PERASYS LABS · WHITEPAPER

Logos

The authority and accountability layer for AI that acts.

Perasys Labs · June 2026 · A technical and plain-language overview

Summary

Logos is a governed operating environment for AI-mediated action. It sits between AI systems and the actions they can take, enforcing that consequential actions are authorized by an accountable party, attributing every action to its author, and recording every prediction, authorization, action, and outcome on an append-only, tamper-evident ledger. Its founding principle is that capability is not authority: an AI may be arbitrarily capable, yet the authority to act where consequences are real belongs to the party who bears them. This paper describes the principle, the computational model, the guarantees Logos proves, and the architecture that delivers them, in terms intended to be legible to engineers, operators, buyers, and compliance readers alike. It describes guarantees and structure, not implementation.

IN PLAIN WORDS

AI systems are starting to *do* things, not just answer questions: sending invoices, moving money, updating records, contacting customers. The convenience is real, and so is the risk, because a mistake now *happens* instead of merely being *suggested*.

Logos sits between an AI and the actions it can take. The AI can recommend anything. But when an action carries real consequences, Logos requires a named person to approve it, and writes down what happened in a record that cannot be altered afterward. **Nothing important happens without a person knowing, and you can prove it later.**

Capability is not authority. An AI system may become arbitrarily capable, but capability alone never grants the right to act where the consequences are real. The authority to act belongs to the party who bears those consequences, which today means a human being or the organization they represent. Logos is built to keep that line intact: machines observe, calculate, predict, and advise; humans grant, deny, delegate, revoke, and authorize.

The problem, stated carefully

Two facts now hold at once. First, AI systems are increasingly capable of taking actions in the world rather than only producing text. Second, the controls most organizations have for AI were designed for a world in which a human was the actor and the AI was a tool that informed them. When the AI becomes the actor, the

controls built around human sessions and human review no longer see the relevant events, and errors can scale at machine speed before a person notices.

The naive response is a binary one: either lock the AI down so it can do nothing consequential, or trust it fully and hope. Both fail. Locking it down destroys the value; trusting it fully reproduces the risk the controls were meant to manage. The correct response is graduated: distinguish the action's stakes, let routine actions proceed, and route consequential ones to an accountable human, while recording all of it. This is the distinction industry analysis now identifies as the difference between enterprises that succeed and fail with agents: separating an agent's ability to act from the scope it is granted [4].

> **For the non-specialist:** the goal is not to slow AI down. It is to let it move fast on what does not matter and pause on what does, with a clear, permanent record of who decided what.

Why this matters now

Through 2026, enterprises moved from AI that *answers* to AI that *acts*. Industry analysts report that the majority of organizations are now deploying autonomous agents, though most deployments remain early-stage rather than full production [1][2]. The governance gap is the dominant risk in the category: Gartner predicts that more than 40% of agentic AI projects will be canceled by the end of 2027 due to escalating costs, unclear business value, or inadequate risk controls [3], and separately that 40% of enterprises will demote or decommission autonomous agents by 2027 because of governance gaps surfaced only after production incidents [4]. Gartner identifies the root cause precisely: failures occur when organizations do not distinguish an agent's *ability to act* from the *scope of access* it is granted [4].

That distinction is the founding idea of Logos, stated before the market named it: capability is not authority. In parallel, regulation has set a hard line. Under the EU AI Act, most obligations, including high-risk system requirements and transparency duties, apply from August 2, 2026, with full rollout by August 2, 2027 [5], and Article 14 requires high-risk systems to be designed for effective human oversight [6]. Organizations increasingly need not only to exercise oversight but to *demonstrate* it.

> **For the non-specialist:** the law and the market are now asking the same question Logos was built to answer: how do you let AI act quickly on the routine things while keeping a person genuinely in control of the consequential ones, and prove afterward that you did?

The model

Most AI products are arranged so the model is the center of the system:

```
User  ->  Model  ->  Action
```

Logos inverts this. The record is the center, and every action is a governed transition recorded on it:

```
Observation
   -> Recommendation        (the AI proposes)
   -> Authorization         (an accountable human decides)
   -> Action                (only what was authorized)
   -> Outcome               (what actually happened)
   -> Recorded Consequence  (immutable, attributable)
```

The AI becomes a component *inside* the recommendation step, not the architecture itself. Authority enters only at the authorization step, and only from an accountable party. Nothing reaches Action without passing through Authorization.

> **For engineers:** this is event sourcing with a policy enforcement point on the write path. State is derived from an append-only, hash-chained event log rather than mutable rows; reads are projections over that log (a CQRS shape); and the authorization check is the single gate through which any state-changing commit must pass.

Accountability as the organizing principle

Logos treats accountability not as a logging feature added at the end, but as the organizing principle of the architecture. An action is meaningful only if some party can be held to its consequences. That party, the accountable principal, is the only source of authority in the system. An AI can recommend; it cannot authorize, because it bears no consequence. A human using an AI is an accountable party; the AI alone is not. This is why the recommending system and the authorizing party are kept structurally separate, and why all authority must trace to an accountable principal.

From this principle the rest follows. If authority must trace to an accountable party, then every authorization must be attributable to one, and every attribution must be verifiable, or the accountability is hollow. If actions have consequences that matter, the record of them must be tamper-evident, or it cannot be relied on after the fact. The invariants are not an arbitrary checklist; they are the minimal set of properties that make accountability real rather than asserted.

What Logos guarantees

The difference between a governance *claim* and a governance *system* is whether its guarantees can be proven. Most products in this space assert that they log actions and require approvals. Logos states its guarantees as formal invariants and proves them the way a database proves its consistency guarantees or a protocol proves its correctness: as properties that must hold across the entire space of possible inputs, verified by testing that actively tries and fails to break them.

