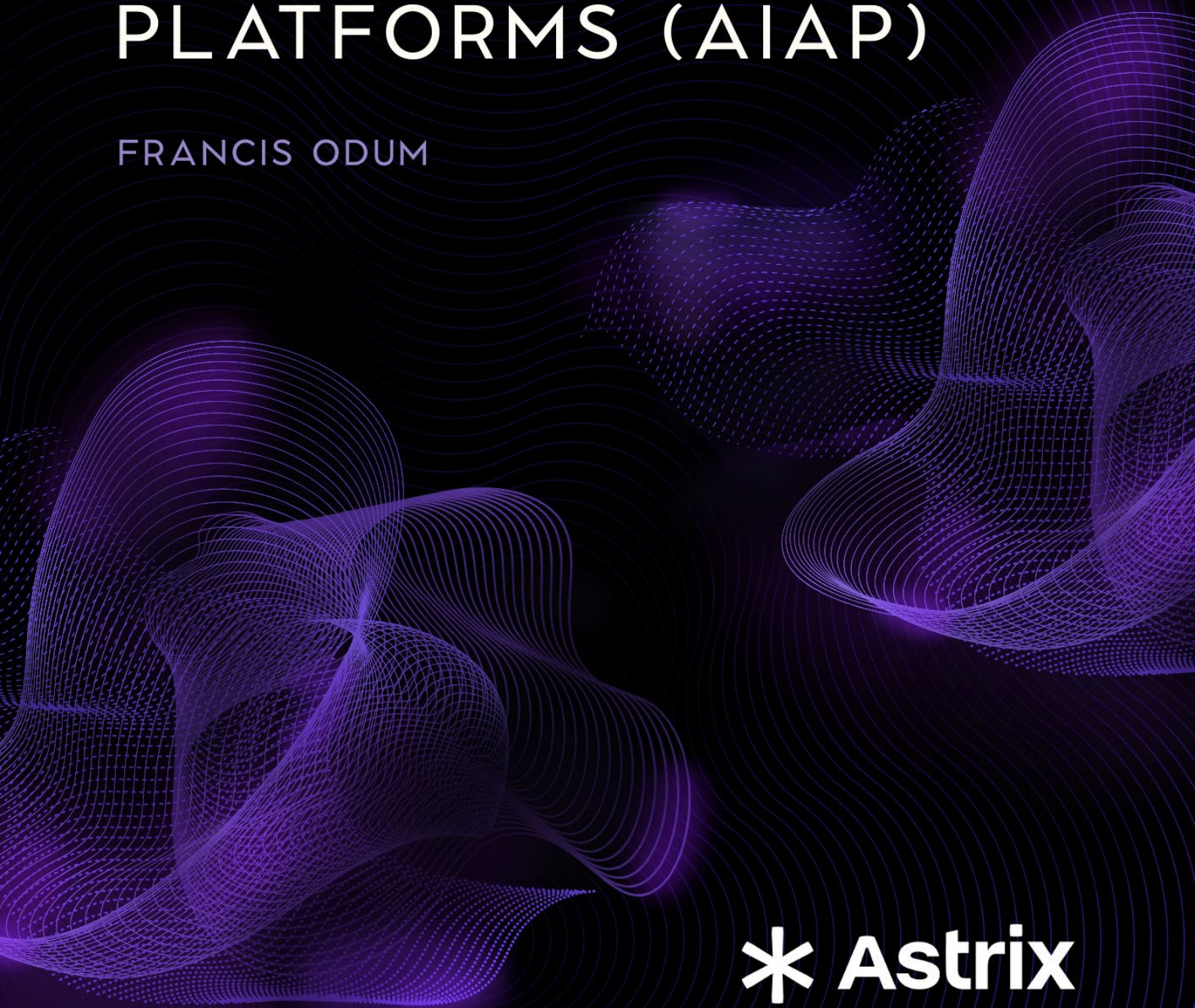
SACR®

EMERGING AGENTIC IDENTITY ACCESS PLATFORMS (AIAP)

FRANCIS ODUM



Astrix

Research

We explore the newest frontiers of cybersecurity.

Whether you're looking at emerging vendors, evolving threats, or shifting architectures, our timely, opinionated insights help modern security leaders make smarter, faster decisions.

