Quantum Field INC.

The future will be *verified.*

The trust infrastructure for the agentic economy — identity, authority, and governance made independently verifiable. An executive and technical exposition of the PRAXIS Agent ID protocol.

PRAXIS.

Reference implementation @praxis/agent-id · Tezos L1 + Arweave

EXECUTIVE PAPER · JULY 2026 · CONFIDENTIAL

DOCUMENT CONTROL

How to read this *paper*.

This is a single document written for three audiences at once — institutional and standards reviewers, engineering, and investors — because the credibility argument is identical for all three: every claim about what exists today is status-labeled and independently checkable against the reference implementation.

Purpose	The executive and system-level exposition of PRAXIS — the brand thesis, the architecture, and the verification doctrine — mapping each key flow with the reference implementation located honestly against it.
Companion documents	PRAXIS Agent ID — Technical Overview (Third Edition); the Architecture Exposition; ARCHITECTURE.md and AGENT_BRIEF.txt inside the reference repository.
Reference implementation	@praxis/agent-id — TypeScript SDK, SmartPy FA2 contract (audit-gated draft), JSON Schemas, RDF ontology, golden vectors, CI with an artifact-drift gate.
Verification state	Strict typecheck, lint, and format clean · **55/55 tests** · chain + anchor re-verified · deliberate tamper rejected · drift gate proven in both directions · KG artifacts proven in Oxigraph.
Figure provenance	All hashes and counts are real values emitted by the package pipeline (`npm ci && npm run ci`) from the shipped example chain, except where a figure is explicitly marked illustrative.

THE HONESTY LEGEND

Every claim about what exists today carries a status drawn from the repository's vision-to-code cross-reference. A box is `SHIPPED` only if the CI pipeline exercises it today.

`SHIPPED`	Implemented, tested, and enforced in the reference repository now — exercised by the CI pipeline today.
`PARTIAL`	A defined seam exists — an interface, field, or anchor — with the remainder in a named roadmap phase.
`ROADMAP(N)`	Lands in phase N of the plan (0 substrate — complete · 1 contract/audit · 2 providers · 3 service · 4 standards · 5 hardening).
`VISION`	A sibling system in the wider PRAXIS/TAP stack, out of this repository's scope by design.

SECTION 01 — EXECUTIVE SUMMARY

The limiting factor was never intelligence. It is *trust*.

Software is becoming an economic actor. Autonomous agents are already initiating payments, executing settlement legs, and negotiating inside regulated workflows. Every digital system ever built assumed a human stayed inside the loop — and that assumption is ending. The defining question of the agentic internet is no longer whether machines can think. It is whether they can be trusted to act, and prove it.

Authentication tells a system who logged in. It says nothing about whether an autonomous agent is **authorized to act**, on whose authority, under which constraints, and who answers when it goes wrong — the questions a bank, a regulator, or a court will actually ask. Existing infrastructure answers none: reputation scores are farmable and uncitable, certificates bind a key to a name and stop there, and registries reduce to "trust the registry."

Quantum Field builds the missing layer. **PRAXIS** is an open protocol family in which identity, authority, and accountability are machine-readable, cryptographically anchored, and independently verifiable — without trusting any intermediary. Its foundational specification, **Agent ID**, gives every autonomous entity a verifiable legal identity record: anchored on Tezos, stored permanently on Arweave, and resolvable to a determination anyone can recompute from public artifacts alone.

ANCHORED

On-chain

Identity commits to a soulbound Tezos anchor. The document is re-hashed against that commitment before anyone relies on it — the gateway is never trusted.

ACCOUNTABLE

Depth-0 root

Every delegation chain must terminate in a legally answerable principal. An identity that cannot name one is rejected at validation.

STATELESS

No score

Trust is recomputed from scratch at each decision. There is no balance to farm, buy, or Sybil-inflate — only replayable evidence.

