BUSINESS EXPLAINER

# Cognous Agent Control Plane

## Runtime control and replay for enterprise AI agents

> *Before an enterprise delegates meaningful work to AI agents, it must be able to reconstruct what those agents proposed, what they were permitted to do, what was blocked, and what they relied on.*

Business explainer prepared from the Cognous Agent Control Plane whitepaper (de Lima, 2026).

## 1. Executive Summary

Cognous Agent Control Plane is an open-source reference implementation for making AI-agent workflows inspectable, governable, and replayable. It is a runtime control layer that sits beside an AI agent — not inside it — and records what the agent proposed, what policy decided, what was blocked, what authority existed, what the agent relied on, and how the run can be reconstructed later.

It is deliberately not several things. It is not an agent framework. It is not a model. It is not a dashboard. It is not a full enterprise governance suite. It is the evidence and control layer that those other tools generally do not provide: a structured, replayable record of agent action, created at the moment actions are proposed rather than after the fact.

The central message for enterprise leaders is this:

> *As enterprises move from AI chatbots to AI agents that can use tools, access systems, and trigger workflows, governance must move from policy documents and after-the-fact logs to runtime control and replay.*

A chatbot that produces text can be governed by reviewing its output. An agent that reads customer records, drafts emails, proposes database updates, and triggers workflows cannot. For agents, the governance question is no longer “what did the model say?” but “what did the agent try to do, what was it permitted to do, what was stopped, and can we prove it?”

Agent Control Plane exists to make those questions answerable with records rather than reconstruction.

## 2. The Business Problem

AI agents are no longer just producing text. Across enterprises, they are beginning to:

- call tools and APIs;
- access enterprise data sources;
- summarize records and files;
- trigger workflows;
- draft and send messages;
- update systems of record;
- make recommendations that influence human decisions.

This shift creates a new class of business risk. With a chatbot, the output is the whole story. With an agent, the output is only the last step of a longer chain of proposed and executed actions. An enterprise may know the final result — an email was drafted, a summary was produced, a record was updated — without knowing what the agent proposed along the way, what was blocked, what it was allowed to do, what information it relied on, or how the decision path can be reconstructed if something goes wrong.

That gap shows up as six recurring pain points.

**1. Unrecorded agent actions.** Agents propose and take actions through tools. If those actions are not recorded as they are proposed, the enterprise has no reliable account of agent behavior — only whatever the application happened to log.

**2. Unauthorized or over-scoped tool use.** An agent granted access to a system for one purpose may use it more broadly. Without explicit, scoped authority records, it is difficult to say whether a given action was actually within the permission granted for that run.

**3. Missing records of blocked actions.** When a risky action is stopped, most systems record nothing. The absence of a bad outcome is not evidence that a control worked. Risk and audit teams need blocked behavior to be visible, not silently absent.

**4. Opaque source reliance.** Agent outputs depend on retrieved documents, database queries, tool responses, and user inputs. If that reliance is not recorded, a reviewer cannot distinguish a source-grounded answer from an unsupported one.

**5. Inconsistent policy decisions.** When permission logic is scattered across application code, the same action under the same conditions may be handled differently in different places. Inconsistency undermines both control and audit defensibility.

**6. Runs that cannot be reconstructed after an incident.** When something goes wrong, investigation often becomes forensic guesswork across scattered logs. Enterprises need a

portable, structured record of the run — what was proposed, decided, blocked, relied on, and produced.

These pain points are concrete. Consider four common deployments:

- A **customer-service agent** reviews an account and drafts (or attempts to send) a customer email.
- An **internal research agent** retrieves files and produces a summary that informs a decision.
- A **workflow agent** proposes an update to a production database.
- A **procurement or finance agent** proposes an action that should require explicit approval.

In each case, the business exposure is not primarily the model's text. It is the action layer: what the agent tried to do, under what authority, with what information, and with what record.

## 3. What Agent Control Plane Is

Agent Control Plane is a lightweight runtime layer that sits beside an AI agent. It does not replace the agent, the framework that orchestrates it, or the model that powers it. Instead, it records and governs the action layer around the agent.

