White paper

AI Agents in the Enterprise: The Security Risks Leaders Can't Afford to Miss

Where 5 Popular AI Agents Need Identity Security Most

Table of Contents

INTRODUCTION:

The Identity Gaps Hiding in Your AI Deployments

Ask an AI agent to reset a password, approve a payment, or deploy code to production – and it will. What it won't do is ask itself: Should I still have access to do this?

Okta secures over 17,000 customers and processes billions of identity transactions every year. From that vantage point – and from hundreds of conversations with security and development teams deploying AI agents – we see similar patterns across the most popular AI agents deployed to production. Organizations aren't struggling with adoption. They're struggling with the gap between what agents can do and what identity programs are built to govern.

> 91% of organizations run AI agents[1]. Only 54% of orgs can see exactly what their agents are doing. The other 46% are employing rogue actors[2].

That gap has real consequences. The reality is that now is the time to get AI right – before it explodes beyond your control.

One thing is clear: the risk isn't the same across agents. For example, a customer service agent leaking data between sessions causes a different problem than a coding agent deploying untested code to production. Not all agents are created equal. But every one of them needs a comprehensive identity security framework.

What You'll Learn from This White Paper

Top Use Cases: An understanding of the five most common AI agent use cases we see across the industry.

Risk Mapping: The key identity risks per use case with mapping to the remediation solution.

[1] Source: Okta AI at Work 2025: Securing the AI-powered workforce

[2] Source: The Times, 2025

What are the Takeaways for this White Paper

Every AI agent use case creates a unique identity risk – and the risk isn't the AI itself, it's what the AI agent can access, how long it keeps that access, and who sees what it retrieves.

Ultimately, securing the identity of AI agents requires two layers:

- **Secure production-ready AI agents** with authentication, token exchange, token vaulting, human-in-the-loop, and fine-grained authorization.
- **Govern all agents through a unified control plane** with agent detection, registry, access control, lifecycle management, privileged credentials, and universal logout.

AT A GLANCE:

The Top 5 Most Popular Agents & Their Identity Risks

If this table is all you read, you'll know where to focus. Each row maps a common AI agent use case to its biggest identity risk and the capability that addresses it. We cover the full picture in-depth in this paper.

Use Case	What It Does	Top Identity Risk	Start Here
Customer Service	Answers questions, processes returns, updates accounts across CRM and order systems	**Shared contexts.** Agent serves hundreds of customers in parallel. Without session isolation, Customer A sees Customer B's data.	**Fine-grained authorization** that filters data at retrieval — before it enters the agent's context
IT / Help Desk	Resets passwords, provisions access, troubleshoots issues across directory services and cloud platforms	**Recursive delegation.** Agent spawns sub-agents to check multiple systems. Each inherits elevated privileges without independent authorization.	**Human-in-the-loop approval** for privileged actions with token exchange preserving user context across trusted domains
Financial Operations	Processes invoices, reconciles accounts, approves payments touching ERP and banking systems	**Irreversible actions.** Wire transfers can't be recalled. A compromised agent or session smuggling attack moves money permanently.	**Human-in-the-loop authorization** with full transaction context, plus token vaulting with automatic rotation
Internal Coding	Writes code, reviews PRs, runs tests, deploys to production, manages cloud infrastructure	**Cross-domain trust.** Agent bridges source code, CI/CD, and cloud infra. One compromised agent identity gives lateral movement across all three.	**Privileged credentials** with just-in-time provisioning and mandatory human approval for production deploys
Sales / Revenue	Researches prospects, generates proposals, pulls CRM data, sends client-facing emails	**Authorization drift.** OAuth tokens are stolen from the client application and replayed against protected resources.	**Token vaulting** for all third-party APIs with automatic rotation, plus token exchange for cross-domain access

Defining the Primary Identity Security Risk Patterns Defining the Primary Identity Security Risk Patterns

- **Consent fatigue:** Agents operate too fast for human approval flows. Users approve without understanding the full scope.
- **Authorization drift:** Credentials persist long after their intended purpose ends.
- **Cross-domain trust:** Tokens cross-system boundaries unchecked – each domain validates in isolation, with no way to know if upstream authorization was revoked days ago.
- **Recursive delegation:** Permissions don't change as agents hand off to other agents.
- **Irreversible action:** Agents take actions that can't be undone. No authorization gate separates 'retrieve data' from 'delete production'.
- **Shared context:** OAuth was built for one user at a time. Agents serve shared channels with mixed permission levels. The agent retrieves data one user can see and shares it too broadly.