INV-1	Only admissible references may participate in computation.
INV-2	Only authorized commits may modify state.
INV-3	Every commit extends exactly one valid chain (tamper-evident).

INV-4	Authorization comes from an accountable party, never from AI capability.
INV-5	The recommending system and the authorizing party remain separate.
INV-6	All authority traces to an accountable party.
INV-7	Every state change is attributable.
INV-8	Every attribution is verifiable.

These eight properties are the proof obligation of the system. They are verified by property-based testing across many thousands of generated cases per invariant, including adversarial ones, with no counterexample found. The verification is reproducible: any failure would replay exactly.

> **Why this is the point:** a platform earns the right to be built upon when its guarantees hold by construction, from within, regardless of the application running on top, the way TCP is proven by its protocol invariants rather than by any one successful stream. Logos is designed to be proven the same way. Invariant 4 is the load-bearing one: it is the formal statement of *capability is not authority*, and it is enforced in the running system, not merely asserted in documentation.

Architecture

Logos is a composition of five subsystems coordinated by a control plane. A deliberate design rule holds throughout: no single subsystem is Logos. The control plane composes them; each is replaceable without redefining the whole. This prevents the common failure of collapsing the system into its policy engine, which can fail, be misconfigured, or be replaced.

> **The ontology, stated plainly.** Logos is not a single subsystem. Logos is the operating environment that composes Specification, Telemetry, Control, Principal, and Store into one accountable system. Each subsystem is contained by Logos; none of them is Logos. The authority and accountability layer described throughout this paper is an *emergent property of that composition*, not the behavior of any one part. This distinction is load-bearing: because the control plane is a component rather than the whole, a control-plane failure is contained rather than fatal. The Specification still defines expected state, the Store still holds what happened, and a replacement control plane re-derives state from the record. A system that equated itself with its policy engine would have no layer at which integrity survives that engine's failure.

SUBSYSTEM	ROLE, IN STANDARD TERMS
Specification	the invariants and the schema the system holds itself to
Telemetry	measurement and observability over current state (read-only)
Control	policy enforcement: the authorization check on every action
Principal	

	the accountable party who holds and exercises authority (the access-control term for an authenticated actor with standing)
Store	the append-only, hash-chained, tamper-evident record

The core abstraction underneath is a single primitive: a *reference*, any state used to justify a future state. Inputs, commands, signals, and records differ only in origin; all are references. The pipeline is uniform: reference, compute, authorized commit, consequence. The system governs which references are admissible and which commits are permitted; everything between is the work of the intelligence, which Logos does not perform and does not need to inspect.

> **On form factor:** the core is a dependency-free runtime with no network or server assumption built in, so it can be embedded where it is needed: in a local-first desktop application, an on-premise service, or a hosted control plane. Deployment topology is a choice the adopter makes, not a property of Logos.

> **Confidentiality note:** this paper describes what Logos guarantees and how it is structured, not how its internal mechanisms are implemented. The specific schema, the standing and delegation logic, and the methods by which the system measures and prioritizes are deliberately omitted.

Where it sits

The clearest way to place Logos is among runtimes that govern a layer of a system:

```
Linux       governs processes
Kubernetes  governs containers
Temporal    governs workflows
OPA         governs policy
Logos       governs AI-mediated action
```

Logos is not an agent platform and not a model. It is the authorization and accountability layer beneath whatever agents or models an organization runs. It is adjacent to AI security tools that watch what people do with AI, and to guardrails that filter what models say, but it occupies a distinct position: the authorization of, and the permanent record of, what an AI is about to *do*. The category is honestly contested as of 2026, with open-source and commercial entrants addressing the action boundary. Logos's distinguishing commitment is that its guarantees are formally proven rather than asserted, and that authority is modeled as a first-class, traceable, accountable thing rather than a configuration flag.

What this paper does not claim

Three honesties keep the document trustworthy. First, the category is contested: as of 2026, open-source and commercial systems address the action boundary, and Logos does not claim to have invented it. Its distinguishing commitments are that its guarantees are formally proven rather than asserted, and that authority

is a first-class, traceable, accountable object rather than a configuration setting. Second, human oversight has a known failure mode, automation bias, where reviewers trust the system more than warranted; Logos's graduated design, which reserves human attention for the consequential few rather than demanding it on everything, is a response to this, not an exemption from it. Third, this paper deliberately omits implementation: the schema, the standing and delegation logic, and the internal measurement and prioritization methods are not described here.

Machines may observe, calculate, predict, and advise.
Humans retain the authority to grant, deny, delegate, revoke, and authorize.
Every prediction, authorization, action, and outcome is recorded, so that intelligent systems remain accountable to the consequences they produce.

References

1. KPMG, *AI Quarterly Pulse Survey* (2026): more than half of organizations actively deploying AI agents. kpmg.com
2. Forrester / enterprise survey coverage (2026): broad agentic AI adoption reported, with most implementations still at pilot stage. itpro.com
3. Gartner, "Over 40% of Agentic AI Projects Will Be Canceled by End of 2027" (June 2025). gartner.com
4. Gartner, "Applying Uniform Governance Across AI Agents Will Lead to Enterprise AI Agent Failure" (May 26, 2026). gartner.com
5. European Commission, EU AI Act implementation timeline: most obligations apply from August 2, 2026; full rollout by August 2, 2027. ai-act-service-desk.ec.europa.eu
6. EU AI Act, Article 14 (human oversight requirements for high-risk systems). artificialintelligenceact.eu

Market and regulatory statements above are attributed to their sources and were current as of June 2026; figures and timelines should be verified against the primary sources before reuse. This document describes architecture and guarantees, not implementation.