PRAXIS ARCHITECTURE

Architecture Exposition

How Knowledge Graphs, Tezos, AI Agents, and the Trust Infrastructure Layer Compose into an Operational Trust System

Autonomous intelligence requires a trust substrate that separates computation, authority, verification, and accountability.

THE KNOWLEDGE GRAPH CREATES UNDERSTANDING · THE AI CREATES INTELLIGENCE · TEZOS CREATES FINALITY · PRAXIS CREATES TRUST.

Companion to the PRAXIS Agent ID Technical Overview (Third Edition)

Reference implementation @praxis/agent-id · Tezos L1 + Arweave · every claim status-labeled

Quantum Field Inc. · Powered by Tezos · The Future Will Be Verified. · July 2026

Document Control

Title	PRAXIS Architecture — Architecture Exposition
Purpose	The system-level exposition: how the layers interoperate into an operational trust system. An architecture narrative, not a module catalogue — the module catalogue is the reference package itself and its Technical Overview.
Audiences	External institutional and standards reviewers · internal engineering · investors. One document, because the credibility argument is identical for all three: every claim is status-labeled and independently checkable.
Companion documents	PRAXIS Agent ID — Technical Overview (**Third Edition**); `ARCHITECTURE.md` (the vision-to-code cross-reference inside the reference repository); `AGENT_BRIEF.txt` (the eleven invariants).
Reference implementation	`@praxis/agent-id` — TypeScript SDK, SmartPy FA2 contract (audit-gated draft), JSON Schemas, RDF ontology, golden vectors, CI with 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 in this document 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*.
Classification	Confidential — institutional, engineering, and investor review

Contents

I — The Architectural Thesis

PRAXIS is designed around a single architectural premise: **autonomous intelligence requires a trust substrate that separates computation, authority, verification, and accountability.** An AI model can reason. An agent runtime can execute. A blockchain can settle. A database can store information. None of these, individually, answers the fundamental question of the agentic economy:

> ***"How does another party know that this autonomous entity is authorized to act, within a defined scope, under a legally accountable principal, at this exact point in time?"***

PRAXIS answers by composing a layered trust architecture in which **each component does exactly one job**. AI agents perform actions. Knowledge graphs provide semantic understanding. Tezos anchors cryptographic commitments and decentralized verification. PRAXIS identity objects provide machine-readable legal identity. Attestation and governance layers define authority. Transparency logs preserve accountability. The result is not a product feature; it is a **trust operating system for autonomous software**.

The composition principle in one line: *the knowledge graph creates understanding; the AI creates intelligence; Tezos creates finality; PRAXIS creates trust.* Together they form the missing infrastructure layer between autonomous computation and human institutions.