What a reader should take away

- **The thesis.** Trust must move from institutional convention into computational infrastructure. "The future will be verified" is a design constraint, not a slogan.
- **The architecture.** A layered system in which each component does one job — the knowledge graph creates understanding, the AI creates intelligence, Tezos creates finality, and **PRAXIS creates trust**.
- **The discipline.** The reference implementation is real and status-labeled: **55/55 tests**, an artifact-drift gate proven both ways, every figure reproducible from one command. What is not yet done is stated as plainly as what is.

SECTION 02 — THE BRAND THESIS

A single principle, read differently across every field it touches.

"The future will be verified" is not marketing. It is a statement about where trust has to live in an economy where the actors are increasingly autonomous. Below is the same principle, expressed in the vocabulary of each domain where autonomous systems are about to become participants.

DOMAIN	WHAT "VERIFIED" MEANS THERE
Intelligence	AI systems carry verifiable identities, scoped permissions, and provenance for every action they take.
Finance	Transactions and assets settle against cryptographic proof rather than reputation or intermediation.
Law	Authority and accountability become machine-verifiable — delegation encoded, not merely documented.
Knowledge	Facts, provenance, and lineage are cryptographically anchored — a knowledge graph of trust itself.
Commerce	Autonomous agents transact on demonstrated authority — proving what they are permitted to do.

The design creed

One sentence governs every field in the schema and every line of the validator:

CREED

Identity is a function of *authorization and accountability* — never reputation.

There is no score, no endorsement count, no activity volume anywhere in the schema. An identity that cannot show its accountable principal, its authorizing attestation, the assurance of its keys, and the unbroken history of its own versions has not anchored identity — it has asked to be trusted.

SECTION 02 — THE BRAND THESIS · CONTINUED

Every existing mechanism fails the *agent question*.

Each mechanism below establishes something real — and none of it is what a counterparty relying on an autonomous agent actually needs. The failures are structural, not incidental: they are properties of the design, not gaps to be patched.

MECHANISM	WHAT IT ESTABLISHES	WHY IT FAILS THE AGENT QUESTION
API keys / OAuth	A session is authenticated.	Authenticates a session, not an accountable principal — and leaves no tamper-evident history a third party can replay.
PKI certificates	A key is bound to a name.	Says nothing about delegated scope, assurance level, or lifecycle; revocation is a weak, poll-based afterthought.
Reputation / scores	An aggregate signal of past behavior.	Actively harmful here: farmable, purchasable, Sybil-inflatable — and uncitable by a regulator or court.
Web2 registries	A directory entry exists.	Reduces to "trust the registry" — a single point of failure, capture, and silent edit.

> **THE MISSING PRIMITIVE**
>
> A cryptographically anchored, append-only, standards-graded identity record whose every claim is **independently checkable from public artifacts alone** — with no reliance on the issuer's continued cooperation, honesty, or existence.

The remainder of this paper describes how PRAXIS supplies that primitive: the architectural model that composes it (Section 03), the identity record at its center (Sections 04–05), the anchoring and verification doctrine that makes it checkable (Sections 06–08), and the protocol family, engineering baseline, and boundaries that make it real (Sections 09–11).

SECTION 03 — THE CORE ARCHITECTURAL MODEL

Five planes. Each component does *exactly one job*.

The architecture is best understood as five interconnected planes. Applications consume trust; the AI agent layer acts; the PRAXIS trust layer establishes identity, authority, and accountability; and two substrate layers — the semantic knowledge graph and the Tezos cryptographic anchor — supply meaning and proof respectively.

