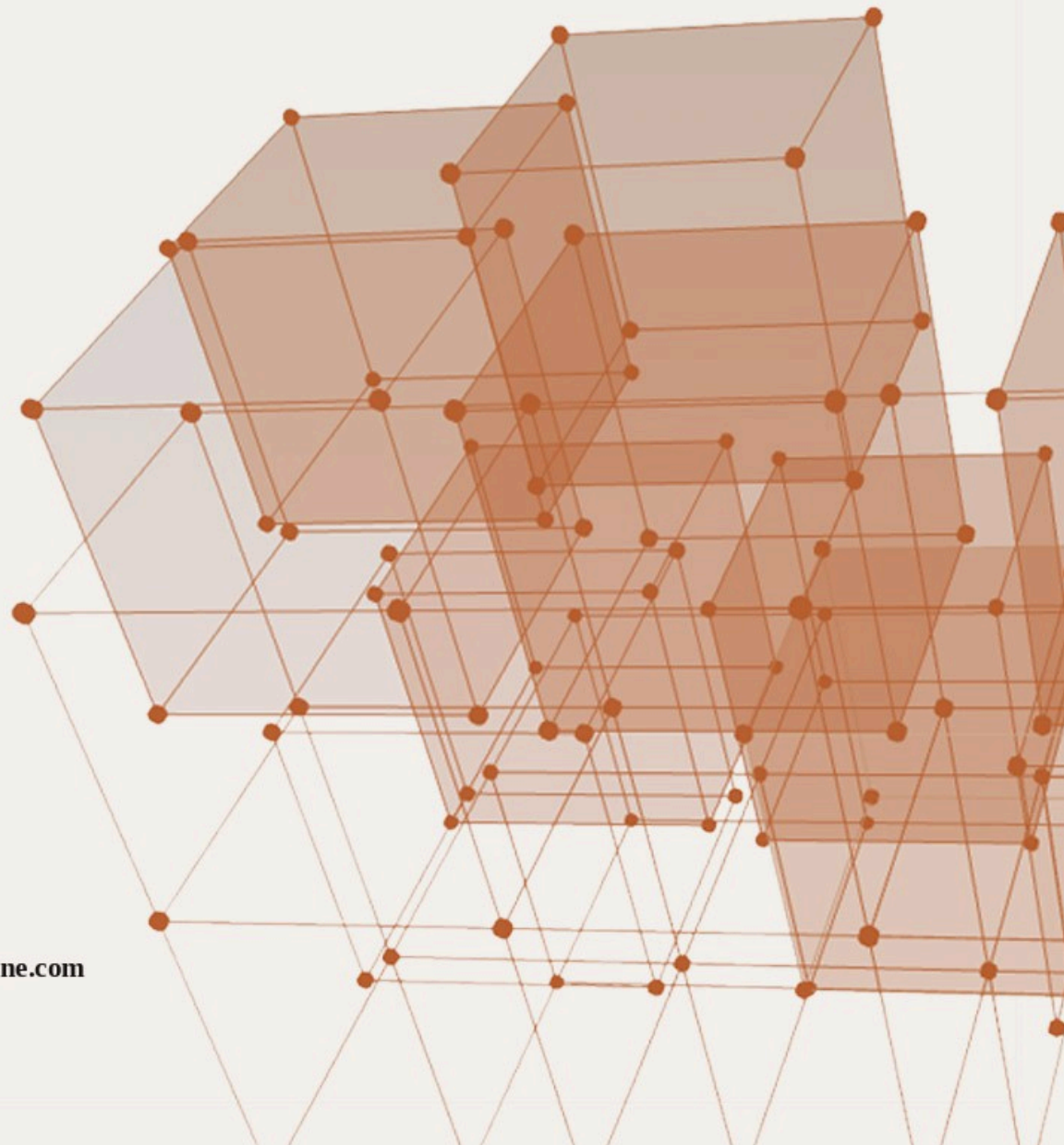


WHITEPAPER

The Authority Layer

Feb 2026 · CognaOne.com





WHITEPAPER • FEBRUARY 2026 • V3.1

The Authority Layer

Why AI agents need authorisation, not just observation.

Daniel Mackle • CognaOne • cognaone.com

The AI infrastructure market has built extensively around observation, orchestration, and memory. None of them answer the runtime question that becomes unavoidable once agents start taking meaningful action: should this action be allowed to happen? This paper defines the authorisation layer, explains why it is architecturally distinct from all existing categories, and describes how CognaOne implements it.

Every serious system that handles consequential actions eventually lands on the same principle: nothing executes until it is authorised.