About Software Analyst Cybersecurity Research

SACR is a modern research and advisory firm built for today's cybersecurity leaders. We deliver in-depth, timely analysis across SOC operations, Identity, Network, Cloud, Application Security, Data, and AI Security; equipping CISOs, security teams, founders, investors, and practitioners with the insight they need to navigate high-stakes decisions.

With an engaged community of over **80,000** readers and followers, SACR connects with a global network of cybersecurity decision-makers and innovators. Our access to leaders across categories and industries gives us a direct line to the conversations shaping the market. By pairing these insights with rigorous technical analysis and continuous market tracking, we produce research that is both data-driven and grounded in the realities of modern security operations.

Whether you're seeking clarity on emerging technologies, evaluating vendors, or tracking market shifts, SACR delivers trusted, independent research designed to help you see clearly and decide with confidence.

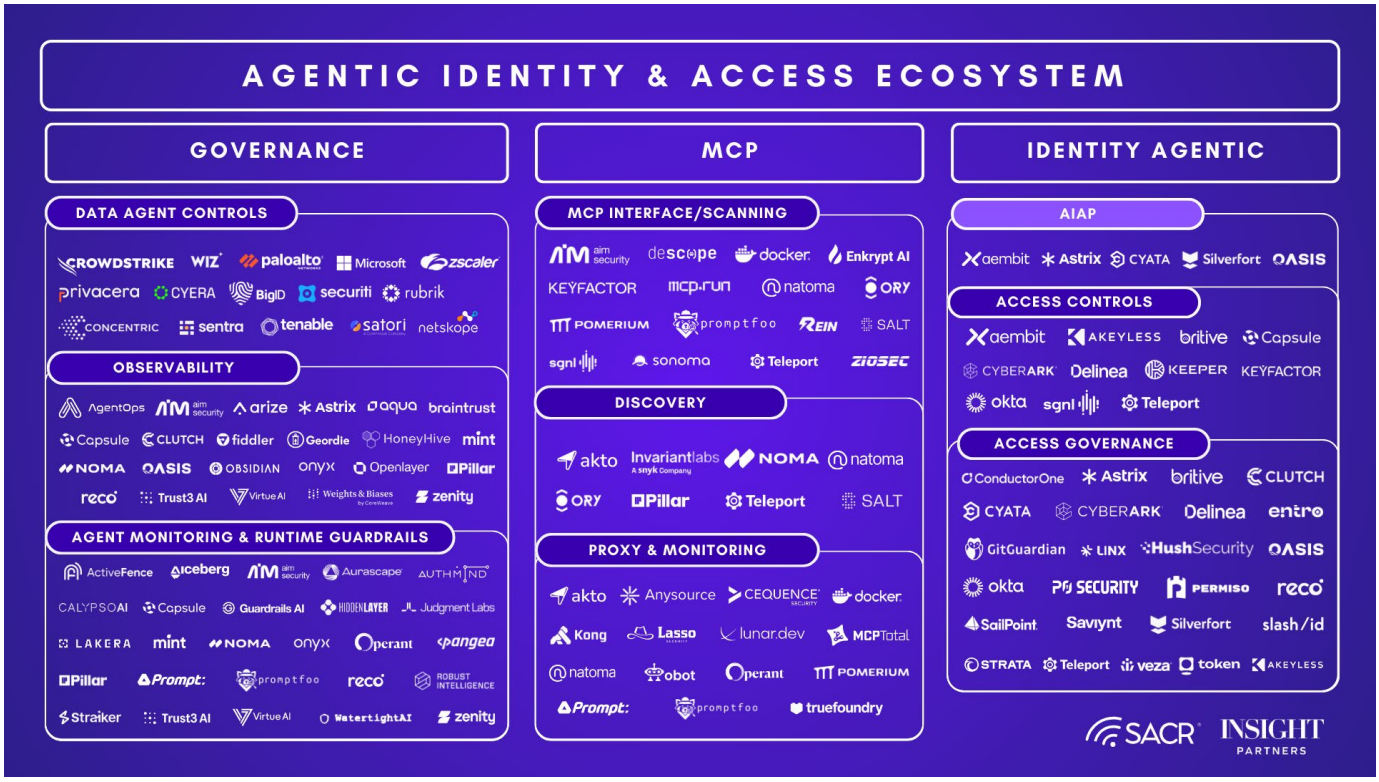
Author

- **Francis Odum** is the Founder/CEO of the Software Analyst Cyber Research, where he leads the firm's research and engagement with cybersecurity leaders.