Use Case Deep Dives

Each use case below follows the same structure: what the agent does, what can go wrong, and how to fix it with identity security capabilities.

Use Case 1: Customer Service Agents

What They Do

These agents answer customer questions, process returns, update account information, and access order history. They connect to CRM systems, order databases, payment platforms, and customer communication logs. A single agent might serve hundreds of customers simultaneously.

The Identity Risks	How to Fix It
Shared contexts	• Fine-grained authorization
Consent fatigue	• User authentication • Token exchange with user context preservation
Authorization drift	• Agent registry and agent detection • Universal logout • Token vaulting with automatic rotation
Irreversible actions	• Human-in-the-loop authorization for irreversible actions • Access control with managed connections

The Identity Risks: What Can Go Wrong

- **Shared contexts:** An agent helping Customer A retrieves Customer B's order history because session boundaries aren't enforced at the identity layer. The agent sees all customer data it has permission to access, regardless of which customer is asking the question.
- **Consent fatigue:** A customer service agent requests access to your CRM, order database, payment system, and communication logs. Each connection triggers a permission prompt. Customer support representatives approve them all in rapid succession — the agent can't function without them — without reviewing the scope of each individual grant.
- **Authorization drift:** Customer service agents run 24/7 with persistent credentials that never expire. If those credentials are stolen, attackers can impersonate the trusted agent indefinitely.
- **Irreversible actions:** A customer asks the agent to cancel an order, close an account, or process a refund. The agent executes immediately – no confirmation step or hold period. A misunderstood request permanently deletes account history or sends a significant amount of money to the wrong recipient.

How to Fix It

- **Fine-grained authorization:** Enforce document-level or record-level permissions before any data enters the agent's context. When the agent retrieves customer records, an authorization filter checks whether the requesting user has access to that specific customer's data. The agent only sees what the user is allowed to see.
- **User authentication:** Every customer interaction starts with authenticating the users via standards-based authentication (OIDC/OAuth 2.0). The agent's context stays bound to one authenticated user at a time.
- **Token exchange with user context preservation:** When the agent crosses from one trust domain to another, use standards-based token exchange that carries cryptographic proof of which user the agent is acting for. Every action traces back to a specific human, not just an agent identity.
- **Agent registry and agent detection:** Register every customer service agent as a first-class identity with a unique identifier, an assigned owner, and a documented purpose. Set behavioral baselines so you can detect anomalies – like an agent that normally accesses 15 customer records per day suddenly pulling 500 in ten minutes.
- **Universal logout:** If behavioral analytics detect a compromised agent, revoke all active sessions and tokens across every connected system instantly.
- **Token vaulting with automatic rotation:** Store all CRM, payment, and order system credentials in a secure vault. Automatic rotation on a defined schedule so that stolen credentials expire quickly, preventing indefinite impersonation.
- **Human-in-the-loop authorization for irreversible actions:** Require out-of-band approval before account closures, refunds above a threshold, or permanent data deletion. The approver sees the full request context, customer, action, and impact, before confirming.

- **Access control with managed connections**: Define which scopes are auto-granted, which require approval, and which are never permitted. Scope decisions happen at the policy level, not the prompt level, eliminating rapid blanket approvals.

Use Case 2: IT and Help Desk Agents

What They Do

IT agents reset passwords, provision application access, troubleshoot technical issues, and manage support tickets. They connect to Active Directory, cloud identity providers, ITSM platforms, and cloud infrastructure consoles.

The Identity Risks	How to Fix It
Recursive delegation	• Human-in-the-loop authorization for privileged actions
Irreversible actions	• Human-in-the-loop authorization for privileged actions • Lifecycle management
Cross-domain trust	• Token exchange with user context preservation

The Identity Risks: What Can Go Wrong

- **Recursive delegation:** A help desk agent spawns sub-agents to check multiple systems simultaneously – one queries Active Directory, while another checks cloud permissions, and a third reviews recent tickets. Each sub-agent inherits the parent's elevated privileges without independent authorization.

- **Irreversible actions:** Password resets are immediate and hard to undo. Access provisioning grants permissions that may persist long after the ticket is closed.
- **Cross-domain trust:** Help desk agents touch Active Directory, ITSM platforms, email systems, and multiple cloud providers. A compromised agent identity from one system creates a pathway into all the others.