CONTENTS

The problem

Why this matters now

What CognaOne is

How authorisation works

Human oversight with real authority

What the system produces

How it integrates

Where the industry is moving

The market gap

The business case

From observation to authorisation

The problem

Every serious system that handles consequential actions eventually lands on the same principle: nothing executes until it is authorised.

Payments do not clear just because a system wants them to. Medical systems do not act just because an instruction exists. Industrial systems do not move from signal to actuation without control points in between. In high-stakes environments, intent alone is never enough. There is always a layer between intent and execution.

AI agents are now operating in environments where that same principle matters. They approve transactions, modify records, write and deploy code, trigger workflows, send communications, and interact with external tools. But much of the infrastructure around them is still designed for observation. It records what happened, raises alerts, and produces logs after the fact.

That is useful. It is not enough. What is missing is authorisation: the layer that sits between what an agent intends to do and what it is actually allowed to do, and makes a binding decision before execution.

That is the layer CognaOne is built to provide.

Why this matters now

The issue is no longer theoretical.

Across enterprise copilots, coding agents, internal AI assistants, tool-connected workflows, and multi-agent systems, the same control gap keeps showing up in different forms. Systems act outside intended scope. They act on incomplete or unstable state. They lose critical instructions across longer workflows. They inherit risk from tools and integrations. And when something goes wrong, most organisations can explain it only after the fact.

The names of the incidents will change over time. The pattern will not.

That pattern is simple: the system had the ability to act, but there was no reliable layer determining whether that action should be allowed to happen: in that context, on that state, under that policy, at that moment.

Observation can tell you what went wrong. Authorisation can stop it before it happens.

What CognaOne is

CognaOne is the action authorisation layer for AI agents.

It does not replace orchestration, memory, or observability. Those systems still matter. They help agents coordinate, retain context, and surface telemetry. But none of them answers the core runtime question:

Should this action be allowed to execute right now?

CognaOne is designed to answer that question before side effects occur. This is not a dashboard. It is not a logging product. It is not just policy storage. It is the runtime control point that evaluates agent actions against context, state, and policy before they reach execution.

How authorisation works

CognaOne approaches the problem through four linked stages.

Context binding. Every action is bound to the specific session, workflow, scope, and operating context that produced it. Agentic systems drift easily when context is loose. Authorisation only works if the system knows which user, task, thread, policy surface, and workflow boundary an action belongs to.

Belief normalisation. Agents express meaning variably. The same underlying claim can appear in different language, different formats, or different levels of confidence. Before a system can govern action properly, it needs to normalise what the agent appears to believe into something canonical enough to evaluate. This is one of the core differences between generic workflow control and actual authorisation for agents: the system is not just checking an API call, it is checking an action against interpreted state that must first be made stable enough to reason over.

Conflict detection. Once state is normalised, the runtime evaluates it for contradictions, constraint violations, missing preconditions, scope breaks, or policy conflicts. This is where unstable belief meets real operating boundaries. The outcome is not a probabilistic warning. It is a decision surface.

Authorisation decision. If the action is within scope, consistent with policy, and supported by active verified state, it proceeds. If not, it is declined, escalated, or routed for approval depending on the configured control path. The important property here is determinism. The governance layer does not drift because the model changes. It does not weaken because prompts get more persuasive. It does not collapse because a workflow gets more complex.

Reasoning can remain flexible. Action control cannot.

Human oversight with real authority

Human oversight is often described too vaguely.

In practice, it only matters if humans retain real authority over what an agent is allowed to do. Not a person somewhere in the loop. A person with a real control surface.

That means a human can define scope, require approval for sensitive actions, interrupt execution, escalate higher-risk paths, and review a durable record of what the system attempted and why, before and after the fact.

Many AI systems today technically include a human somewhere in the process while still leaving the real action path too open. A dashboard is not oversight. A post-event log is not oversight. A notification after execution is not oversight. Meaningful oversight means the human still holds authority over consequential action.

CognaOne is built around that model.

What the system produces

When authorisation is treated as a first-class runtime layer, three things become possible as a structural product outcome.

Verified state. Agents do not act on raw, unstable, or contradictory state without scrutiny. The system normalises, compares, and evaluates what is being acted on before anything executes.

Authorised actions. Actions do not execute simply because a model produced them or because a tool call is technically possible. They execute because they passed a control point tied to context, state, and policy.

Signed receipts. Every meaningful decision generates a durable record of what was believed, what was attempted, what was approved or declined, and why. That record is created at the moment of decision, not reconstructed later from fragmented telemetry.

Together, these outputs turn agent control from soft supervision into operational governance.

