APP-NATIVE AI DISCOVERY AND GOVERNANCE

# Govern the AI agents built into your third-party apps

Salesforce Agentforce, Databricks, GitHub, Microsoft 365, Snowflake, ServiceNow, and Workday now ship agents of their own. Obsidian identifies them, maps what each one can reach, and flags risky configurations.

01

### Find the agents inside your apps

Including ones users built, vendors added, or that arrived switched on.

02

### Map what each agent can reach

The identities it runs as, and the tools and data those identities reach.

03

### Fix risky configurations

Agent findings sit with each app's other control failures, so you fix them the same way.

THE PROBLEM

## The apps you've already approved are shipping agents of their own

When security approves a third-party app, the review covers the app. An agent that appears inside it later often isn't reviewed the same way. Gartner predicts 40% of enterprise apps will include task-specific AI agents by the end of 2026, up from less than 5% in 2025 (Gartner, August 2025).

- ⚠ **They arrive without review.** AI features switch on, vendors add agents, and users build their own.
- ⚠ **They reach beyond the app.** An agent may call tools or reach other apps.
- ⚠ **They run on real identities.** A person's credentials, or a service account with broad access.
- ⚠ **Each app has its own console.** Seven apps means seven places to look, with no shared view.

WHAT SECURITY TEAMS NEED TO KNOW

# Four questions to answer before you can govern in-app agents

**Which agents exist**
Agents users built, agents the vendor added, and AI features that arrived switched on.

**Who's behind each one**
The identities the agent runs as.

**What each agent can reach**
The tools it calls and the data its identities can read or change.

**Which configurations carry risk**
Configurations that give an agent more access than its work needs.

| WHERE THE AGENT RUNS | WHAT IT SITS NEXT TO |
|---|---|
| **Salesforce Agentforce** | Customer records |
| **Databricks** | Business data |
| **GitHub** | Source code |
| **Microsoft 365** | Email, documents, files |
| **Snowflake** | Business data |
| **ServiceNow** | Operational workflows |
| **Workday** | People data |

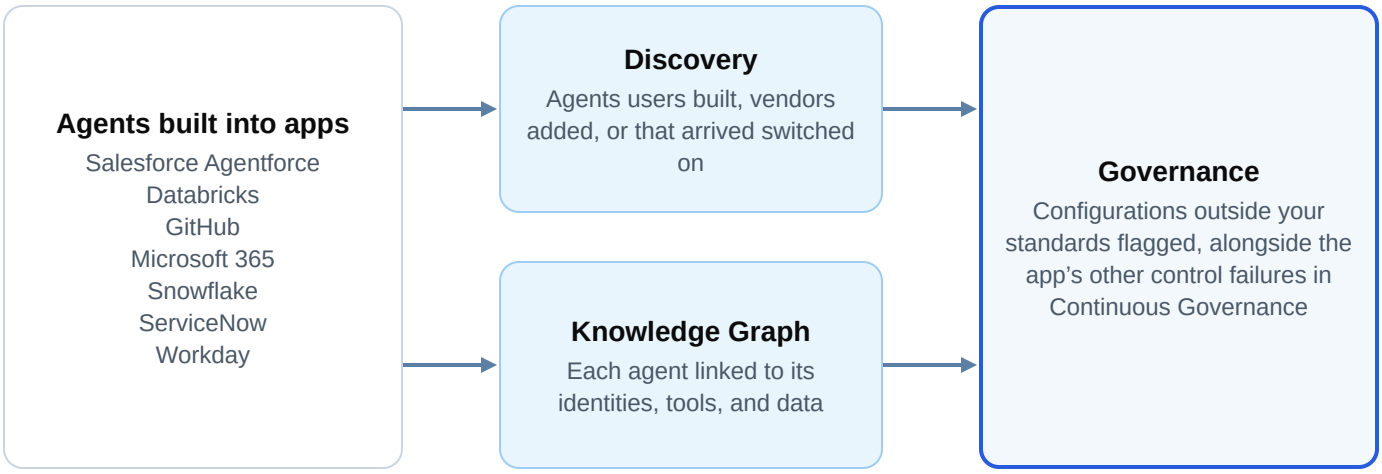
EACH APP'S ADMINISTRATION STAYS IN PLACE. HERE'S WHAT OBSIDIAN ADDS.

| CONTROL YOU RUN | WHAT IT ESTABLISHES | WHAT OBSIDIAN ADDS |
|---|---|---|
| **Each app's admin console** | Configures and administers the agents inside that app. | Agents from seven supported apps in one place, alongside agents on Codex, Claude Code, Cowork, Cursor, and other platforms. |
| **Roles and permissions in the app** | Decide what each identity can do in that app. | Each agent linked to the identities it uses and the tools and data those identities can reach. |
| **Security review at approval** | Covers the app's permissions, data, and integrations when it's approved. | Agent configurations outside your standards flagged as findings in Continuous Governance. |

- What Obsidian can see depends on what each app exposes, so depth varies by app.

HOW IT WORKS

# Obsidian finds in-app agents and maps their reach. It doesn't intercept their actions.




**Not intercepted:** the actions these agents take inside the app. Runtime decisions apply only where an action passes through a supported enforcement point.

**1.** Obsidian connects to supported apps and identifies the agents built into them.

**2.** The Knowledge Graph links each agent to its identities, tools, and data.

**3.** Agent configurations that fall outside your standards are flagged.

**4.** Findings sit with the app's other control failures, and your team fixes them the same way.

## TWO DIFFERENT CONTROLS

| | DISCOVERY AND GOVERNANCE | RUNTIME POLICY DECISION |
|---|---|---|
| **What it answers** | Which agents exist, what they can reach, and which configurations carry risk | Whether a specific action should proceed |
| **Where it applies** | Agents built into the seven apps above, and agents on supported agent platforms | Actions that pass through a supported enforcement point |
| **What security gets** | Agents identified, access mapped, and risky configurations flagged | Allow, block, or require approval where supported, with a reason |

GOVERNANCE COMES FIRST

# For an agent that acts entirely inside an app, governance is the control that applies

An app-native agent may never pass through an enforcement point. Finding it and narrowing its access keeps the risk to what its work requires, and keeps later runtime decisions accurate, since a policy can only account for the agents and access security knows about.

APPS WITH BUILT-IN AGENT COVERAGE

**Salesforce Agentforce** · **Databricks** · **GitHub** · **Microsoft 365** · **Snowflake** · **ServiceNow** · **Workday**

These agents appear in the same Obsidian inventory as agents on Codex, Claude Code and Cowork, Cursor, Microsoft Copilot, Amazon Bedrock, Google Vertex, and n8n.

EVALUATE

## Four questions to ask any vendor

**01** Which apps' built-in agents do you find today, including ones that arrived switched on?

**02** Can you show the identities each agent runs as and the data those identities reach?

**03** How do risky agent configurations reach the team that owns the app?

**04** For agents that act entirely inside an app, where does discovery end and enforcement begin?

A question a vendor can't answer clearly is a gap your team covers by hand.

## Find the agents inside your third-party apps

Request a demo to see which agents are built into your apps and what each one can reach.

**Request a Demo** obsidiansecurity.com/get-a-demo

Scan to book a demo