In one sentence:

> *It gives enterprises a structured record of what the agent tried to do, what was allowed, what was blocked, why policy decided that way, what sources were relied on, and how the run can be replayed.*

The term "control plane" is worth translating into business language. Agent Control Plane is:

- **not the AI model itself** — it does not generate or reason;
- **not the workflow application** — it does not plan tasks or orchestrate tools;
- **not a dashboard after the fact** — it does not visualize; it creates the records a dashboard could later display;
- **a layer that records and governs agent actions as they are proposed** — before those actions are treated as executable.

The pattern is simple to state: before an agent's proposed action touches an external system, the proposal is recorded, evaluated against policy, and either allowed, blocked, or escalated — and every one of those events becomes part of a structured, exportable run record.

# 4. What It Does

## Records proposed actions before execution

Before an agent uses a tool or system, the proposed action is recorded: which tool, what type of action, what target.

**Business value:** the enterprise can distinguish what the agent wanted to do from what it actually did. That distinction is the foundation of meaningful agent oversight.

## Applies a deterministic policy gate

Each action proposal passes through a policy gate that evaluates it against allowed tools, blocked tools, the type of action, and the authority in scope for that run. Identical inputs produce identical decisions.

**Business value:** the same action under the same conditions receives the same decision, every time. This reduces the inconsistent, ad hoc permission handling that emerges when policy logic is scattered through application code.

## Creates blocked-action records

When the policy gate denies an action, the denial itself becomes a record — a first-class event in the run history, not a silent non-event.

**Business value:** blocked behavior becomes evidence. Risk teams can show not only that nothing bad happened, but that a specific risky action was proposed and stopped.

## Records authority context

The system records what authority existed for an actor or agent in that run — scoped by run, by actor, by permission type, and optionally by expiry.

**Business value:** reviewers can see whether a privileged action had the right authority at the time, rather than inferring it from configuration files or institutional memory.

## Records source reliance

The system records what the agent relied on during a run: tools, APIs, files, databases, user inputs, or model outputs.

**Business value:** this creates provenance for decisions and outputs. A reviewer can see what external information stood behind a summary, a recommendation, or an action.

## Produces policy evaluation traces

A policy decision can include an evaluation trace showing which rule matched, which rules were checked and did not match, and why the result was allow, block, or escalate.

**Business value:** audit and risk teams see not just the outcome but the decision path. “The email was blocked” becomes “the email was blocked because outbound-send authority was missing, after the tool itself passed the allowed-tool check.”

## Produces replay bundles

At the end of a run, the full record — context, proposals, decisions, traces, authority, reliance, blocked actions, and final output — can be packaged into a portable replay bundle for later reconstruction.

**Business value:** incident review becomes structured instead of speculative. The question “what happened in that run?” has an artifact as its answer.

## Supports validation reports

An exported run record or replay bundle can be checked for internal consistency: do decisions reference known actions, do blocked actions correspond to block decisions, do the pieces of the record actually fit together?

**Business value:** teams can detect broken records, missing references, and incomplete evidence packages before they are needed in a review.

## Supports redacted exports

Sensitive payloads, targets, final outputs, and decision reasons can be redacted while preserving the record’s structure, identifiers, and relationships.

**Business value:** records can be shared with auditors, legal, security, or customers without exposing unnecessary sensitive data — structural transparency without content exposure.

## Supports signed replay bundles

Replay bundles can carry export-integrity metadata, so a reviewer can detect whether the bundle was modified after it left the runtime environment.

**Business value:** records remain trustworthy once they are handed to another team, another system, or an external party.

## Provides a tool adapter pattern

The reference implementation includes a pattern in which a tool can be executed only after the control plane has recorded the proposed action and the policy gate has allowed it.

**Business value:** when the application integrates the pattern correctly, the agent cannot silently bypass the governance trail. Tool execution and record-keeping travel together.

# 5. Why It Is Valuable

**Better visibility.** Enterprises can see what agents proposed, what was allowed, what was blocked, what was relied on, and what was produced — across the whole run, not just at the output.

**Better auditability.** Audit teams receive structured, schema-governed records instead of scattered application logs that were never designed for governance review.

**Better incident reconstruction.** When something goes wrong, security and risk teams can reconstruct the run from a replay bundle rather than piecing together fragments after the fact.

**Better policy consistency.** Deterministic policy decisions reduce the variability that comes from ad hoc, code-embedded permission handling.

**Better evidence of blocked behavior.** Blocked actions become part of the record. Controls that worked leave proof that they worked.