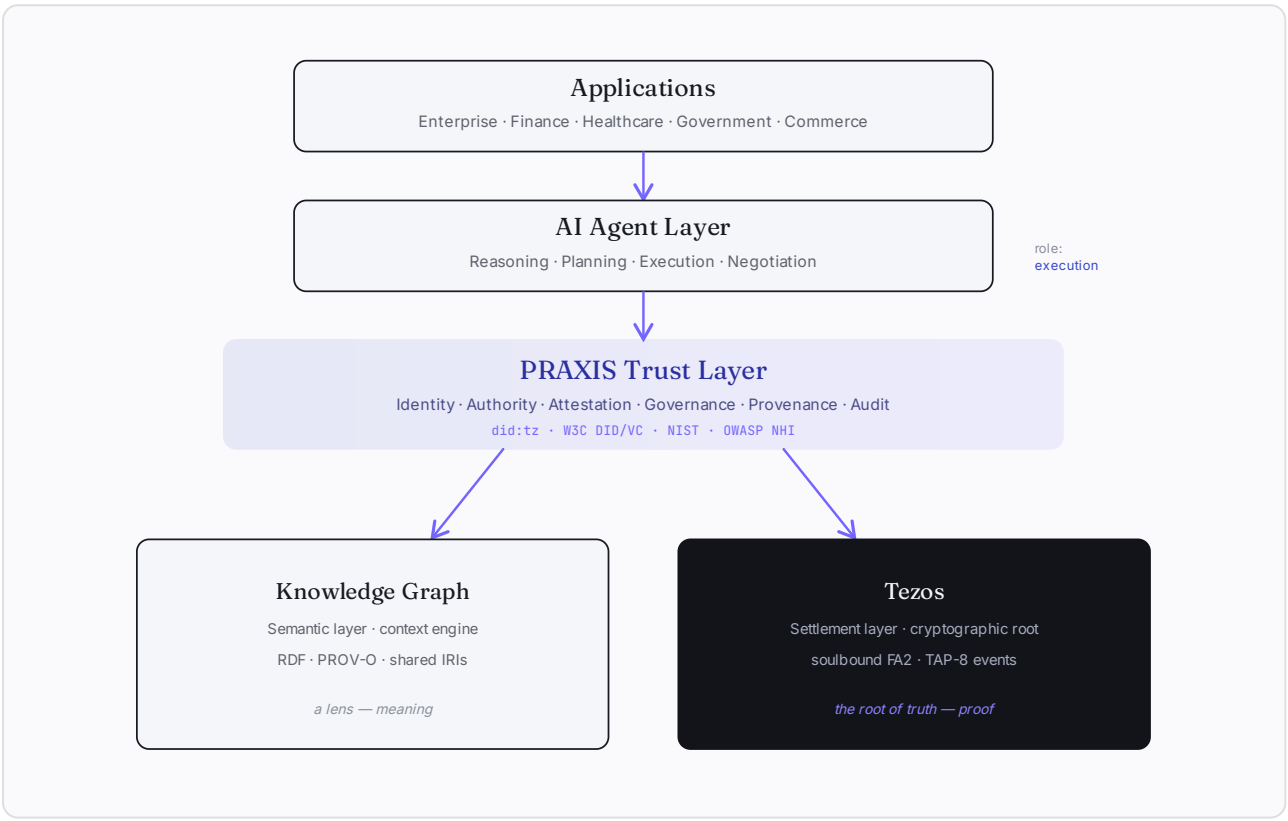

Figure 1 — The five-plane model. The two substrate layers are not peers in kind: the knowledge graph is a lens; the blockchain is the root of truth, always able to overrule meaning. That asymmetry is the single most important property of the design.

THE COMPOSITION PRINCIPLE

Applications consume trust; the AI agent layer acts; the PRAXIS layer establishes identity, authority, and accountability; and the two substrates supply meaning and proof. Proof always overrules meaning — `store proof on-chain, store complexity off-chain` (invariant I2, enforced by construction).

SECTION 04 — THE AI AGENT LAYER · ROLE: EXECUTION

The model is not the identity.

The AI agent is the actor — it reasons, plans, calls APIs, negotiates, and executes. But the model itself is not the identity, and that distinction is the hinge of the entire architecture. A model, whether a frontier system or an enterprise fine-tune, is analogous to a human's cognitive capability. It is not the legal principal.

COMPONENT	FUNCTION
Model	Intelligence — the reasoning capability.
Agent runtime	Execution — the environment that carries out actions.
Agent ID (AID)	Identity — the machine-readable legal record of who this is.
CPoA	Authority — the live, scoped mandate the agent acts under.
Controller	Accountability — the legally answerable principal.

Table — the separation of concerns that lets identity outlive any particular model.

