FORTUNE 50 CPG COMPANY DEPLOYS RUNTIME AGENT ACCESS CONTROL ACROSS THREE PRODUCTION-READY USE CASES

Overview

The company had been running its own LLM gateway since 2024, connecting multiple AI providers across a large engineering organization. As teams started building AI agents, the platform team saw a problem the gateway didn't address.

Each agent type carried a different identity profile and a different access risk. Agents acting on a user's behalf hold delegated authority. The agent inherits the user's permissions which often contains sensitive or destructive entitlements that only a human should be trusted with. Shared agents use a single service identity no matter which user interacts with them. As with all impersonation, this often allows privilege escalation or data exfiltration. The shared agent must have wide enough permissions to serve multiple personas at the company, opening up access that the users do not hold themselves.

On the other end of the spectrum, headless agents operate with no user context. The agent runs without human supervision until a task is accomplished. Context drift, context poisoning, and the fundamental stochastic nature of AI poses the risk of privilege misuse. The agent may decide to take actions it was not intended to take, with no preventive human oversight. In the best case the harmful actions are detected after the fact, but more often than not they go unnoticed.

Multi-agent chains lose track of the delegation chain, whether user-initiated or driven by an autonomous agent. Scoping access to the last actor in the chain, that takes the action or accesses data, introduces in the same privilege escalation risk as a shared service identity. Even if the solution manages to carry the originating user's authority across handoffs between agents, the final agent shouldn't inherit the user's permissions due to the higher trust level and clear accountability that comes with human access. The platform team needed to show that a single authorization model could govern all three before committing to a production architecture.

> *'We knew agents needed runtime authorization. The question was whether one model could handle every mode we're building: user-initiated, headless, agent chains. That's what P0 proved was possible to do, and do well.'*
>
> **AI Platform Leader**,
> Fortune 50 CPG company

Solution

P0 Security set up a runtime authorization layer across three AI agent architectures in the organization's GCP environment, using its identity provider for inbound authentication and native GCP mechanisms for outbound access. All testing ran against live infrastructure.

1. **User-initiated agents: GCP service account expert**
When a developer interacts with the agent , P0 issues a short-lived credential that binds the requesting user's identity with the agent's. Two sessions have different access scoped to the intent of the session, bounded by a policy for the particular user and agent combination.. On-call engineers get write-capable IAM roles; standard developers get read-only access through the same agent. P0 ties every action to a specific person and a session-scoped grant.

2. **Headless agents: VM size optimizer**
A rightsizing agent runs on a recurring schedule with persistent read access to Compute instance metrics. When the agent flags a candidate to downsize, the write action requires human approval before it proceeds. P0 then provisions an ephemeral entitlement to the specific instance, and revokes it when the action completes. P0 never issues standing credentials for any operation.

3. **Agent-to-agent: Reporting chain**
A reporting agent calls a BigQuery agent to query a dataset; the results write to Cloud Storage, where a reading agent surfaces them to the requester. At each hop, P0 evaluates the originating identity and the final actor agent's identity in the chain before allowing the call. The full delegation context of every preceding hop travels through the chain.

P0 generated an audit record for each action across all three use cases, covering the originating identity, acting agent, tool called, policy applied and outcome. No log correlation was needed after the fact.

Why it matters

The organization ran an LLM gateway in production before the proof of value. The evaluation drew a sharper line between what the gateway handles cost-based provider routing, content guardrails) and what is handled by the P0 runtime access for agents: authentication, cryptographically verifiable provenance chain, and runtime access provisioning in GCP for that originator and actor, for that task, at that moment.

With authorization validated across all three agent patterns, the enterprise has a model it can apply to new agent workflows without granting broad permissions or creating gaps in audit history.

Results

3
Agent patterns validated under a single authorization model

0
Changes required to existing MCP servers or agent execution infrastructure

0
Standing credentials issued for write operations in the headless flow

100
Percent of agent actions attributed to a specific user-plus-agent pair at runtime