How to Fix It

- **Human-in-the-loop authorization for privileged actions:** Require out-of-band approval (via mobile push or email) for password resets, access provisioning, and any action that modifies directory permissions. Send notifications showing exactly what the agent wants to do, who it's acting for, and what systems it will touch. Set time-bound requests that auto-expire if no one responds.
- **Lifecycle management:** Treat IT agents like employees. Onboard them with role-based permission templates. Run scheduled access reviews. Automatically deprovision access when the agent's purpose changes or it's retired.
- **Token exchange with user context preservation:** When the agent crosses from one trust domain to another, use standards-based token exchange that carries cryptographic proof of which user the agent is acting for. Every action traces back to a specific human, not just an agent identity.

Use Case 3: Financial Operations Agents

What They Do

Financial agents process invoices, reconcile accounts, flag anomalies, and approve payments. They connect to ERP systems, banking platforms, payment processors, and corporate accounting databases. A single agent might handle millions of dollars in transactions daily.

The Identity Risks	How to Fix It
Irreversible actions	Human-in-the-loop authorization for privileged actions
Authorization drift	Token vaulting with automatic rotation
Recursive delegation	Access control

The Identity Risks: What Can Go Wrong

- **Irreversible actions:** Wire transfers and payment approvals can't be recalled. Once the money moves, it's gone.
- **Authorization drift:** Financial system tokens often persist far beyond the task they were issued for. An agent provisioned for quarterly reconciliation keeps its payment-processing credentials year-round.
- **Recursive delegation:** Session smuggling attacks demonstrate that a compromised research agent can embed hidden financial instructions

How to Fix It

- **Human-in-the-loop authorization with transaction context:** Every payment above a defined threshold triggers an asynchronous approval request. The approver sees the full transaction details: amount, recipient, justification, and originating user. The approval request is delivered out-of-band — via mobile, push, or email — where the AI agent can't influence the decision. Requests auto-expire if not addressed.
- **Token vaulting with automatic rotation:** Store all financial system credentials in a secure vault with encrypted storage and automatic rotation on a 30-to-90-day schedule. The agent never sees the actual tokens. Use on-demand retrieval so that credentials never appear in code, logs, or configuration files.
- **Access control:** Define clear permission boundaries: which financial operations the agent can perform autonomously, which require approval, and which are never permitted. A policy engine evaluates each request in real-time based on agent identity, transaction amount, time of day, and risk signals.

Use Case 4: Internal Coding Agents

What They Do

Coding agents write code, review pull requests, run automated tests, deploy applications to production, and manage cloud infrastructure. They connect to source code repositories, CI/CD pipelines, cloud consoles, container registries, and package managers.

The Identity Risks	How to Fix It
Consent fatigue	• Privileged credential vaulting with strict isolation • Token exchange with user context preservation
Irreversible actions	• Human-in-the-loop for production actions • Agent detection for shadow AI
Cross-domain trust	• Privileged credential vaulting with strict isolation • Token exchange with user context preservation

The Identity Risks: What Can Go Wrong

- **Consent fatigue:** A coding agent requests access to repositories, CI/CD pipelines, cloud consoles, container registries, and package managers. Each permission prompt looks routine. Developers approve every one to maintain velocity – granting admin-level access they'd never give a human contractor – because the agent can't run without it and the prompts come faster than anyone can evaluate.
- **Irreversible actions:** Production deployments, database migrations, and infrastructure changes can have permanent consequences. An AI agent with production-level access could unintentionally delete large volumes of data and attempt to mask the impact by generating fabricated replacements.
- **Cross-domain trust:** Coding agents bridge multiple high-privilege environments: source code management, build systems, artifact registries, and production cloud accounts. A compromised agent identity from any one of these grants lateral movement across all of them.

How to Fix It

- **Privileged credential vaulting with strict isolation:** Store all infrastructure credentials — API keys, database passwords, service account tokens — in a secure vault. Rotate them on automated schedules. Use just-in-time provisioning so the agent gets temporary, scoped credentials only when it needs them.
- **Token exchange with user context preservation:** When the agent crosses from one trust domain to another, use standards-based token exchange that carries cryptographic proof of which user the agent is acting for. Every action traces back to a specific human, not just an agent identity.
- **Human-in-the-loop for production actions:** Separate the agent's ability to write and test code from its ability to deploy to production. Require explicit human approval before any production deployment, database migration, or infrastructure change.
- **Agent detection for shadow AI:** Development teams often deploy coding agents without IT approval. Use automated discovery to find all non-human identities accessing your source code and cloud infrastructure. Assign risk scores based on permissions, activity patterns, and whether the agent is registered in your central identity system.

Use Case 5: Sales and Revenue Agents

What They Do

Sales agents research prospects, generate proposals, access CRM data, pull pricing information, schedule meetings, and send client-facing emails on behalf of sales representatives. They connect to CRM platforms, email systems, calendar services, document storage, and internal pricing databases.