Table of Contents

Executive Summary

SACR is betting that **Identity security will change significantly in 2026,** shifting from how IAM / Security leaders have historically thought about it. The new identity architecture is undergoing a fundamental change. This is a comprehensive report on the evolving dynamics in our newly released agentic identity and access ecosystem map!

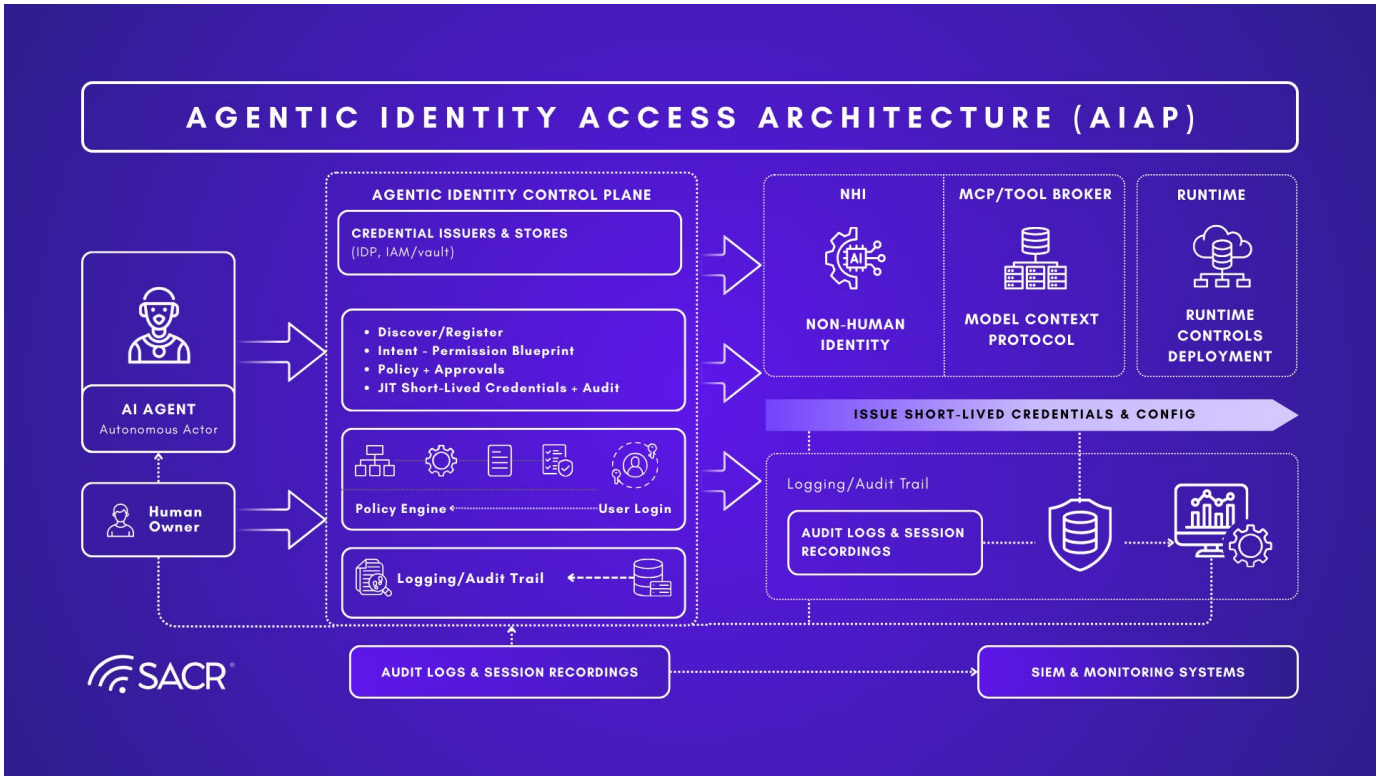


Full SACR MM. Excerpts from Insight Partners.

