TECHNICAL ARCHITECTURE BRIEF

AI Agent Security Architecture

A proposed design for visibility and policy across MCP, APIs, connectors, browsers, code, workflows, and delegated agents.

PANTHEON CONTROL | CONCEPT DOCUMENT | OCTOBER 2026

Purpose and status

This brief describes a proposed architecture for Pantheon Control. It is intended to guide customer discovery and a bounded prototype. It is not a description of a deployed product, certified control, or live integration. The public demo is simulated and does not connect to agents, MCP servers, identity providers, endpoints, cloud accounts, or data stores.

Architecture principle: separate the control plane that reasons over evidence from the enforcement points that actually allow, deny, pause, or change an action.

1. Reference architecture

AGENT RUNTIMES	Custom assistants / API functions | MCP clients | SaaS connectors | Browser agents | Shell and code tools | Workflow orchestrators | Delegated agents
ACTION-PATH CONTROL POINTS	Agent host or runtime | MCP gateway or server | Application/API gateway | OAuth and SaaS scopes | Managed browser | Endpoint / container / CI | Workflow approvals
PANTHEON CONTROL - PROPOSED PLANE	Inventory and ownership | Identity and permission graph | Action event normalization | Deterministic policy decision | Incident correlation | Approval and response orchestration | Verification record
EXISTING CUSTOMER CONTROLS	Identity provider | EDR / endpoint | Cloud and CNAPP | API and network gateway | SaaS and data controls | Vulnerability and patch tools | Backup and recovery

Telemetry and identity context flow from the agent host, tool route, and existing systems into the proposed Pantheon evidence plane. For prevention, a decision must also return to an in-path control point before the destination executes the action. For post-event response, authorized actions go through existing customer controls and must be verified there.

2. Action-path coverage model

A single “agent integration” is not enough to describe coverage. For each route, the design needs to name the telemetry source, actual identity, point where a request can be intercepted, and supported response actions.

Path	What to inventory and observe	Candidate enforcement point	Known limitation
Remote MCP	Client, server, owner, transport, tool list, resource and calls.	Managed MCP gateway or server plus target authorization.	Unrouted calls bypass the gateway.
Local MCP stdio	Agent host, launched process, tool configuration, file/network reach.	Host policy, endpoint, process controls, or managed runtime.	A network sensor may not see standard-input/output messages.
API / function	Model tool request, app execution, service principal, API target, result.	Application logic, tool runtime, API gateway, destination.	A model request is not equivalent to an executed API operation.
SaaS connector	Connector owner, OAuth grants, user/service principal, records and actions.	Connector policy, identity scope, SaaS native controls.	A human grant may exceed the agent task.
Browser use	Agent browser, session identity, sites, actions, approvals.	Managed browser, session policy, application-side check.	UI activity may be missing from API/MCP logs.
Shell / code	Host, container, notebook, CI job, commands, files, secrets.	Endpoint, container runtime, CI permissions, secrets and IAM.	Child processes and environment secrets add reach.
Workflow / event	Trigger, queue, workflow graph, agent step, identity transitions.	Orchestrator approval, queue policy, destination controls.	Actions can be delayed and spread across services.
Delegated agent	Parent task, child agent, identity, tools, context and output.	Orchestrator, delegation policy, each child action boundary.	Lineage and effective permission can be lost in handoffs.

Enforcement labels to expose in the product

- **Pre-execution:** compatible enforcement point acknowledged the deny or approval hold before the downstream action ran.
- **Observed:** event arrived after execution or through read-only telemetry; no prevention claim.
- **Response-ready:** a suggested containment or remediation action is staged but has not been approved or executed.
- **Verified:** the source system was re-read and confirms the intended state.

Fail-safe behavior must be explicit. For a high-impact action, missing identity, stale policy, unavailable telemetry, or failed enforcement should not be presented as a successful block. The customer decides whether the source system should fail closed, require review, or follow a documented exception.

3. Data model and decision sequence

Normalize an agent action into a record that preserves both the requested action and the downstream operation. A minimum event shape should include:

```
event_id | timestamp | agent_id | runtime_id | human_owner | actual_principal | delegated_scope | path_type | server_or_tool | operation | target | data_class | environment | policy_id | decision | approval | enforcement_ack | downstream_result
```

Sequence	Design behavior	Evidence retained
Discover	Register agents, runtimes, owners, tools, connector grants, identity and reachable resources. Begin read-only.	Source, timestamp, owner, permissions, confidence and coverage gap.
Observe	Normalize events from agent hosts, gateways, APIs, identity, endpoint, cloud, SaaS and data systems.	Original event reference and normalized action record.
Correlate	Join action to principal, target resource, recent permission change, exposure and business context.	Explainable evidence graph with source attribution.
Evaluate	Apply customer rules to tool, action, scope, data, destination, environment and approval.	Policy version, inputs, decision reason and unresolved fields.
Enforce	Return allow, deny, or approval hold through the relevant action-path control.	Control identity, decision receipt, timing, and bypass state.
Respond	Stage bounded containment or remediation in existing customer controls.	Previewed scope, authorized approver, action receipt.
Verify	Re-read source systems and check the risky path and intended workflow.	Post-change state, residual risk and closure evidence.