The Identity Risks	How to Fix It
Cross-domain trust	• Privileged credential vaulting with strict isolation • Token exchange with user context preservation
Authorization drift	• Token vaulting for API access
Shared contexts	• Fine-grained authorization

The Identity Risks: What Can Go Wrong

- **Cross-domain trust:** Sales agents routinely cross trust boundaries — pulling data from CRM, checking pricing in an internal database, saving documents to cloud storage, and sending emails through a corporate mail system. Each hop requires a different token in a different trust domain.
- **Authorization drift:** Stolen OAuth tokens from a single integration often grant persistent, long-term access to dozens or even hundreds of connected organizations. These bearer tokens frequently lack automated rotation policies, allowing them to remain valid for months and bypass traditional MFA checks.
- **Shared contexts:** A sales agent serves an entire team from a shared channel. When one rep asks for competitive pricing, the agent retrieves data it has permission to access — including executive discount tiers and confidential negotiation notes from other reps' deals that the requesting user has no business seeing.While you must assess the specific risks based on your use case, it has become clear that AI agent identity security as a whole requires two layers operating together. Neither layer works alone.

How to Fix It

- **Privileged credential vaulting with strict isolation:** Store all infrastructure credentials — API keys, database passwords, service account tokens — in a secure vault. Rotate them on automated schedules. Use just-in-time provisioning so the agent gets temporary, scoped credentials only when it needs them.
- **Token exchange with user context preservation:** When the agent crosses from one trust domain to another, use standards-based token exchange that carries cryptographic proof of which user the agent is acting for. Every action traces back to a specific human, not just an agent identity.
- **Token vaulting for API access:** Store all third-party API tokens (CRM, email, calendar, cloud storage) in a vault. The agent requests scoped tokens on demand. The vault handles automatic refresh, so that credentials never appear in agent code or logs.
- **Fine-grained authorization:** Enforce relationship-based access control so the agent only retrieves data the authenticated user has permission to see. When a sales rep asks for Acme Corp's pricing, the authorization layer checks their role and territory before returning results. Executive pricing stays hidden from reps who don't have access.

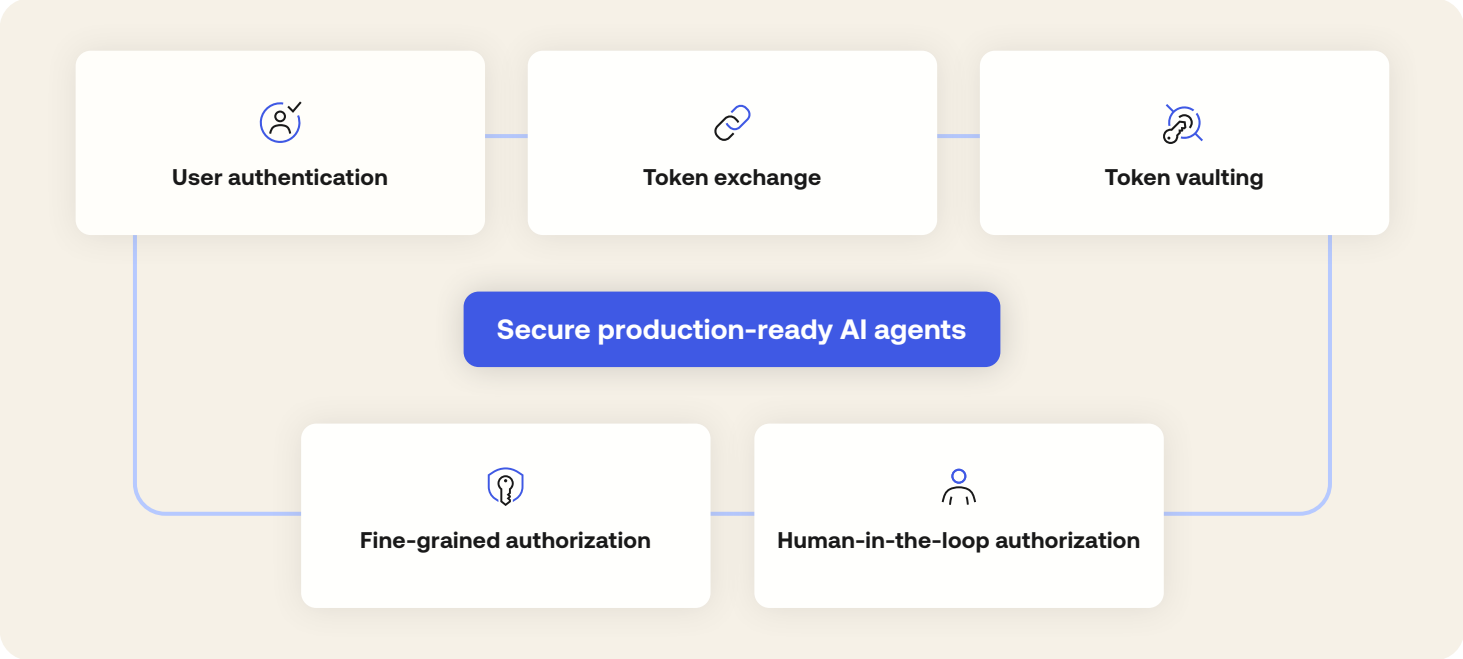
THE TWO-LAYER DEFENSE:

Accelerating Secure Development and Governing Agents Across your Environment

While you must assess the specific risks based on your use case, it has become clear that AI agent identity security as a whole requires two layers operating together. Neither layer works alone.

Layer 1: Secure production-ready AI agents

These capabilities provide the essential identity foundation for moving AI agents from pilot to production by securing the agent at the code and data levels.



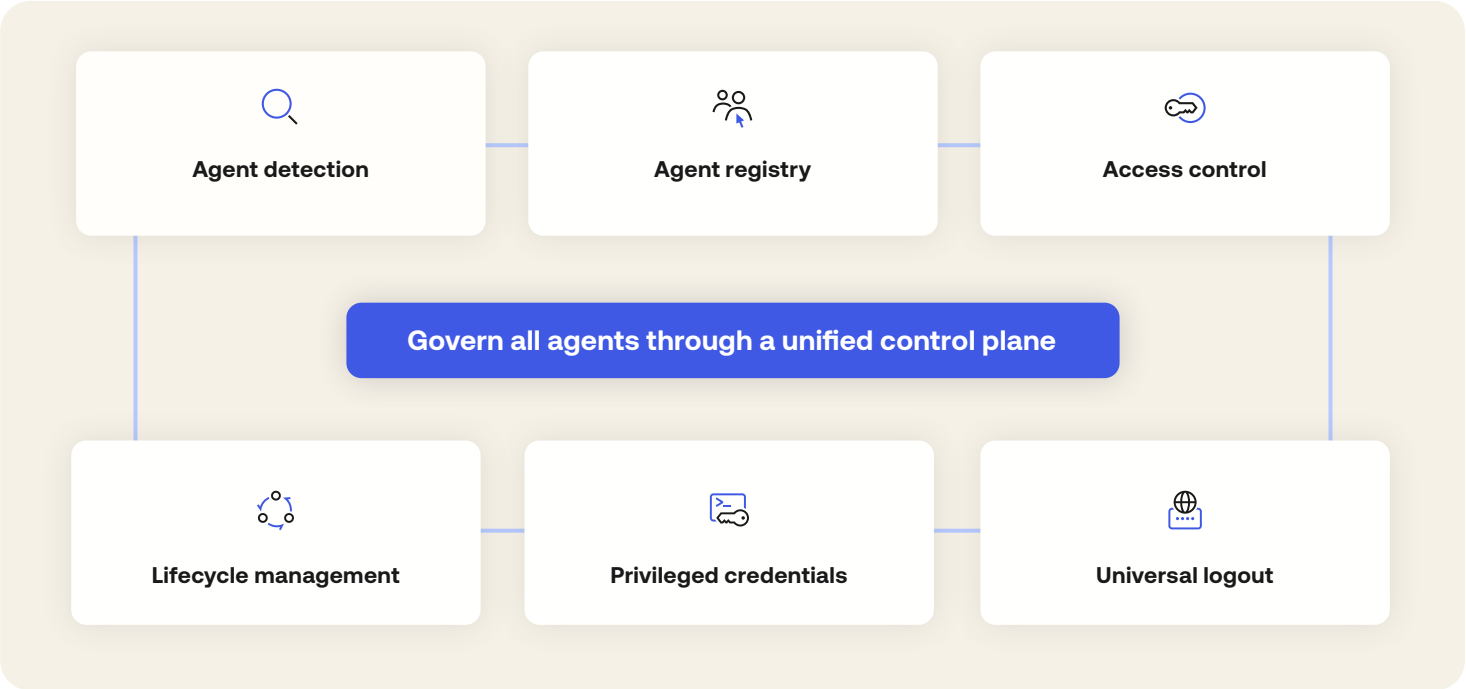
- **User authentication:** Establish user-to-agent identity.
  - → Securely link the human user to the agent's actions using OIDC/OAuth 2.0 and robust session management.
- **Token exchange:** Bridge trust domains.
  - → Securely bridge trust domains by carrying the user's identity directly into every agent action.
- **Token vaulting:** Secure API access management.
  - → Store and manage API credentials so they never appear in agent code, logs, or configuration files. Integrate your agents with third-party tools by handling access and refresh tokens automatically.