**Better provenance.** Reliance records clarify what external information the agent used, supporting review of whether outputs were grounded in appropriate sources.

**Better controlled deployment.** The control plane gives platform and governance teams a concrete mechanism for moving from agent demos to governed deployment, with evidence generated from the first run onward.

**Better separation of responsibilities.** Application teams keep their agent framework and model choices. The control layer sits beside them. Governance requirements do not force a rebuild of the agent stack.

# 6. How It Is Different from Other Approaches

Enterprises evaluating agent governance encounter several overlapping tool categories. Agent Control Plane is complementary to most of them, and it is important to be precise about the differences.

## Versus agent frameworks

Agent frameworks help agents plan, use tools, and orchestrate multi-step workflows. They are essential for building agents — and they are focused on making agents act, not on giving enterprises a governance record of that action.

Agent Control Plane does not try to be the framework. It sits beside the framework and records and governs proposed actions, policy decisions, blocked actions, authority, reliance, and replay artifacts.

**Difference:** Frameworks help agents act. Agent Control Plane helps enterprises govern and reconstruct agent action.

## Versus observability and logging tools

Observability and logging tools record system behavior, typically as events emitted during or after execution. They are valuable for operations, but their records are usually unstructured for governance purposes and say little about intent, permission, or denial.

Agent Control Plane records action proposals, policy decisions, blocked operations, authority context, and reliance as first-class governance artifacts, created at the moment of decision.

**Difference:** Logging says what happened. Agent Control Plane records what was proposed, what was allowed or blocked, why, and how the run can be replayed.

## Versus dashboards

Dashboards visualize data. They are only as good as the records beneath them.

Agent Control Plane creates the structured records that a dashboard could later display. It has no user interface of its own by design.

**Difference:** A dashboard is a view. Agent Control Plane is a runtime record layer.

## Versus policy documents

Policy documents state what should happen: acceptable use, approval requirements, data-handling rules. They define intent, but they generate no runtime evidence.

Agent Control Plane records what policy actually decided during each run, action by action.

**Difference:** Policy documents define intent. Agent Control Plane records runtime evidence.

## Versus model safety filters

Model safety filters evaluate generated content — screening what a model says. That remains necessary, but it does not address the action layer: which tools were invoked, under what authority, against what targets.

Agent Control Plane focuses on agent actions, tool use, authority, reliance, and replay.

**Difference:** Safety filters govern generated content. Agent Control Plane governs the action trail around agent workflows.

## Versus full enterprise AI governance platforms

Enterprise governance platforms may cover model inventories, approval workflows, risk assessments, dashboards, and compliance reporting. That is program-level scope.

Agent Control Plane is intentionally narrower: an open-source reference implementation for runtime action control and replayable records. It could feed evidence into a broader platform, but it does not attempt to be one.

**Difference:** Governance platforms manage programs. Agent Control Plane captures the runtime evidence layer.

### Comparison at a glance

| Category | What it does | What it misses | How Agent Control Plane differs |
|---|---|---|---|
| **Agent frameworks** | Help agents plan, use tools, orchestrate workflows | Structured governance records of proposed and blocked actions | Sits beside the framework; records and gates action rather than orchestrating it |
| **Observability / logging** | Records system events, often after the fact | Intent, permission context, denial records, replayable structure | Records proposals, decisions, blocks, authority, and reliance as governance artifacts |
| **Dashboards** | Visualize data | The underlying structured records | Produces the runtime records a dashboard could display |
| **Policy documents** | Define what should happen | Runtime evidence that policy was applied | Records what policy decided in each run, with evaluation traces |
| **Model safety filters** | Screen generated content | Tool use, authority, action targets | Governs the action layer around the agent, not just the text |
| **Enterprise governance platforms** | Manage AI programs: inventories, approvals, reporting | Fine-grained runtime action evidence | Provides the narrow runtime control and replay layer such platforms can consume |

*Table 1. Agent Control Plane compared with adjacent tool categories.*

## 7. What It Is Not

Clear scope is part of the design. Cognous Agent Control Plane is **not**:

- a complete enterprise AI governance platform;
- an agent framework;
- a model runtime;
- a hosted dashboard;
- a production key-management system;
- a substitute for good policy design;

- a guarantee that model outputs are correct;
- a full compliance automation system;
- a complete security boundary by itself.