How it integrates

The authorisation layer is designed to sit beneath existing infrastructure, not force organisations to replace it.

CognaOne integrates through standard interfaces: REST API, MCP, and webhooks. Organisations keep the orchestration systems, memory stores, and observability stacks they already use. The missing layer is added underneath: the point where action is evaluated before execution.

This also means governance becomes more durable across model changes. If a workflow uses one model today and another tomorrow, the control layer does not have to move with every swap. The authorisation posture remains stable even as reasoning layers evolve.

Where the industry is moving

Across AI regulation, enterprise governance programmes, standards work, and risk frameworks, the same expectations are appearing again and again: stronger human oversight, bounded autonomy, clearer intervention paths, better auditability, tighter control over high-impact actions, and more operational accountability around what autonomous systems are allowed to do.

CognaOne is not trying to bolt on compliance features after the fact. The product is built around the same runtime control principles that serious governance frameworks are moving toward: control before execution, bounded authority, traceable decisions, and meaningful human intervention.

GOVERNANCE THEME	WHAT ORGANISATIONS INCREASINGLY NEED	HOW COGNAONE FITS
Human oversight	Real ability to intervene, approve, interrupt, or stop consequential actions	Approval gates, escalation paths, interruptibility, and policy-bound decision points
Bounded autonomy	Agents should not operate with open-ended authority	Scope-bound execution tied to session, workflow, and policy context
Auditability	Clear records of what happened and why	Signed receipts and durable decision records at the moment of authorisation
Pre-execution control	Risk needs to be managed before side effects occur	Deterministic authorisation before execution
Operational accountability	Systems need reviewable control surfaces, not just raw logs	Structured decisions with reasons, outcomes, and escalation paths
Tool and third-party risk	External integrations need tighter runtime control	Action-level evaluation of parameters, scope, and policy before execution

The market gap

The AI infrastructure market has developed quickly around observation, orchestration, and memory. All three matter. None of them is authorisation.

LAYER	WHAT IT DOES	WHAT IT STILL LEAVES OPEN
Observability	Monitoring, dashboards, traces, logs	Explains what happened after the fact; does not decide whether action should proceed
Orchestration	Coordinates workflows, agents, and tool use	Routes action; does not authorise it

LAYER	WHAT IT DOES	WHAT IT STILL LEAVES OPEN
Memory and state	Stores context, retrieval outputs, and working information	Preserves state; does not verify integrity or govern action on it
Authorisation	Evaluates whether an action is allowed before execution	The missing runtime control layer: what CogniaOne is built to provide

The market does not need another log viewer with policy language wrapped around it. It needs a control layer that makes binding decisions before side effects happen. That is a harder technical problem, and a more durable one.

The direction is already visible in M&A. In February 2026, Proofpoint acquired AI security startup Acuvity specifically for its runtime enforcement capability across agents and MCP integrations. The strategic premium being assigned to enforcement-oriented architectures is real and growing.

The business case

For enterprise teams

Without authorisation, every new agent deployment expands the risk surface without creating a matching control surface. With authorisation in place, organisations can move faster without giving up operational discipline. Agents can act, but only within a runtime structure that preserves scope, control, and traceability.

In practice that means: faster internal approval for deployment because controls are visible and concrete; stronger oversight for sensitive workflows without blocking lower-risk automation; lower remediation cost because risky actions can be declined before impact; clearer evidence for operations, governance, audit, and incident review; no need to rebuild surrounding infrastructure.

For the market

Authorisation is not just another governance feature. It is a structural layer. The long-term value comes from correctness, not from cosmetic feature breadth. The challenge is not merely to observe what agents do; it is to create a stable runtime where agent reasoning can remain flexible while consequential action remains bounded, reviewable, and enforceable. That is what makes this category durable.

From observation to authorisation

The last generation of AI governance focused on visibility. Dashboards. Alerts. Traces. Audit logs. Those tools matter and will continue to matter.

But they do not answer the runtime question that becomes unavoidable once agents start taking meaningful action:

Should this action be allowed to happen?

Authorisation changes the posture completely. Every agent action is checked before execution against context, state, and policy. If it fits, it proceeds. If it does not, it stops, escalates, or waits for approval.

A system built this way does three things better: it acts on verified state rather than unchecked assumptions; it executes within bounded authority rather than open-ended permission; it produces durable decision records rather than trying to reconstruct truth afterward.

AI systems are crossing a threshold from generating outputs to taking actions. Once that happens, observation is necessary but insufficient. The control question moves to the centre: what is this system allowed to do, in this context, on this state, under this policy, right now?

That question needs a runtime answer.

CognaOne is the action authorisation layer for AI agents.