1.1 How to read this document

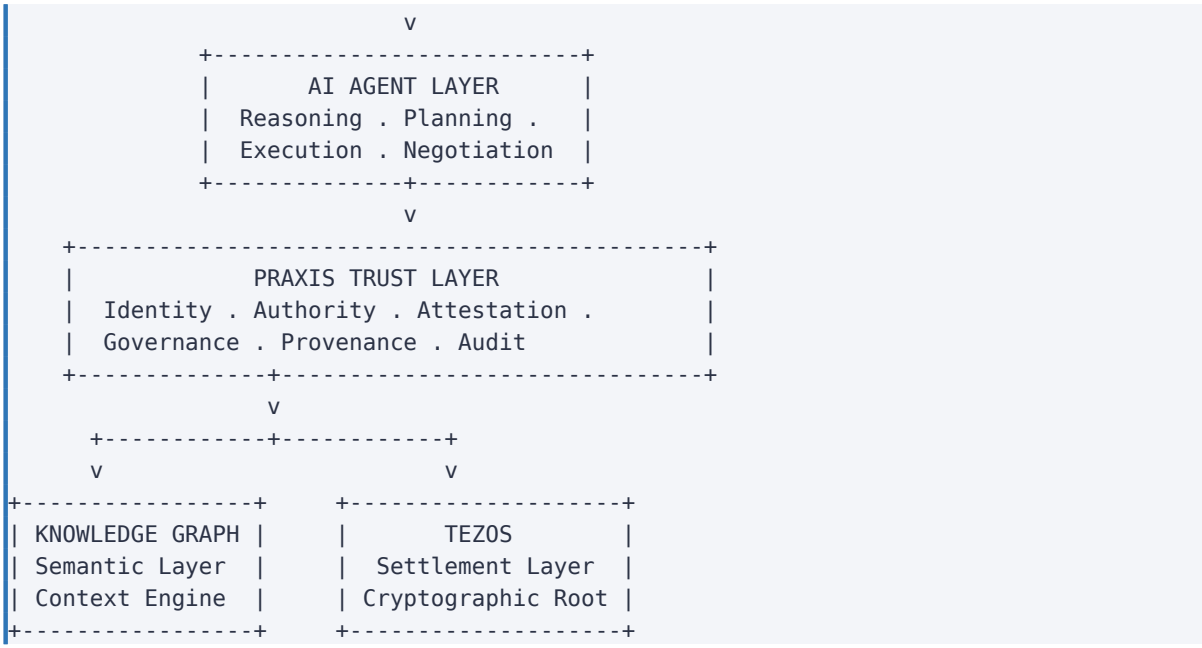
This is an architecture exposition, not a module catalogue. It explains how the parts interoperate into an operational system — and it holds itself to the reference implementation's own honesty discipline (invariant I11). Every claim about what exists today carries a status drawn directly from the repository's vision-to-code cross-reference, so any reader — examiner, engineer, or investor — can separate what is enforced now from what is scoped for a named roadmap phase.

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

Table 1 — the honesty legend. A box is SHIPPED only if the CI pipeline exercises it today.

II — The Core Architectural Model

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.



The two substrate layers are **not peers in kind**. The knowledge graph is a lens; the blockchain is the root of truth. That asymmetry — meaning on one side, proof on the other, with proof always able to overrule meaning — is the single most important property of the design, and Section VIII returns to it in full.

III — The AI Agent Layer

Role: execution. The AI agent is the actor. It performs reasoning, planning, API calls, negotiations, transactions, workflow execution, and communication. But the AI 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 2 — 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. The `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*.

```
Agent ID:          did:tz:mainnet:tz4...   (agent.procurement.acme)   [illustrative]
Current runtime:   frontier-model procurement engine, v4
Previous runtime:  prior-generation procurement engine, v3
```

```
Authority:          Purchase inventory up to $250,000   (live CPoA)
Controller:         Acme Corporation (Delaware C-corp)  tz1...
```

The runtime beneath the identity may change; the did:tz name, the authority, and the accountable controller do not. (Illustrative values; the agent is a tz4 account, the controller a tz1 legal principal, per TAP-1.)

This is why the AI agent is never the trust layer — it operates **inside** it. 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.

IV — The Agentic Identity Document

The Agentic Identity Document (AID) is the canonical identity artifact — not metadata, but a cryptographically verifiable legal identity record. It answers the four questions a skeptical counterparty must resolve before relying on an agent.

Who is this?

```
Agent:          did:tz:<network>:<tz4 account>
Controller:     the accountable legal principal (tz1..., with legal_entity_ref)
Status:         active | superseded | revoked   (+ point-in-time effectiveStatus)
```

What can it do?

The AID carries a `capabilities[]` **discovery surface** — dotted namespaces such as `payments.transfer`, `procurement.purchase`, `vendor.communication`. This is a registry of what the agent is *registered to attempt* — **never a grant of authority**. Actual authority to act is the live, scoped, revocable CPoA referenced by `cpoaAnchors[]`, default-deny at the point of action.

