P0 SECURITY

AuthZ Control Plane™ for Agents

Secure sensitive systems with runtime access control

How agents break traditional access management models

AI agents do not create a new access problem. They make an existing one much harder. Agents act autonomously, across more systems and at machine speed. They often rely on broad permissions, standing access and static credentials that were never designed for this kind of activity.

And an agent does not have to be compromised to cause harm. A well-functioning agent can misunderstand intent, choose the wrong course of action or use available permissions in ways the user never intended. That creates three core risks:

Multiple identities, no traceability

Every agent action involves at least the originator and the agent. Often, the agent inherits permissions and assumes the identity of their originator, making it difficult to figure out what was done, by whom.

Too much standing access for too long

Originators and agents often have broad permissions or static credentials that exceed the task and remain available long after they are needed.

Lack of enforcement, accountability breakdown

Agent workflows span multiple control points. If identity and policy enforcement are not carried out across the action chain, agents can easily take rogue actions and create unintentional failures, making audit impossible.

The three requirements for agentic access control

Session identity and provenance

Establish a unique session identity and preserve provenance and auditability across the entire action chain.

Runtime tool authorization

Determine which tools, capabilities or services the agent can use as the task is being executed, with policy deciding which actions can proceed autonomously and which require approval.

Task-specific access scope

Give the originator and agent only the permissions required for the current task and only as long as needed.

Why choose P0 Security for agentic runtime access control?

Discover

Know which agents exist and what they can reach.

- Inventory all agents, permissions and access paths across local and remote environments
- Identify shadow agents, standing access and emerging risk

Control

Decide what an agent can do, when and based on the privilege overlap between the originator and agent.

- Issue a blended identity that preserves originating users and acting agents for each session
- Enforce runtime policy based on originator, agent, action and resource context, with per-task JIT authorization in target systems to eliminate standing access

Prove

Understand what happened, who initiated it and why it was allowed.

- Capture the full action chain, provenance and attribution from originator through one or more agents, tools and target resources
- Monitor agent activity, access decisions and policy enforcement as actions occur, with audit history for governance, compliance, investigation and accountability

How P0 enforces policy end-to-end

What differentiates P0 is that runtime authorization does not stop at a single gateway or policy decision. P0 applies four core controls to each agentic request. Agent autonomy cannot depend on a human approving every action. P0 lets policy determine which actions can proceed autonomously and where human intervention is required.

Track the blended identity

P0 keeps originator and agent identities distinct and carries both through the session. Access decisions reflect the permissions of each rather than treating the agent as the user or a standalone service account. That means the same agent can be allowed to take different actions depending on who initiated the task.

Enforce policy at runtime

Before an action proceeds, P0 evaluates the originator, agent, requested action, target resource and relevant context. That context can be used to determine whether access is allowed, denied, requires approval or should be scoped more narrowly for the task.

Push least privilege into the target system

Where P0 integrates with the backend, enforcement extends beyond the agent or MCP gateway. P0 provisions task-scoped, just-in-time access through the target system's native IAM, reducing standing privilege and static credentials.

Preserve audit evidence

Each decision and action is recorded in Audit History with the identity context, policy decision, resource and outcome needed for governance, compliance and investigation.