It enforces the policy rules and authority records it is given. If configured policies are incomplete or overly permissive, the control plane will faithfully record and apply them — it will not infer missing governance intent. Its value is focused: it creates structured runtime records and control points around AI-agent actions, and it should be deployed alongside application-level authorization, infrastructure security, model-risk controls, and operational monitoring.

## 8. Example Scenario

Consider a customer-service agent asked to review a customer account and draft a response.

1. **The task is framed.** The run begins with a recorded context: the task, the acting agent, the environment, the tools allowed and blocked, and the active policy version.
2. **The agent proposes reading a CRM record.** The proposal is recorded before anything is executed.
3. **The policy gate allows the CRM read.** The tool is on the allowed list and the action is a read. The decision and its evaluation trace are recorded.
4. **A reliance record captures that the CRM tool was used.** The run now documents what information source the agent depended on.
5. **The agent proposes sending an email.** Again, recorded before execution.
6. **The policy gate blocks the email.** Outbound-send authority was not granted for this run. The evaluation trace shows exactly which rule produced the block — a missing-authority block, not a banned tool.
7. **A blocked-action record is created.** The denied send is now a first-class event in the run history.
8. **The run completes** with a draft prepared for human review — and no email sent.
9. **A replay bundle is exported,** packaging the full run for later inspection.
10. **A validation report confirms** the exported record is internally consistent — every decision references a known action, every blocked action corresponds to a block decision.
11. **A redacted copy can be shared** with risk or compliance, preserving the structure and decisions while withholding sensitive customer content.

The business outcome: the enterprise can show, with structured evidence, what the agent proposed, what it was allowed to do, what it was blocked from doing, and what information it

relied on. If a question arises next quarter — from an auditor, a regulator, a customer, or an internal review — the answer is a record, not a reconstruction.

## 9. Why This Matters Now

Enterprises are moving from AI experimentation to AI deployment, and the risk profile changes at exactly that transition. A pilot agent that drafts text in a sandbox is a productivity experiment. A production agent that reads customer data, proposes system changes, and sends communications is an operational actor — and operational actors require operational controls.

> **As delegated authority increases, runtime evidence becomes essential.** *Every additional tool an agent can call, every system it can touch, and every workflow it can trigger expands the set of things an enterprise must be able to explain after the fact.*

A useful framing for leadership:

- **Chatbots require output review.** The text is the risk surface.
- **Agents require action governance.** The tool calls, data access, and system changes are the risk surface.
- **Enterprise agents require replayable records.** When agents act at scale, on behalf of the business, the ability to reconstruct any run becomes a baseline expectation — for security, for audit, for legal, and increasingly for customers and regulators.

Organizations that establish the evidence layer before scaling agent deployment will find governance a manageable engineering pattern. Organizations that scale first will find themselves retrofitting evidence onto systems that were never built to produce it.

## 10. Open-Source Reference Implementation

Agent Control Plane is published as a public, open-source reference implementation. The repository is public so teams can inspect the pattern, run the examples, adapt the schemas, test deterministic policy gating, validate exported records, redact sensitive fields, sign replay bundles, and experiment with controlled tool execution — before making any platform commitments.

In practical terms, the repository includes:

- a Python package with a clearly bounded public object model (built on Pydantic);
- published JSON schemas defining the record formats, so other systems can consume the records;
- a command-line interface for validating, redacting, signing, and verifying exported records;

- worked examples, including a mock framework-integration pattern;
- a test suite run through GitHub Actions across supported Python versions;
- an Apache-2.0 license.

The implementation is intentionally small. It is meant to be readable and verifiable by an enterprise platform team — a pattern to evaluate and build on, not a black box to trust.

## 11. Strategic Summary

Agent Control Plane gives enterprises a practical way to add runtime control and replay to AI-agent workflows without replacing their agent framework or model stack. It creates evidence where ordinary logs are insufficient: proposed actions, policy decisions, blocked operations, authority context, source reliance, validation reports, redacted exports, signed replay bundles, and replayable run records.

It does not claim to solve AI governance. It claims something narrower and more useful: to make agent runs inspectable, governable, and replayable at the layer where agents actually touch enterprise systems.

> *Before enterprises delegate meaningful work to AI agents, they need to know what those agents proposed, what they were allowed to do, what was blocked, what they relied on, and how the run can be reconstructed. That is the role of the Agent Control Plane.*