Traditional IAM, IGA, and PAM were designed for human login patterns. They struggle to govern agents and non-human identities. Enterprises are moving from a world where identity governance assumes slow human actions to one where agents execute high-frequency autonomous actions at machine speed. The net result is a widening gap where enterprises are increasingly deploying agents that operate like privileged insiders, while the underlying identity architecture remains fragmented, static, and overly credential-centric. This creates fragile identity chains (Agent between its NHI or Enterprise target system) in which the agent's autonomy is only as secure as the non-human identity (NHI) and behaviours it holds.

To close this gap, **our new report** introduces a concept focused on end-to-end architecture in which an **Agentic Identity Access Platform (AIAP)** functions as a "new SSO for Agents." Rather than allowing agents to connect directly to SaaS APIs and infrastructure endpoints with embedded secrets, AIAPs introduce a centralized broker that standardizes how agents request access, translates intent into deterministic authorization decisions, and ensures access is ephemeral, scoped, and continuously enforceable.

This report frames this as the emergence of an **"Okta + Sailpoint for Agents or firewalls for the internet."** However, this is a fundamentally different architecture with a centralized broker that shifts governance from *who* a human is to *why* an agent or user is acting, and issues task-scoped identities and permissions and entitlements only when an authorized action is being requested (considering the prompt intent) or in progress during an agent or tool execution phase.



SACR believes that a modern and ideal Agentic Identity Access Platform (AIAP) converges on four operational model phases for practitioners, including:

- **Phase 1: Discover, Inventory & Agent Registration**
- **Phase 2: Translate & Authorize (Intent Policy Layer)**
- **Phase 3: Broker & Inject (Access Fulfillment Layer)**
- **Phase 4: Watch & Terminate (Runtime Threat Layer)**

All of this occurs while implementing agent-specific primitives: EDR-driven discovery, owner attestation, intent-aware authorization, and zero-standing-privileges (ZSP). The near-term market is fragmented but converging. We're observing that vendors are differentiating on:

1. **Visibility depth and breadth:** Going beyond mere breadth of discovery (across endpoints, SaaS, and cloud environments) to provide deep, comprehensive visibility.
2. **Enforcement and context:** Moving past basic visibility and activity logs ("what") to offer runtime enforcement and, critically, the ability to capture the underlying context, intent, and attribution ("why").
3. **Strong user awareness and experience:** This focuses on gaining a broader context of how the user and the controls they are using at a specific point in time.

Structure For The Report

Practical guidance

The remainder of the report grounds this architecture in real-world implementation patterns through case studies and representative vendors. Each vendor is analyzed based on where it fits across the four phases: visibility and inventory, intent-to-policy translation, credential brokerage and secretless injection, and runtime enforcement and response, highlighting both strengths and gaps. The goal is not to declare a single "winner," but to provide security and IAM leaders with a practical lens for evaluating how the market is converging on a unified access control plane for autonomous systems, and how to assemble an end-to-end program that is deployable today while aligned with the inevitable future: centralized brokering, agent-to-agent governance, and identity as a temporary, continuously enforced state.

Vendor and case studies

Disclosure and Research Methodology

The second half of the report identifies five representative vendors that serve as examples and have published strong research supporting their status as leading players in this category. SACR collaborated with five vendors referenced in this report by reviewing product materials, participating in briefings and demonstrations, and validating technical claims where possible. This collaboration improved the factual accuracy of the analysis and ensured the report reflects current product capabilities. SACR maintained full editorial control over the report's structure, evaluation criteria, and conclusions. All opinions, assessments, and comparative judgments are those of SACR and were developed independently to remain objective and practitioner-focused

- Vendor Case Studies: How the Market Implements AIAP
- Where Vendors Differ: Visibility vs Policy vs Brokerage vs Runtime
- Reference Architecture: Recommended Deployment Patterns

Key Insights From This Report in 5 Minutes

1. **Agents are changing the traditional identity architecture:** Enterprises are moving from governance models built for slow, human-initiated actions to environments where agents execute high-frequency autonomous actions using long-lived secrets that were never designed for non-deterministic actors. The practical consequence is a fragile identity chain: Agent to NHI / MCP to Enterprise system, where the agent's autonomy is only as safe as the non-human identities (NHIs) and tool paths it can reach (see the early "identity chain" framing and the "Emerging Shift" sections).