Architecture at a glance

P0 Security runtime access control for agents

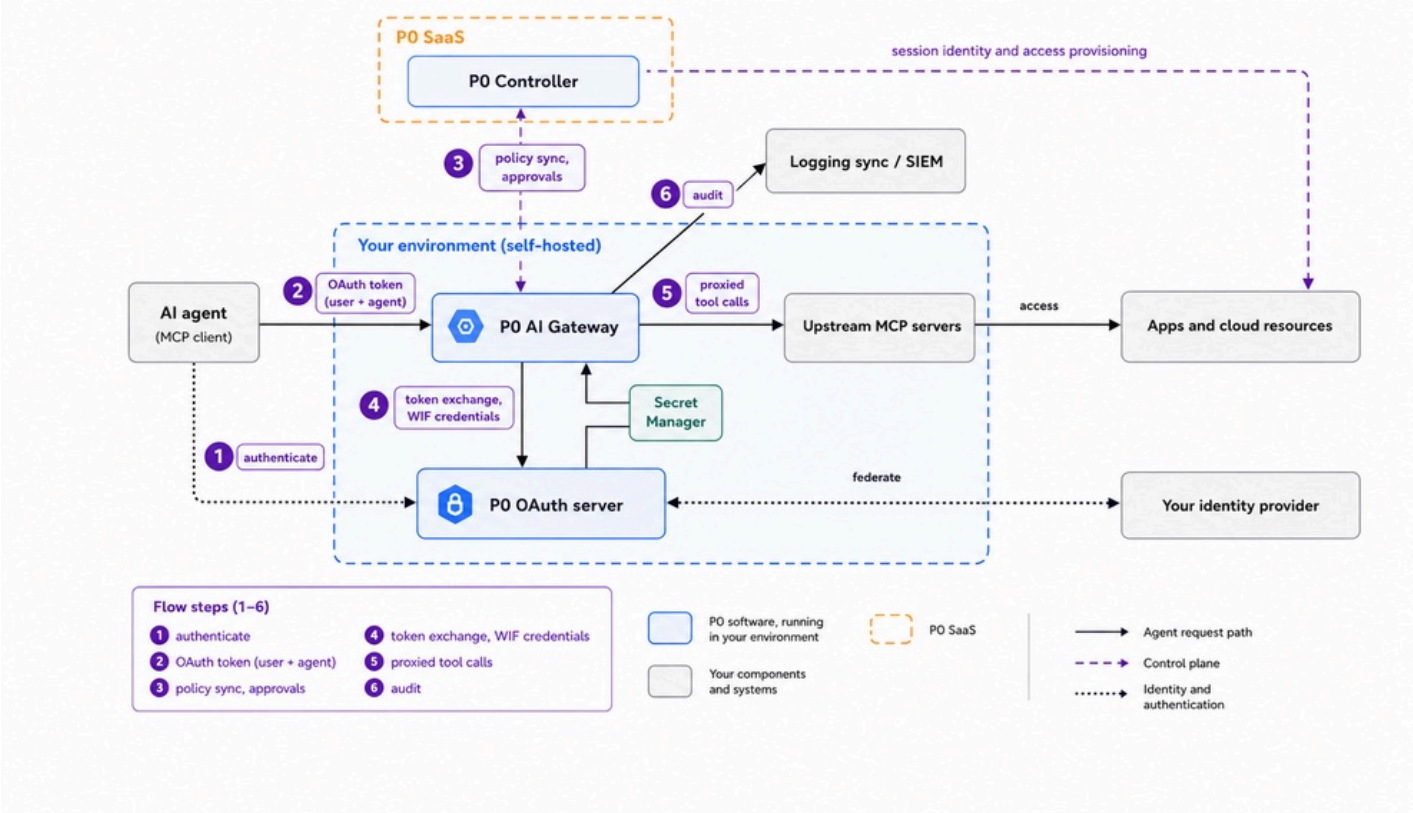
P0 separates centralized authorization policy from runtime enforcement. The OAuth Server and AI Gateway run in your environment, keeping enforcement close to the systems being protected, while policy and governance are managed centrally through the P0 AuthZ Control Plane.

P0 OAuth Server
Runs in your environment, federates with your existing IdP and issues session-scoped tokens that preserve originator and agent identity for authorization.

P0 AI Gateway
The P0 AI Gateway intercepts tool calls, evaluates policy before they proceed, then handles approvals and records activity:

- *Tool-level enforcement:* Controls which tools an originator and agent combination can invoke
- *Deep enforcement:* Provisions and enforces task-scoped access through native IAM where P0 integrates with the target system
- *Approvals:* Pauses sensitive actions and routes them for approval based on policy
- *Audit history:* Records the originator, agent, tool, outcome and policy decision for each action

P0 AuthZ Control Plane
Defines and manages the authorization policy that governs agent access, including roles, just-in-time (JIT) access, approvals, governance exceptions and Audit History.



How P0 fits into your environment

Components and deployment model

Component	Runs where	Role
AI agent	Customer environment	Initiates calls using OAuth tokens
P0 OAuth Server	Customer environment	Issues signed tokens binding originator and agent identity and federates with your IdP
P0 AI Gateway	Customer environment	Intercepts tool calls, evaluates policy, handles approvals and records outcomes
P0 AuthZ Control Plane	P0 SaaS	Defines authorization policy, roles, JIT access, governance and audit history
P0 Identity Inventory	P0 SaaS	Tracks agents, owners, permissions, access relationships and risk
MCP servers / target systems	Customer environment	Existing systems remain in place and are governed through P0 enforcement

Getting started

P0 works with your existing identity, agent and MCP infrastructure. Deployment connects those systems to P0 and defines how agent access should be authenticated, authorized and monitored.

1. Deploy the P0 OAuth Server and AI Gateway in your environment
2. Connect P0 to your existing IdP
3. Register agents and MCP servers
4. Define roles, permissions and approval policies
5. Route agent calls through the gateway and monitor activity in P0

To learn more, get the full deployment guide **here**.