Under whose authority?

```
Acme Corporation                      (depth 0 — the accountable root)
     |  delegates to
     v
Procurement Department                (depth 1)
     |  delegates to
     v
AI Procurement Agent                  (delegation[] — depth-0 root ENFORCED)
```

What evidence exists?

```
v1  Created
v2  Updated permissions
v3  Rotated keys
v4  Revoked authority        (append-only; revocation terminal)
```

The delegation chain **must terminate in a depth-0 accountable root** — the anti-"unaccountable void" rule, enforced by `validateAid`, not merely suggested. An identity that cannot name a legally answerable principal is rejected at validation. This is what makes attribution a property of construction rather than an exercise in forensic reconstruction.

V — Tezos: The Trust Anchor

Tezos functions as the decentralized verification substrate. Its role is not to store everything — that would be inefficient and unnecessary. It stores only the commitments that matter, and it becomes the immutable referee that answers a single question: **what identity artifact was authoritative at this point in time?** The governing principle is *store proof on-chain, store complexity off-chain* — invariant I2, enforced by construction. The soulbound FA2 token carries only a commitment; the document lives elsewhere; the graph is derived.

```
Tezos — Agent ID NFT (soulbound, TZIP-12/16/21)
{
  aid_hash:     c2349d4a69dacea6d742a7371bfb1b314bd08ed98e2d06a0296b34d9e976b332
  aid_uri:      ar://<txid>          // where to fetch the document
  aid_version:  1                    // which version is authoritative
  genesis_hash: 1fe732078376b28a24aa09973d84aa876b64b18050740833663ecc22ce70394f
  controller:   tz1KqTpEZ7Yob7QbPE4Hy4Wo8fHG8LhKxZSx   // who is accountable
  soulbound:    true
  status:       active               // is it live
}
```

The on-chain anchor holds a commitment, never identity data. These are the real, coherent v1 values from the shipped example chain — reproducible from the pipeline.

On-chain, the contract stores the agent identifier, the current identity version, the content hash, the genesis hash, the controller reference, and the status — and it **emits a TAP-8 event for every state change**, so the full history is replayable from the chain itself (events live in chain history, not contract storage; nothing mutates silently). Everything else — credentials, attestations, provenance, policies, compliance blocks — is too large and too dynamic to belong on a chain, and is committed to transitively through a single hash.

Anchor element	Status	Where it lives
Identity commitments	SHIPPED	`aid_hash` + `genesis_hash` in FA2 metadata; write-once genesis at mint. Compilation, audit, origination: ROADMAP(1).
Governance events	SHIPPED	Five TAP-8 events: `registration`, `metadata_update`, `identity_revoked`, and the two handoff events. No silent mutation.
Authorization proofs	PARTIAL	The `proof` envelope is shipped (excluded from both hashes); BLS12-381 aggregate verification behind it fails closed until wired — ROADMAP(2).
Evidence records	SHIPPED	The version chain on Arweave plus deterministic evidentiary rendering. Live Arweave publisher behind the fail-closed interface: ROADMAP(2).

Table 3 — the Tezos anchor, labeled honestly. The contract is an audit-gated draft; the commitment model it enforces is fixed.

VI — Arweave: Permanent Identity Storage

The complete identity documents live off-chain, on Arweave, written once and immutably. Identity documents carry credentials, attestations, provenance, and compliance information — content that is too

large and too dynamic for a blockchain, but that must nonetheless survive independently of any operator's server. Arweave buys survivability; the chain buys liveness and control; the pairing is deliberate.

```
      Tezos                          Arweave
  ---------------                 -----------------
  Hash commitment   -------->     Full AID record
  { aid_hash }        binds       (source of truth)
```

The storage layer is never trusted; it is verified. The verification is mechanical and runs on public artifacts alone:

```
1.  Fetch the AID from ar://<aid_uri>          (source is untrusted)
2.  Recompute aidContentHash over the fetched document
3.  Require equality with the on-chain aid_hash    -> else STOP
4.  Walk the version chain back to genesis (verifyChain)
5.  Accept or reject
```

The binding rule. The NFT could point anywhere, so the resolver re-hashes the fetched document and compares it to the on-chain commitment. The document could claim any predecessor, so the chain is verified link by link back to genesis, including the constancy of the genesis root. A resolver that checks only one of these has not verified identity.