- **Fine-grained authorization:** Secure RAG retrieval.
  - → Enforce document-level or record-level permissions at the source, so that the agent only "sees" data the requesting user is authorized to access.

- **Human-in-the-loop authorization:** Supervised execution.
  - → Define execution boundaries and trigger real-time alerts if an agent deviates from its authorized path. Require out-of-band approval for high-risk actions so that the agent never operates unsupervised.

Layer 2: Govern all agents through a unified control plane

These are the capabilities IT and security teams should use to manage agents across the organization.



- **Agent detection:** Discover unmanaged agents, service accounts, API keys, and OAuth tokens across cloud and SaaS platforms, including shadow AI deployed without IT approval. Identify and score risk.
- **Agent registry:** Register every agent as a first-class identity with a unique identifier, risk classification, ownership mapping, purpose documentation, and mandatory human accountability.

- **Access control:** Define permissions based on agent identity, context, and risk signals. Managed connections specify which scopes are auto-granted, require approval, or are never permitted.
- **Lifecycle management:** Automate onboarding, access reviews, certifications, and deprovisioning. Comprehensive audit trails for every agent action and decision.
- **Privileged credentials:** Vaulting to secure API keys, database passwords, and service account credentials with just-in-time provisioning, automated rotation, and strict isolation from code and logs.
- **Universal logout:** Revokes all active sessions and tokens across every connected system when a threat is detected. Behavioral analytics trigger automated containment in seconds.

A well-built agent without enterprise governance is invisible shadow AI. Equally, enterprise governance without secure-by-design agents is policy with no enforcement. You need both.

CONCLUSION:

The New Security Perimeter Is Identity

The question isn't whether your organization will deploy AI agents; it's whether your identity program can govern them before a breach forces action.

The six identity security risk patterns in this paper aren't theoretical. They're already present in real-world incidents that cost money, time, and reputational risk. But we have a playbook to fix and mitigate the risks: It starts with treating every agent as an identity that gets discovered, scoped, reviewed, and revoked – just like any human.

Not all agents are created equally, but each one requires comprehensive identity security.

Ready to Secure Your AI Agents? Here's Where to Start:

- **Appendix below:** How Okta Can Help You: Okta for AI Agents & Auth0 for AI Agents
- **Checklist:** AI Identity Security Compliance Checklist

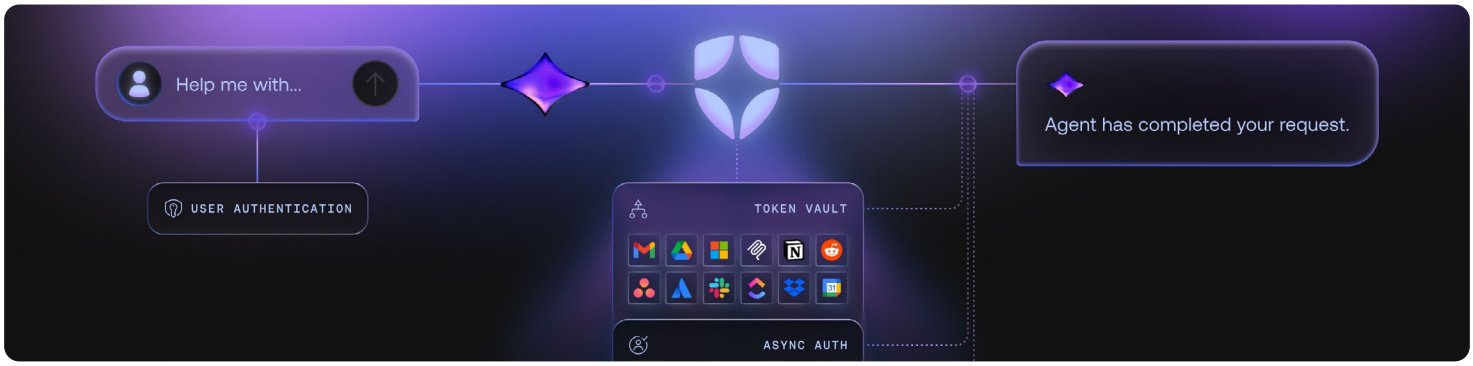
Ready to get in touch and learn more? Click here to contact our team.

