do**, and keeps a **tamper-proof record** of everything they do. The analogy: TCP/IP let any computer trust any other on a network. SYNTHERA is that trust layer for autonomous agents.

FRAMEWORK-AGNOSTIC

Governs agents on ADK, LangChain, or CrewAI — it sits below them, not inside any one.

CLOUD-AGNOSTIC

Deployable on GCP, AWS, or Azure. Not tied to any one vendor's stack.

A CONTROL PLANE

Wraps agent deployments and mediates identity, policy, and containment.

NOT AN APP OR A MODEL

The plumbing every serious multi-agent system needs and currently hand-rolls badly.

WHY IT EXISTS

Multi-agent systems have no trust layer — so everyone fakes one

01 **No real identity.** Agents identify with a string in a request body. Anyone can claim to be any agent.

02 **No enforced limits.** “This agent can only do X” lives in app code, applied inconsistently and bypassable through side doors.

03 **No safe coordination.** Agents calling other agents do so with no verified boundary between them.

04 **No tamper-proof trail.** When something goes wrong, there is no cryptographic record of who did what, under whose authority.

> The cost: it works in the demo, and becomes a security incident in production.

THE ARCHITECTURE

One foundation. A stack of products built on it.

Identity, policy, audit, containment, and coordination live in the foundation as primitives — available to everything above. Build a new tool on top and it inherits all of them for free.

Identity · Policy · Audit · Containment · Coordination

VAID
The core identity primitive — the signed proof every agent carries

UAB
Compose governed multi-agent systems on any protocol

Statute
AI governance — rules as a versioned constitution

Crucible
Evaluation — score agent behaviour against a rubric before it ships

Sentinel
Agentic security — watch and threat-classify in real time

ShadowStack
Containment — ghost-deploy changes against a digital twin

AI Factory
Adversarial build-and-review with a human final gate

SYNTHERA
The foundation itself — where the value lives

THE CORE PRIMITIVE

VAID — a passport for every agent

Every agent carries a **Verifiable Agent Identity Document**: a cryptographically signed record proving who it is, what it's allowed to do, and where it came from. It isn't a bearer token you can steal and reuse — to act, an agent signs each request with its private key. You can't impersonate an agent without its key.

IDENTITY	SCOPE	CAPABILITIES	LINEAGE	SIGNATURE
Which agent, which product	The resources it may touch	The actions it may take	Who minted it — the chain of authority	Cryptographic seal — tamper-evident

HOW CONTROL WORKS

Every action is authorized at one chokepoint

Every operation is checked against a cryptographically verified identity, at a single point every request must pass through. No side doors.

1

Who are you?

Verify the signature against the VAID. Forged or unsigned → denied.

2

Are you allowed?

Check the operation against the VAID's scope and capabilities. Not granted → denied.

3

Across organisations?

Calling another party requires that party's explicit consent. No consent → denied.

4

Recorded.

Every decision — allow or deny — written to a tamper-proof audit ledger.

> Default is deny. Anything not explicitly permitted is refused — the opposite of how most systems leak.

WHY IT WINS

Double neutrality — the ground no incumbent can hold

CROSS-FRAMEWORK

Any agent framework

Works across ADK, LangChain, and CrewAI — not locked to one framework's way of building agents.

CROSS-CLOUD

Any cloud

Runs on GCP, AWS, and Azure — not locked to one cloud's stack. Two axes of neutrality at once.

> A first-party agent stack can't follow here. Its framework is its funnel — treating a competitor's agent, on a competitor's cloud, as a first-class citizen dismantles the very funnel that justifies the platform. Neutral, cryptographic, portable identity has to come from outside the funnel. That intersection is the defensible ground.

THE MODEL

Give away the standard. Run the engine.

OPEN — THE STANDARD

The VAID spec & reference SDKs

The signing format, the policy language, and the reference implementations — free, public, and neutral. This is the cable everyone plugs into. Adoption is the goal; there is no lock-in here 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.

> The hinge: **the free standard creates the demand; the engine is the only thing that fulfils it.** The same open-core playbook that built the last generation of infrastructure companies.

WHERE WE ARE

Not a deck-deep idea — built, and proven to run

FOUNDATION HARDENED

Identity, policy, and audit were built and hardened through repeated internal security review; the identity signing primitive is guarded by a cross-language conformance suite (8/8) in CI. Authorization is enforced at a single chokepoint.

PRODUCTS LIVE ON IT

The reference products all run on the foundation through one canonical signing layer — proving the shared-primitive thesis in practice.

DELEGATION WORKS

Agents mint attenuated sub-agents end-to-end, each holding its own key. The least-privilege model runs, not just compiles.

PROVEN LIVE

A stand-in consumer exercised the full allow/deny contract (9/9) against a running foundation instance; the deployed Cloud Run foundation independently enforces identity minting and least-privilege delegation over HTTPS.

VERIFY IT YOURSELF

The signing path runs in your browser today against a frozen conformance vector — real cryptography, no trust required.

WHAT'S NEXT

The open standard and its reference signer — release in preparation. The engineering is done; the standard ships next.