Because identity is separated from intelligence, **an agent may change models without changing identity**. The identity persists precisely because the accountable relationship persists — the `did:tz` name, the delegation chain, the controller, and the anchored history all survive a runtime swap. A `provenance { model, modelVersion, operator }` block records which intelligence sits beneath the identity at each version, so a relying party can detect model substitution without the substitution ever touching who the agent is.

```
# illustrative — the runtime beneath the identity may change; the name does not
Agent ID   did:tz:mainnet:tz4…  (agent.procurement.acme)
Runtime    frontier-model procurement engine, v4   ← was v3
Authority  purchase inventory up to $250,000       (live CPoA)
Controller Acme Corporation (Delaware C-corp) tz1…
```

> **WHY THE AGENT IS NEVER THE TRUST LAYER**
>
> The reasoner presents an identity and is checked against it; the trust layer neither reasons nor acts, and the reasoner neither issues nor verifies its own authority. The AI operates **inside** the trust layer, never as it.

SECTION 05 — THE AGENTIC IDENTITY DOCUMENT

Four questions a skeptical counterparty must resolve.

The Agentic Identity Document (AID) is the canonical identity artifact — not metadata, but a cryptographically verifiable legal identity record. It exists to answer, for a bank compliance desk, a regulator, a settlement contract, or another agent, exactly four questions.

Figure 2 — The four questions the AID resolves. Capabilities are a discovery surface, never a grant; authority is always the live, scoped, revocable CPoA, default-deny at the point of action.

The accountable root

The delegation chain **must terminate in a depth-0 accountable root** — the anti-"unaccountable void" rule, enforced by `validateAid`. An identity that cannot name a legally answerable principal is rejected at validation, making attribution a property of construction.

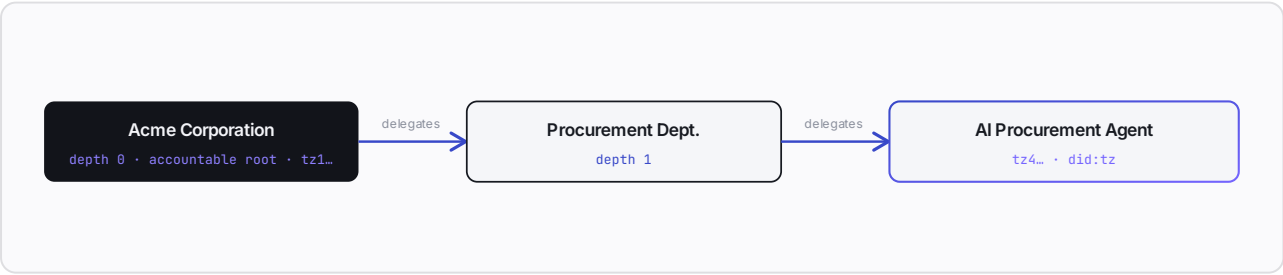

Figure 3 — The delegation chain. Every hop is signed; the chain must resolve to a depth-0 root or the identity is rejected. The agent is a tz4 account; the accountable root a tz1 legal principal (TAP-1).

SECTION 06 — TEZOS + ARWEAVE · THE ANCHORING MODEL

No single layer is trusted alone.

An agent's identity exists at three layers, each doing one job. The document is the source of truth; the token is the anchor; the chain is the transparency log. A resolver that checks only one has not verified identity — it has taken a gateway at its word.