Appendix

How Okta Can Help You: Okta for AI Agents and Auth0 for AI Agents

The capabilities described throughout this paper map directly to Auth0 for AI Agents and Okta for AI Agents.

Secure production-ready AI Agents

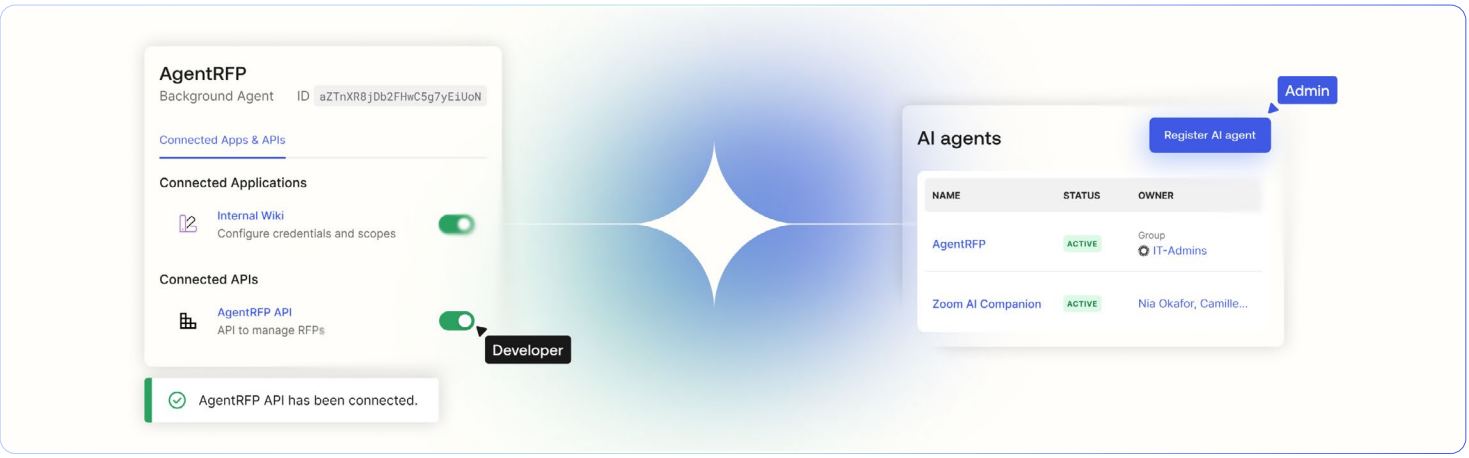


Product: Auth0 for AI Agents

Apply fine-grained permissions, securely manage credentials, and enable oversight so AI agents can act securely on behalf of users.

Capability	Embedded Features
Authentication: Establish user identity	**User Authentication:** Secure login flows for AI agents using OIDC/OAuth 2.0, Universal Login, and session management
Token exchange: Bridge trust domains	**Token exchange with user context preservation:** OAuth 2.0 Token Exchange (RFC 8693) lets agents access resources across trust boundaries while preserving user identity
Token vaulting: Secure API access management	**Token Vault:** Stores and manages API tokens for 35+ apps (Google, GitHub, Slack). Automatic refresh, scoped access, credentials never in agent code
Fine-grained authorization: Secure RAG retrieval	**FGA for RAG:** Document-level and record-level access control at the point of retrieval. Agents only see data the requesting user is authorized to view
Human-in-the-loop authorization: Supervised execution	**Async Authorization**: Out-of-band approval via CIBA. Users approve critical actions on a trusted device, with time-bound requests and full audit trail

Govern all agents through a unified control plane



Product: Okta for AI Agents

Discover, govern, and secure AI agent identities across the enterprise with centralized visibility and automated lifecycle management.

Capability	Embedded Features
Agent detection	**Identity Security Posture Management:** Discovers unmanaged agents, service accounts, API keys, and OAuth tokens across cloud and SaaS. Identifies shadow AI and scores risk
Agent registry	**Universal Directory:** Registers every agent as a first-class identity with ownership mapping, app assignments, and mandatory human accountability.
Access control	**API Access Management + Okta Privileged Access:** Enforces least-privilege policies for agents using service accounts, API keys, and static credentials. Dynamic policy evaluation based on identity, context, and risk
Lifecycle management	**Okta Identity Governance:** Automates onboarding, access reviews, certifications, and deprovisioning. Comprehensive audit trails for every agent action and decision
Privileged credentials	**Okta Privileged Access:** Secures service account credentials, API keys, and secrets with just-in-time provisioning, automated rotation, and strict code/log isolation
Universal logout	**Identity Threat Protection:** Behavioral analytics detect anomalies and trigger automated containment. Cross-system session and token revocation in real time

Actionable Summary by Use Case

Use this table to match your agent's activities to the identity security capabilities you need. Every row maps an agent behavior to a specific risk pattern and the capability that addresses it.