2. **Legacy IAM/SSO solved "who," but agents force you to secure "why" (intent) and "how long" (ephemeral access).**
   Okta/Microsoft Entra-era architecture centralized authentication and improved policy at login time, but it assumes bounded human intent and manageable identity volume. Agents break those assumptions: they run continuously, spawn, chain tools, and operate across multiple activation surfaces, with no single IdP choke point. The result: classic SSO can't answer the questions that matter now: intent, permission blueprint, credential lineage, and end-to-end auditability.

3. **The winning architecture (The AIAP/AAM) model** converges on four operational phases, which are structured across the whole report. It details the discovery process, including runtime controls. This doesn't "secure the model first"; it rebuilds access around intent and ephemeral credentials. This report's core thesis is that the emerging control plane is essentially "SSO for Agents", but architecturally different: a centralized broker that shifts governance from who a human is to why an agent (or user+agent) is acting, then issues task-scoped identities/ permissions only while an approved action is in progress.

4. **Zero Standing Privileges is the execution model that makes everything else real:** ZSP isn't a slogan here, but it's the mechanical advantage: if access is always short-lived and task-scoped, then runtime enforcement becomes decisive (the "kill switch" is simply refusing to renew or revoking an ephemeral session). That's how you prevent credential replay and shrink the blast radius when agents or endpoints get compromised

5. **Three forward shifts to watch (this is where the market is going):**

- NEW-AAIP coincides with the **rise of the centralized identity broker ("SSO for Agents").** Agents no longer connect directly to SaaS/cloud APIs with embedded credentials.
- **Agent-to-Agent protocols become a first-class governance problem.** You'll need workflow-level delegation rules and verification between agents.
- **A unified access layer collapses silos** among NHI, workload identity, and agentic identity into a single dynamic access layer, where identity is a temporary state granted by validated intent and context.

TLDR: This is not "just another AI security tool category." It's an identity control-plane re-platforming. Finally, we believe this is a new model across the industry that re-architects how vendors, practitioners, and identity security teams secure their infrastructure in an agile era.

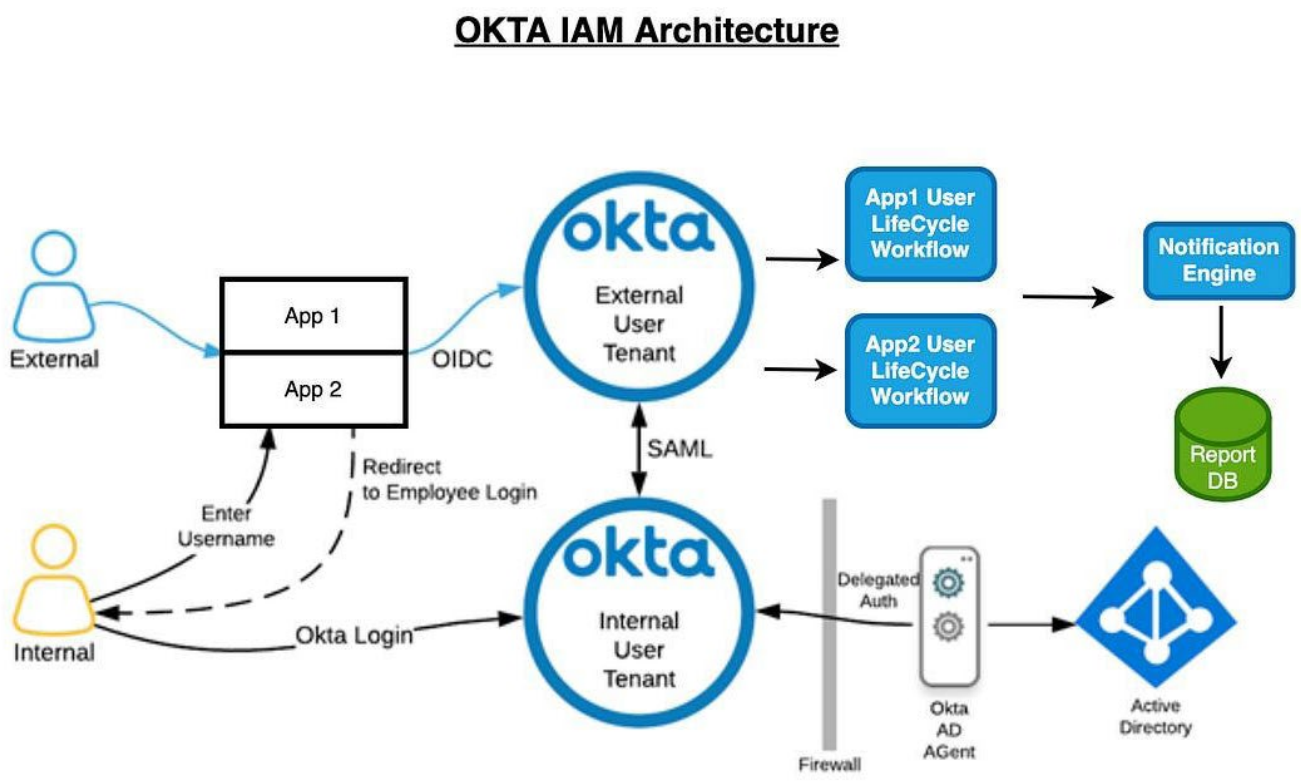
The Old Architecture (How Okta / Microsoft Worked)