Use deterministic conditions for consequential authorization. AI can help summarize an investigation, explain policy evidence, or draft a response plan, but an AI-generated risk score alone should not grant a tool permission or execute a destructive change.

4. Example action paths and control placements

Scenario	Where prevention could occur	After-action response and verification
Research assistant attempts to export restricted records from an MCP tool.	At a managed MCP proxy/server before tools/call executes, or at the data destination. Local stdio requires host controls.	Pause or disable the connector, revoke only the relevant token, recheck policy and query/export logs.
Finance agent calls a direct application function to issue a payment.	In the application function, API gateway, transaction service, or bank/payment approval control.	Freeze or review the transaction per customer authority; verify ledger and identity state.
Customer support agent updates account settings through a SaaS connector.	Connector scopes, OAuth policy, SaaS native approval, or API-side permission check.	Revoke excessive scope, review account changes, and verify the customer workflow still works.
Developer agent runs shell commands in CI and attempts cloud changes.	CI job permissions, endpoint/container policy, secret broker, cloud IAM and change approvals.	Cancel job, rotate exposed secret, reduce IAM, inspect cloud logs, re-run safe validation.
Browser agent attempts an unapproved production change.	Managed browser/session control or target application authorization and confirmation.	Revoke session if needed, inspect target state, restore authorized configuration.
Asynchronous workflow triggers multiple agent steps.	Workflow orchestrator, queue authorization, step-level checks, and target-side approval.	Pause the run, identify completed steps, compensate safely, verify every affected system.

5. JADEPUFFER coverage boundary

Sysdig reported an attacker-operated campaign that exploited an internet-facing vulnerable Langflow service (CVE-2025-3248), then used infrastructure access and credential discovery to reach a separate database environment. The research described destructive activity against Nacos configuration data and a later campaign targeting AI/ML assets. This incident did not start with an enterprise-owned agent making an unsafe tool call. [1][2]

A policy gate for MCP or other agent requests alone would not stop the initial Langflow exploit. A broader product concept could correlate asset exposure, vulnerability state, runtime events, secrets, identity, internal reachability, data access and recovery controls, then stage authorized response. This remains a hypothesis to validate through integration testing and customer discovery.

> **Accurate product language: “Pantheon is designed to help identify and coordinate response to risks across AI infrastructure and agent access paths, subject to connected telemetry and enforcement.” Do not say the prototype stopped JADEPUFFER.**

6. Prototype sequence and validation gates

Build stage	Scope	Exit evidence
1. Discover	One runtime and one chosen route; agent owner, identity, tools, permissions, targets.	A customer validates the inventory and unknowns.
2. Trace	Ingest events and produce a linked action record.	Attribution and source fidelity measured against test events.
3. Simulate	Run policy in shadow mode on a bounded set of actions.	False positives, exceptions, analyst agreement, latency recorded.
4. Enforce	One test-environment pre-action deny and one approval hold.	Enforcement receipt, bypass attempts, unavailable-control behavior, rollback.
5. Respond	One approved containment action through a customer control.	Action scope, approval, receipt, and post-action state verified.
6. Remediate	Fix the root cause and recheck relevant paths.	Patch, token, privilege, or workflow state re-read; approved task still passes.

Measurements

- Inventory coverage by runtime and action path, with confidence and unobserved routes visible.
- Share of consequential actions attributable to the actual principal and target.
- Policy decision agreement, false positives, exception rate, and decision latency.
- Pre-execution enforcement confirmation and bypass rate in controlled tests.
- Time to authorized containment and proportion of actions with verified source-system outcomes.

7. Security and governance requirements

- Least-privilege service identities; separate read-only discovery from response permissions.
- Human approval for high-impact changes, with explicit action preview and rollback plan.
- Tamper-evident audit records for evidence, policy version, approvals, enforcement, and verification.
- Secure secrets, isolate tenants, validate connector scopes, and monitor service accounts.
- Document source retention, telemetry gaps, local-tool behavior, and customer data boundaries.
- Map governance practices to the voluntary NIST AI RMF functions Govern, Map, Measure, and Manage. [3]

References

[1] Sysdig Threat Research, JADEPUFFER: Agentic ransomware for automated database extortion (July 1, 2026).

[2] Sysdig Threat Research, JADEPUFFER evolves: The agentic threat actor deploys ransomware built to destroy AI models (July 20, 2026).

[3] NIST, Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1 (January 2023).

[4] Model Context Protocol, What is MCP? (2026-07-28 specification documentation).

[5] OWASP GenAI Security Project, Practical Guide for Securely Using Third-Party MCP Servers 1.0 (2025).

[6] Anthropic, The 2026 State of AI Agents - Going Deeper.

[7] OpenAI, The State of Enterprise AI (2025).

Document status: proposed architecture, October 2026. No product integration, enforcement path, customer deployment, or JADEPUFFER prevention outcome is represented as implemented or validated.