VII — The Knowledge Graph Layer

A conventional database stores facts. A knowledge graph stores **meaning**. The knowledge graph is where PRAXIS becomes fundamentally different from traditional identity systems. Traditional identity asks a narrow question — “does this credential match?” PRAXIS asks a semantic one: *what is the complete relationship between this agent, its authority, its organization, its policies, and its actions?* The graph transforms static identity documents into a machine-readable model of an institution.

```
Acme Corporation
     |  owns / controls
     v
Procurement Agent
     |  authorized under
     v
Corporate Procurement Policy
     |  permits
     v
Purchase Electronics   (maximum $250,000)
```

The graph carries context no isolated credential can: not the bare fact, but the institutional meaning around it. (Illustrative relationships; policy semantics belong to TAP-3.)

Concretely, the projection is deterministic and pure. `documentToQuads` maps a validated AID into RDF quads, each emitted into a **named graph whose IRI embeds the asserting version's content hash** — `urn:praxis:sha256:<contentHash>`. Because that is the very hash the NFT anchors, *graph provenance is the on-chain commitment*: a SPARQL result returns, in its `?g` position, the anchored hash behind every answer. Vocabulary reuses before it invents — identities are did:tz IRIs, lineage and delegation ride W3C PROV-O, and capabilities are shared IRIs so two operators' `payments.transfer` meet at one node.

7.1 How the graph interacts with Tezos

The graph is explicitly **not authoritative**. This is a critical architectural principle, not an implementation detail. The graph is a semantic index; the blockchain remains the trust root. A discovery query — "find all agents authorized by Acme capable of initiating payments above $100,000" — is answered quickly by the graph, but the answer is advisory until re-verified.

```
KG claim  (fast, advisory)
   |
   v
Retrieve the source AID
   |
   v
Recompute contentHash  ->  compare on-chain anchor   (verifyAgainstAnchor)
   |
   v
Re-project and require set containment               (strictVerify)
   |
   v
Accept / reject
```

Verification is stronger than a label check: it is **set containment against re-projection**. A graph label only names a document; it does not enumerate its quads, so a malicious indexer could attach forged triples under a perfectly genuine label. `verifyThenTrust` therefore obtains the anchored document, requires it to re-hash to the label, re-projects it, and admits a presented quad only if it is a member of the canonical projection. `strictVerify` fails closed on the first rejection. What the KG test suite proves: a forged capability under a valid label is rejected; a one-byte assurance downgrade (`aal3` → `aal1`) is rejected; unknown graphs are rejected rather than skipped; a lying resolver is caught; and honest subsets — exactly what query results are — verify cleanly.

Because no blank nodes are emitted, the sorted N-Quads are byte-stable and the containment check is exact; `quadsCommitment()` therefore yields one SHA-256 for an entire graph — a value a contract, CPoA record, or transparency-log entry can commit to. The shipped artifacts are proven against a real engine: **98 quads across 2 content-hash-named graphs, graph commitment `27408e57…30d7a5f`, loading and querying correctly in Oxigraph.**

VIII — How the Layers Compose

Identity is necessary but not sufficient. The full trust decision an institution must make decomposes cleanly across the PRAXIS stack, and Agent ID is the load-bearing base on which the higher layers rest.

The question an institution asks	Layer	Answered by
What is this entity, and is this version authoritative?	Identity (TAP-1/2)	The AID + soulbound anchor.
What, precisely, is it authorized to do right now?	Authority (TAP-3)	CPoA records referenced by `cpoaAnchors[]`.
Who governs changes to it?	Governance (TAP-5)	The controller + governance multisig.
What happened, and can it be replayed?	Transparency (TAP-8)	The append-only chain + emitted events.

Table 4 — the trust decision, decomposed. Identity says who and what; authority says whether, right now.