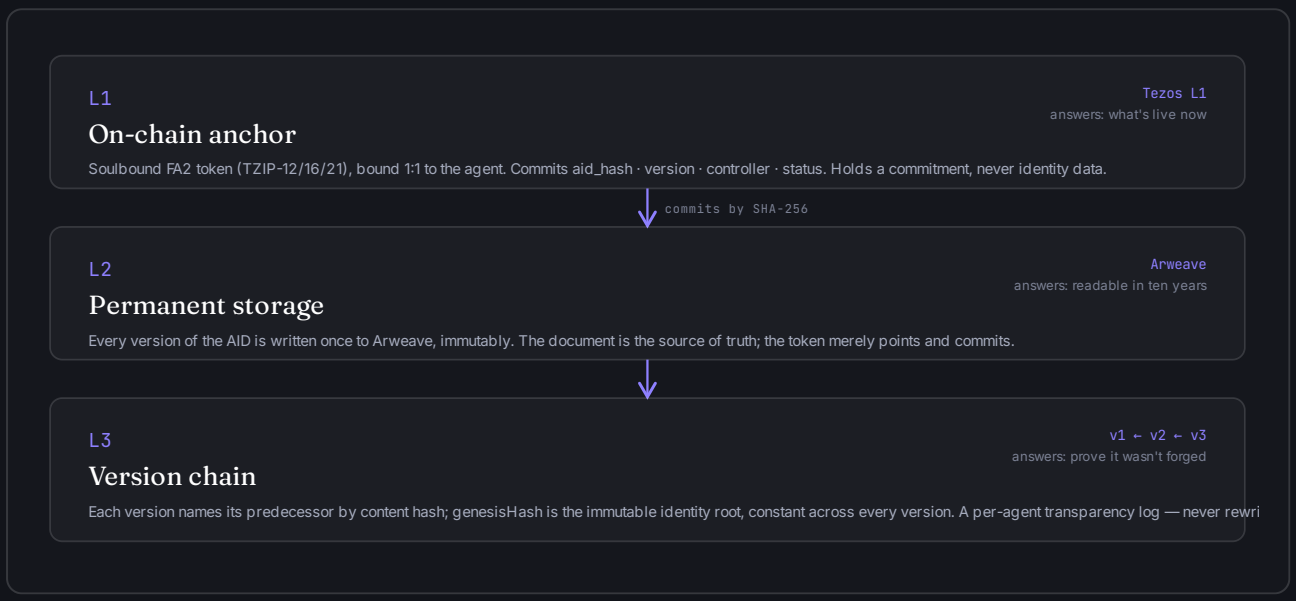

Figure 4 — The three-layer anchoring model. Tezos buys liveness and control; Arweave buys survivability. The pairing is deliberate: neither layer alone can answer both "what is authoritative now" and "will this be readable in ten years."

The on-chain anchor

On-chain, the contract stores only the commitments that matter and emits a TAP-8 event for every state change, so the full history is replayable from the chain itself. These are the real, coherent v1 values from the shipped example chain.

```
# Tezos — Agent ID NFT (soulbound, TZIP-12/16/21)
{
  aid_hash     c2349d4a…e976b332   // content commitment
  aid_uri      ar://<txid>           // where to fetch the document
  aid_version  1                   // which version is authoritative
  genesis_hash 1fe73207…ce70394f   // immutable identity root
  controller   tz1KqTpEZ7Yob7…     // who is accountable
  soulbound    true
  status       active              // is it live
}
```

SECTION 07 — INTEGRITY · THE TWO COMMITMENTS

Two hashes, both *well-defined*.

The whole model rests on two commitments that never fall into a fixpoint. Each is defined over a preimage that excludes itself, so the value is well-defined and its constancy is checkable forever. This is the detail that makes "re-hash and compare" a rigorous operation rather than a hopeful one.

CONTENTHASH(DOC)

The per-version commitment

SHA-256 of the canonical document with the proof envelope removed. This is what chain links reference and what the on-chain NFT anchors. Every resolver recomputes it against the fetched bytes.

```
sha256( canonical( doc \ {proof} ) )
```

GENESISHASH

The immutable identity root

Derived once at genesis, then carried unchanged through every version. Excluding it from its own preimage removes the fixpoint, so the value is well-defined and its constancy is checkable on every chain walk.

```
sha256( canonical( genesis \ {proof, genesisHash} ) )
```

Canonical encoding is the precondition

Hash verification is meaningless if two implementations serialize differently. The document is encoded with a single canonical rule — recursive key sorting, integers or decimal strings with no floats, fixed unicode escaping — frozen as `praxis-canonical-json/1`. A published suite of golden byte-equality vectors fixes `{ input, canonical, sha256 }` across the tricky cases, plus rejection vectors an encoder must refuse. A Python indexer or a Rust verifier reproduces every acceptance vector and rejects every rejection vector, proving byte-for-byte hash compatibility **without reading a line of the reference TypeScript**.