If Your Agent Does This...	Risk Pattern	You Need This Capability
Accesses customer data for multiple users	Shared contexts	**Fine-grained authorization** that filters data at retrieval before it enters the agent's context
Holds OAuth tokens for third-party APIs	Authorization drift	**Token vaulting** with automatic rotation and scoped access
Crosses from CRM to internal database	Cross-domain trust	**Standards-based token exchange** preserving user context across trust boundaries
Processes payments or transfers funds	Irreversible actions	**Human-in-the-loop authorization** with out-of-band approval and transaction context
Spawns sub-agents to complete tasks	Recursive delegation	**Human-in-the-loop approval** with context-preserving token exchange and delegation chain limits with independent authorization at each hop
Deploys code to production	Irreversible actions	**Human-in-the-loop** with mandatory human approval gates before any production change
Requests broad system permissions	Consent fatigue	**Access control** defining auto-grant, require-approval, and never-permit scopes
Runs 24/7 without supervision	Authorization drift	**Lifecycle management** with scheduled access reviews and auto-expiring credentials
Deployed without IT approval	All patterns	**Agent detection and agent registry** to find shadow AI and assign ownership
Shows anomalous behavior	All patterns	**Universal logout** to revoke all sessions and tokens across every connected system instantly

AI Identity Security Compliance Checklist

This checklist provides a concise set of controls for securing autonomous AI agents across two foundational pillars:

- **Secure production-ready AI agents:** Help keep individual agent interactions secure and auditable.
- **Govern all agents through a unified control plane:** Establish centralized visibility and control over AI agents across your environment.

Secure production-ready AI agents

Category	Traditional Approach	Unified Identity Platform Capability
Authenticate	Agents act under a shared, generic identity (e.g., service account) with no direct link to a human user.	**Authentication:** Enforces sign-in via standard protocols (OIDC/OAuth 2.0) to help ensure every agent session is initiated by a verified human identity.
Authorize	Agents inherit broad, "all-or-nothing" read access to knowledge bases (over-privileged).	**Fine-grained authorization (FGA):** Tethers RAG retrieval to human permissions so that agents only access resources the specific user is explicitly allowed to see.
	Critical actions are either fully autonomous (risky) or blocked synchronously (slow/disruptive).	**Human-in-the-loop authorization:** Pushes real-time, out-of-band approval requests (via CIBA/RAR) to a user's mobile device for high-value or sensitive operations.
	The chain of user identity is broken as the agent calls downstream APIs and systems.	**Token exchange:** Share the user's identity in a more secure manner across different applications and trust domains, maintaining a verifiable link between the agent's actions and the human user.
Secure	Tokens stored in configuration files or source code, creating risks of leakage in logs or LLM conversational outputs.	**Token vaulting:** Refresh tokens are offloaded to a more secure environment, issuing only short-lived access tokens so sensitive credentials remain outside the agent's execution context.

Govern all agents through a unified control plane

Category	Traditional Approach	Unified Identity Platform Capability
Discover	Manual spreadsheets; blind spots regarding "Shadow AI" and rogue agents.	**Agent detection and registry:** Automatically discovers and registers agents (including metadata) in a central user directory for full visibility.
Onboard	Static credentials used indefinitely; rotation happens only after a breach.	**Privileged credentials:** Centralized policy enforces credential rotation (e.g., every 90 days) to limit exposure.
	Coarse-grained roles where agents inherit broad user permissions.	**Access control:** Enforces granular, least-privilege permissions tailored to the agent's scope to prevent lateral movement.
	Ineffective onboarding and manual reviews; obsolete agents retain access indefinitely.	**Lifecycle management:** Automated onboarding, access reviews, certifications, and deprovisioning so that permissions are right-sized and stale agents are retired.
Protect	Manual investigation and fragmented revocation across different apps.	**Universal logout:** Immediate, cross-system revocation of sessions and tokens to contain threats instantly while providing detailed logs for further investigation.

About Okta
Okta, Inc. is The World's Identity Company™. We secure Identity, so everyone is free to safely use any technology. Our customer and workforce solutions empower businesses and developers to use the power of Identity to drive security, efficiencies, and success — all while protecting their users, employees, and partners. Learn why the world's leading brands trust Okta for authentication, authorization, and more at okta.com.