8.1 The graph proposes; the anchors confirm

The most important composition rule governs the relationship between the semantic layer and the truth layer. It is tempting to say the knowledge graph "answers" authority queries. That phrasing is precise only with its second half attached: the graph is a **lens, never a source of truth**. It is how a verifier *discovers* a claimed authority chain in milliseconds; the claim is *confirmed* only by re-verification against the anchored AID — containment against re-projection — and, for live authority, the TAP-3 CPoA record. Without this rule, the architecture would quietly reintroduce "trust the registry" — the exact failure mode it exists to eliminate. With it, a SPARQL answer arrives carrying, in its graph position, the precise on-chain commitment that proves it. The graph accelerates discovery; the anchor supplies finality; neither is asked to do the other's job.

8.2 Evidence accumulates; trust is recomputed

A second reconciliation is equally load-bearing. The trust loop ends "knowledge graph updated → future decisions improved." Read as *evidence accumulating*, that is exactly this system. Read as *trust accruing*, it would violate the first invariant — no reputation, ever, because accrued trust can be farmed, purchased, or Sybil-inflated. The architecture fixes the reading unambiguously: **every decision is evaluated from scratch** against anchored identity, live authority, and policy at an explicit instant. There is no balance that grows, no score, no endorsement count — and by standing invariant there never will be. What improves over time is the *evidence base*, never a trust number. This is also the differentiator no reputation-based registry can state: the value that compounds is replayable evidence; the trust decision itself is **stateless**, recomputed on demand from artifacts anyone can independently check.

IX — The Agent Action Flow

The architecture is best seen in motion. Consider an enterprise procurement agent instructed to purchase $150,000 of equipment. Trust is established not by one check but by a sequence in which each layer contributes exactly what it is authoritative for — and each step is labeled for what exists today.

Step	Status	What happens
1 · Task	`SHIPPED`	The agent receives an instruction: purchase $150,000 of equipment.
2 · Identity	`SHIPPED`	Resolve the token, fetch the AID from `ar://`, re-hash against the anchor (`verifyAgainstAnchor`). *Who am I, and is this version authoritative?*
3 · Discovery	`PARTIAL`	The graph proposes the authority chain: agent → Acme Procurement → policy → $250,000 limit. Advisory until confirmed.
4 · Attestation	`PARTIAL`	The CPoA layer confirms live authority: active, in-scope, unexpired, controller verified. Discovery is confirmed by the anchored AID and the TAP-3 record. BLS verification: ROADMAP(2).
5 · Policy	`VISION`	The Policy Engine evaluates the action against budget, vendor, and compliance rules. Its inputs are shipped fields; the evaluator itself belongs to TAP-3.
6 · Execution	`VISION`	The agent's own runtime completes the transaction — outside this trust layer by definition.

Step	Status	What happens
7 · Transparency	`SHIPPED`	PRAXIS records the evidence — agent, action, authority reference, policy version, timestamp, outcome — as an append-only, anchored record.

Table 5 — one action, end to end, labeled honestly. The trust layer verifies and records; it never reasons or executes.

The shape of the flow is the argument. Identity and evidence are enforced today; discovery and attestation have shipped seams with named remainders; policy and execution belong deliberately to layers above and beside this one. No step trusts the issuer, and every step that matters for a dispute — who acted, under what authority, with what outcome — is anchored and replayable.

X — The Strategic Stack, Located

The deepest architectural insight is that the future internet will require not only a data layer but a **trust-graph layer**. The strategic stack runs from raw data up to verification, and PRAXIS connects the layers. What follows is that stack with each rung located honestly against the reference implementation — the discipline that makes the whole credible.