GUARANTEE	MECHANISM	STATUS
Deterministic encoding	praxis-canonical-json/1 · 9 acceptance + 3 rejection golden vectors	SHIPPED
Content binding	contentHash re-computed against fetched bytes at every resolution	SHIPPED
Immutable root	genesisHash constant, checked on every chain walk	SHIPPED
Aggregate signature	BLS12-381 over canonical signing bytes, behind an audited provider	PARTIAL

SECTION 08 — THE VERIFICATION DOCTRINE

Not asserted. *Computed.*

The storage layer is never trusted; it is verified. Resolution is a mechanical pipeline over public artifacts alone — the issuer's cooperation is never required. A bank compliance desk, a regulator, a settlement contract, or another agent reaches the same determination from the same evidence. Any failed check stops the pipeline: the system fails closed.

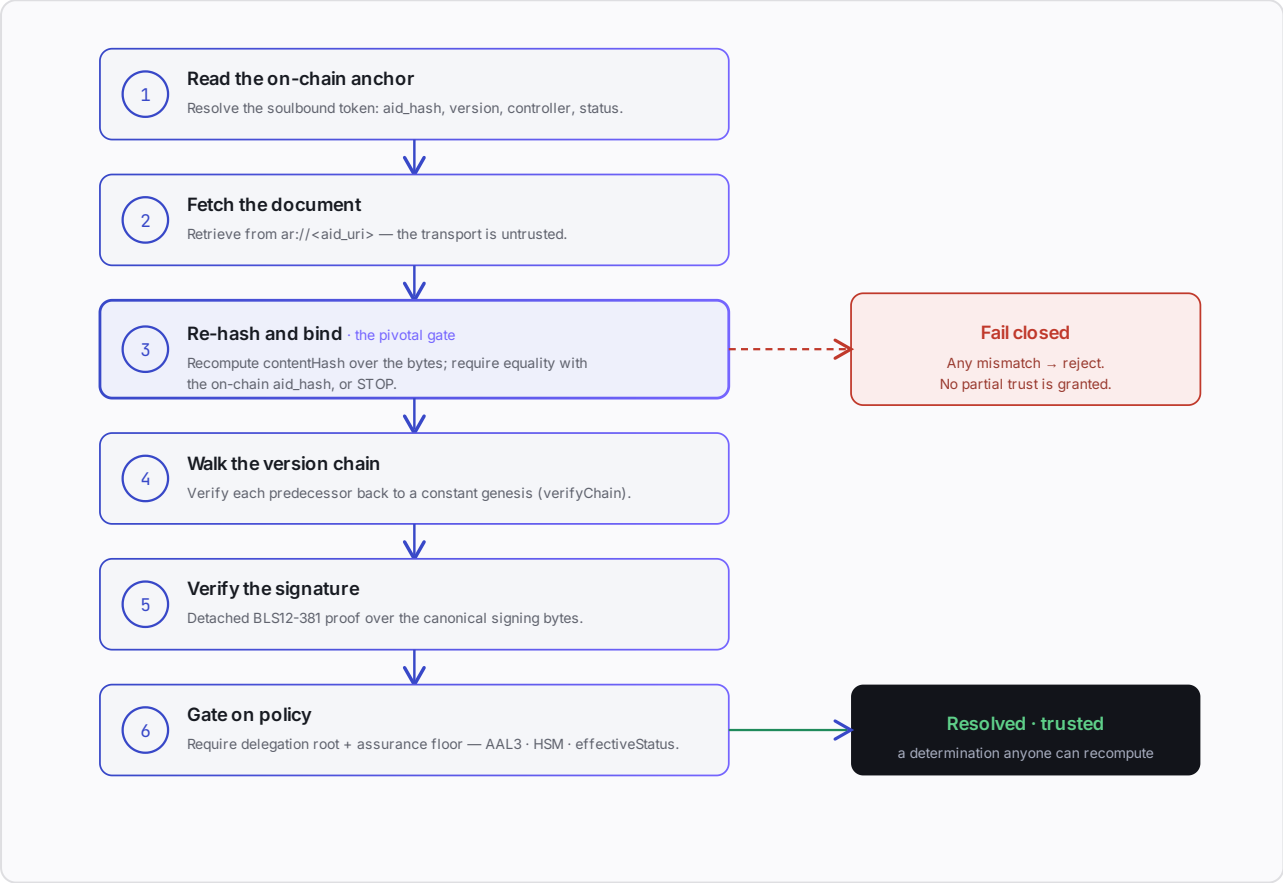

Figure 5 — The resolution pipeline. Step 3 is the pivot: a tampered document hashes to a different value than the on-chain commitment and is rejected there, always. Nothing downstream is trusted transitively; each layer checks the one below it.

THE BINDING RULE

The token could point anywhere, so a resolver re-hashes the fetched document against the on-chain commitment. The document could claim any predecessor, so the chain is walked link by link back to genesis. **Verification never trusts an indexer — it recomputes.**

SECTION 09 — THE PROTOCOL FAMILY

One protocol family, one job each.

Agent ID is the load-bearing base of a wider protocol family — the Tezos Agent Protocol (TAP) series — that extends the same doctrine upward from identity into authority, governance, and transparency. OAuth, DID, and X.509, reconceived for autonomous agents.

LAYER	SPECIFICATION	ROLE IN THE TRUST DECISION	STATUS
TAP-1/2	Identity & Anchor	The AID plus the soulbound FA2 anchor. What is this, and is this version authoritative?	SHIPPED
TAP-3	Authority & CPoA	The live, scoped mandate and the policy engine that evaluates an action against it.	PARTIAL
TAP-5	Governance	The controller plus a tz4 / BLS governance multisig. Who may change the identity, and how.	SHIPPED
TAP-8	Transparency	The append-only chain and five emitted events. What happened, and can it be replayed?	SHIPPED

DISCOVERY VS. GRANT

capabilities[] is a surface

Dotted namespaces register what an agent is *registered to attempt* — a discovery surface for counterparties, never a grant. Authority to act is always the live CPoA, default-deny at the point of action.

STANDARDS, ENFORCED

A check behind every claim

Where the protocol claims a standards property — W3C DID/VC, NIST SP 800-63, NCCoE, OWASP NHI — a validator or the compliance evaluator actually checks it. Citing a control without wiring its check is a defect.

STANDARDS ALIGNMENT — ENFORCED, NOT DECORATIVE

The AID is a `did:tz` with DID-core shapes and credentials as hash references (inline PII is structurally impossible); assurance floors are validator-enforced; provenance and delegation reuse PROV-O with a shipped JSON-LD context and Turtle ontology, deterministic without blank nodes.

SECTION 10 — THE ENGINEERING BASELINE

Every figure is reproducible from *one command*.

The pipeline is the argument. The reference implementation regenerates every example from source, reprojects the knowledge graph, replays the golden byte-equality vectors, re-verifies the chain and anchors, executes a deliberate tamper that must be rejected, and fails on any drift between committed artifacts and what the code generates. No figure in this paper is transcribed by hand.

```
# from a clean checkout
$ npm ci && npm run ci

✓ typecheck        strict · lint · format clean
✓ build:examples   regenerated from source (real hashes)
✓ build:kg         reprojected · strictVerify round-trip
✓ build:vectors    golden byte-equality vectors
✓ validate         AJV 2020-12 · chain · anchor · tamper
✓ test             55 passing (25 core + 17 KG + 13 vectors)
✓ drift gate       git diff — examples fixtures · exit 0

aid v1  contentHash  c2349d4a…e976b332
aid v1  genesisHash  1fe73207…ce70394f
aid v2  contentHash  4ee2ffbc…2829eed4
# v2.previousVersion.contentHash == v1  ·  genesis constant

graph   98 quads / 2 named graphs
commit  27408e57…30d7a5f  ✓ Oxigraph
! tamper injected  → rejected as required
```

55/55
tests passing — core, graph, and golden vectors

0
runtime dependencies in the SDK

1
command reproduces every figure

TWO CONFORMANCE GUARANTEES

Golden vectors let a second implementation prove byte-for-byte agreement without reading the code. **The artifact-drift gate** — proven in both directions, a hand-edit fails and the honest tree passes — means schemas, SDK, and artifacts can never diverge unnoticed. Releases publish with npm provenance, binding the tarball to the exact commit.