Layer	Status	Where it lives
Data	`SHIPPED`	Arweave documents and on-chain anchors.
Knowledge	`SHIPPED`	The KG projection and ontology.
Meaning	`PARTIAL`	PROV-O delegation/lineage and shared capability IRIs are shipped; the full institutional ontology beyond identity is VISION.
Authority	`PARTIAL`	Controller, delegation root, and CPoA anchors are shipped seams; live CPoA evaluation and the policy engine are VISION (TAP-3).
Action	`VISION`	The agent runtime — out of scope by design.
Evidence	`SHIPPED`	The version chain, TAP-8 events, and evidentiary rendering.
Verification	`SHIPPED`	`validateAid` / `verifyChain` / `verifyAgainstAnchor` / `strictVerify` shipped; the hosted resolver and verifier CLI are ROADMAP(3).

Table 6 — the strategic stack, located. SHIPPED means enforced by the CI pipeline today; nothing unbuilt is presented as real.

10.1 A note on “Tezos X”

Outward material brands the anchor layer **Tezos X** — the platform's forward roadmap. The normative surfaces bind to what is verifiable today: Tezos L1 semantics, the `mainnet` and `ghostnet` networks, and FA2/TZIP-12/16/21, with an FA2.1 (TZIP-26) migration note in the contract. “Tezos X” refers to this anchor layer as it evolves; the schemas name concrete networks so an auditor always knows exactly which chain a given contract lives on.

XI — Why This Architecture Creates a New Category

Most AI infrastructure competes on intelligence, models, agents, and automation. Most blockchain infrastructure competes on settlement, assets, and transactions. PRAXIS operates at the intersection none of them occupies:

```
Intelligence + Authorization + Accountability
     + Semantic Understanding + Cryptographic Verification
------------------------------------------------------------
=  Agentic Trust Infrastructure
```

The agentic economy does not fail because machines cannot act. It fails if nobody can prove **who acted, why they were allowed to act, what constrained them, and who is responsible afterward**. PRAXIS turns autonomous action from an assumption into a verifiable event. The layers are not interchangeable and not individually sufficient: AI can act but cannot prove its own legitimacy; a blockchain can verify records but cannot understand their meaning; a knowledge graph can represent relationships but cannot anchor them immutably; identity can authenticate but cannot govern autonomous behavior. Composed, they produce something none yields alone — semantic, cryptographic, and institutional trust in a single architecture.

11.1 The investment logic, precisely

For an investor, the thesis reduces to four properties of the architecture itself — each one checkable, none dependent on adoption narrative alone:

- **The moat is a discipline, not a feature.** Stateless trust recomputation, enforced standards alignment, and status-labeled honesty are architectural commitments competitors cannot bolt on: a reputation-based registry cannot retrofit “no reputation,” and a trusted-registry design cannot retrofit “indexers are never authoritative” without rebuilding.
- **Network effects run through shared semantics, not lock-in.** Capabilities are shared IRIs, so every new operator's agents land on the same nodes; independently built indexes federate by merging quads, each side re-verifying for itself. The graph grows more useful with each participant while no participant must trust another.
- **Switching costs are evidentiary, not contractual.** An institution's anchored identity history — versions, delegations, revocations, events — is the asset it accumulates. It is portable in principle (public artifacts) and yet maximally sticky in practice: history cannot be re-created elsewhere, only continued here.
- **Regulatory alignment 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 it demonstrates them with enforced validators and citable artifacts, not attestations. When agent-identity requirements harden, this is what compliance looks like already-built.

The current state is stated with the same discipline (invariant I11): the SDK, schemas, knowledge graph, golden vectors, and CI substrate are shipped and verified — 55/55 tests, drift gate proven, figures reproducible from one command; the contract is an audit-gated draft; providers and the service layer are named phases with a Ghostnet existence-proof milestone. An investor is being shown a verification discipline with a roadmap, not a demo with a story.

11.2 Closing

The knowledge graph creates understanding. The AI creates intelligence. Tezos creates finality. PRAXIS creates trust. The future economy will not simply contain AI agents — it will contain **verified** AI agents. And 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. That is the architecture PRAXIS provides.

Quantum Field Inc. · PRAXIS — The Trust Layer for the Agentic Economy · The Future Will Be Verified.