SECTION 11 — HONEST BOUNDARIES

What is not yet done, stated as plainly as what is.

A trust system that hides its own edges has not earned trust. Every stub, mock, and unaudited component is labeled — a mock is never presented as real. Phase 0, the engineering baseline, is complete; each remaining phase is gated on the one before it.

STATUS	BOUNDARY
PHASE 1	**The smart contract** is an uncompiled, unaudited, pre-deployment draft. Compilation, scenario tests for every authority path, and an independent audit precede Ghostnet origination.
PHASE 2	**Signature verification and the live Arweave client** sit behind their audited providers and fail closed until wired — the milestone is one real end-to-end identity on Ghostnet.
MOCK	**Example Arweave transaction IDs** are deterministic mocks and example proofs are absent. Every hash, chain link, and vector, by contrast, is real and drift-gated.
PHASE 4	**The w3id.org/praxis identifiers** await registration at stable URLs; the evaluator describes compliance — it does not certify it.

> **INVARIANT I11 — HONESTY ABOUT BOUNDARIES**
>
> The status labels in this paper are not editorial. They are drawn from the repository's vision-to-code cross-reference, and CI enforces the distinction: a box reads SHIPPED only if the pipeline exercises it today. Everything on the roadmap is named, phased, and gated.

SECTION 11.1 — THE INVESTMENT LOGIC, PRECISELY

Four properties of the architecture — each *checkable*.

For an investor, the thesis reduces to four properties of the architecture itself — none dependent on adoption narrative alone. Each is a structural consequence of the design decisions described in the preceding sections, not a projection.

THE MOAT IS A DISCIPLINE

Stateless recomputation, enforced standards, and status-labeled honesty are architectural commitments competitors cannot bolt on: a reputation registry cannot retrofit "no reputation," and a trusted-registry design cannot retrofit "indexers are never authoritative" without rebuilding.

NETWORK EFFECTS, NOT LOCK-IN

Capabilities are shared IRIs, so every new operator's agents land on the same nodes; independent indexes federate by merging quads, each side re-verifying. The graph grows more useful with each participant while no participant must trust another.

SWITCHING COSTS ARE EVIDENTIARY

An institution's anchored identity history — versions, delegations, revocations, events — is the asset it accumulates. Portable in principle, maximally sticky in practice: history cannot be re-created elsewhere, only continued here.

REGULATION IS THE DEMAND DRIVER

The design targets exactly what NIST NCCoE, SP 800-63, and the OWASP NHI work say institutions will be required to demonstrate — and demonstrates them with enforced validators and citable artifacts. This is what compliance looks like already-built.

WHAT AN INVESTOR IS SHOWN

The SDK, schemas, knowledge graph, golden vectors, and CI substrate are shipped and verified; the contract is an audit-gated draft; providers and the service layer are named phases with a Ghostnet existence-proof milestone. A **verification discipline with a roadmap** — not a demo with a story.

THE NEXT INFRASTRUCTURE QUESTION

The future economy will not simply contain AI agents. It will contain *verified* ones.

The foundational primitive for that world is not another model. It is identity, authority, provenance, and accountability — composed into a trust operating system for autonomous software. An identity that cannot show its accountable principal, its authority, the assurance of its keys, and the unbroken history of its versions has asked to be trusted. PRAXIS is built so it never has to ask.

The future will be verified.

Quantum Field Inc. · PRAXIS · Reference implementation @praxis/agent-id

SEMANTIC TRUST INFRASTRUCTURE · POWERED BY TEZOS · CONFIDENTIAL