Today's foundational identity architecture built around Microsoft and Okta solved for the fragmentation in human authentication. Before centralized IdPs and identity federation standards, each application implemented its own authentication, stored its own credentials, and enforced access independently, making user management difficult and less trustworthy. Users accumulated many credentials without centralized password management, access was commonly over-provisioned and user access was rarely revoked on time.

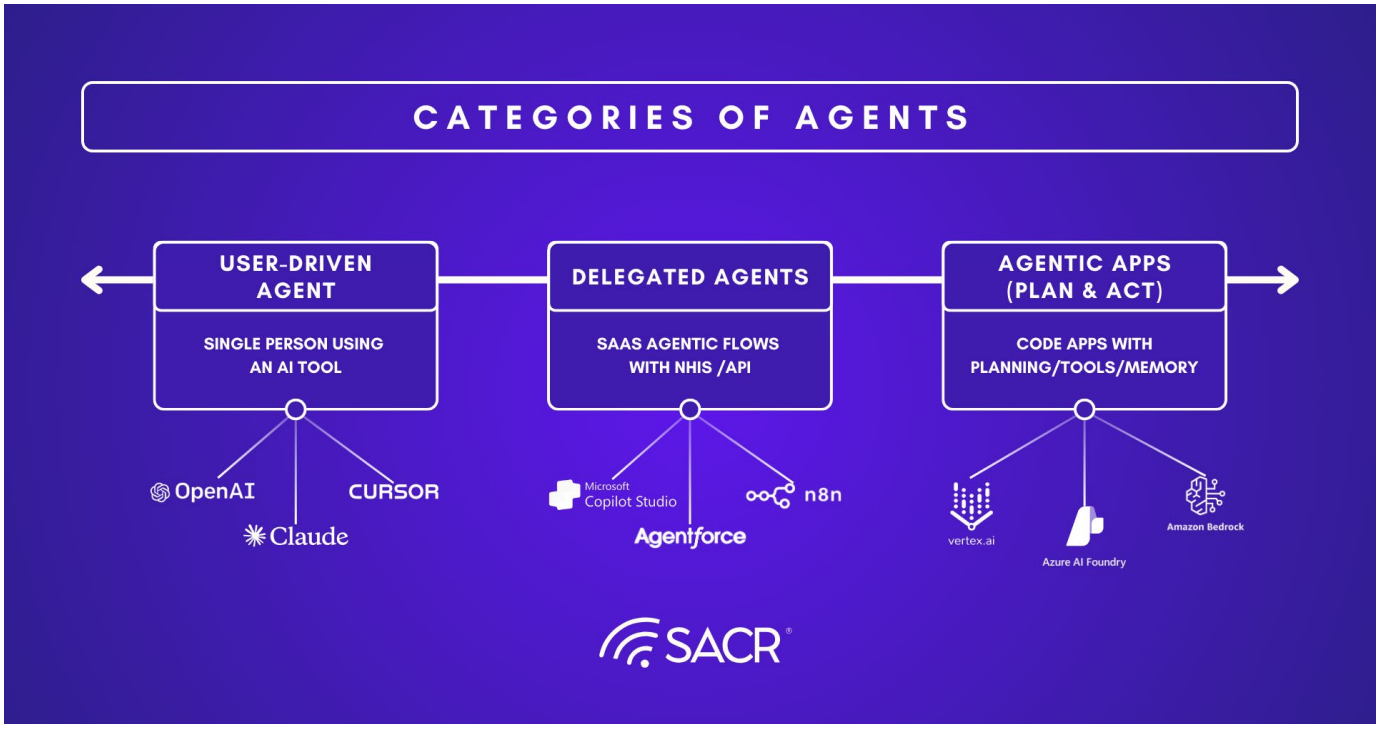
The breakthrough was the human identity control plane where we could authenticate once (SSO), continuously verify risk (MFA/conditional access), and broker access through centralized policy rather than scattered credentials. This decoupled identity from applications and made access a real-time policy decision rather than a one-time login event. However, this architecture assumes:

- The actor is a human with bounded intent.
- Authentication is the primary gate.
- The number of identities is manageable.
- The "who did this?" The question can be answered by linking human identity.

The legacy identity architecture built around **SAML, OAuth 2.0, OpenID Connect (OIDC)** (for authentication and Single Sign-On), and **SCIM** (for provisioning) and vendors Okta SSO or Microsoft Entra were primarily designed for a world where a human (principal) initiates access via a well-defined login/session boundary, and where authorization can be expressed largely as standing entitlements (groups, app assignments, long-lived OAuth grants) enforced at the point of authentication. Authorization was subsequently built around legacy companies like Sailpoint and large incumbents.

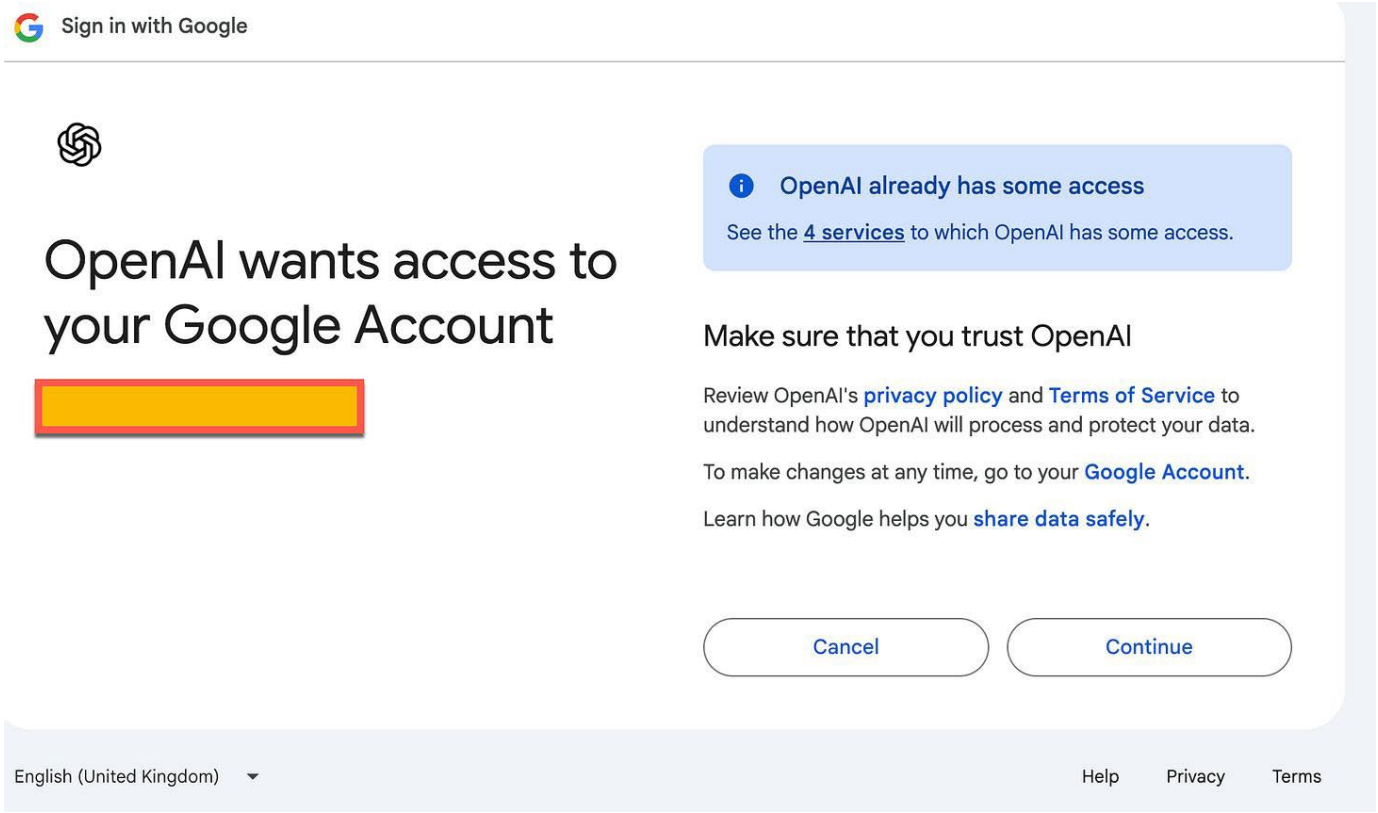


Categorizing Agents

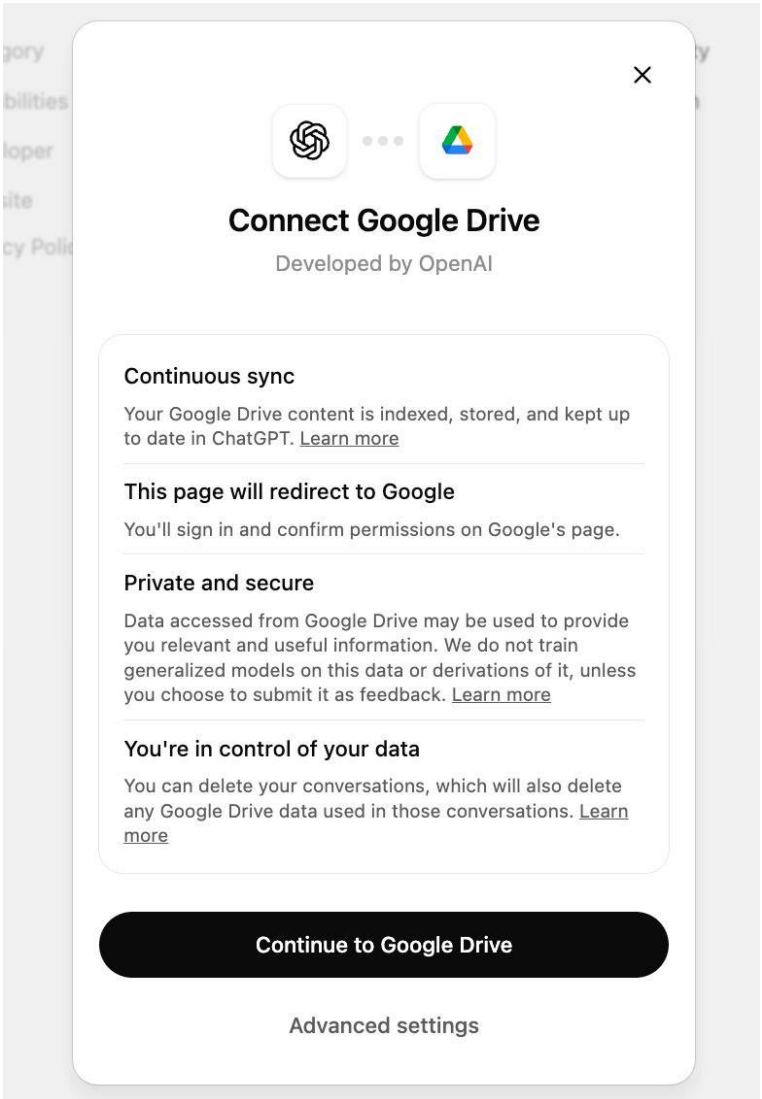


There is a big shift starting in 2026, and we believe that this will manifest itself in 2027. In this new world, the actor behind an action is no longer a human identity, but is now a fully capable agent. For context, we are defining agents into two main categories. We like how Aembit categories the spectrum of Agents:

1. **User-driven Agents (on-behalf-of): These categories of agents** execute tasks that a human explicitly initiates and runs in that user's context. Many agents in today's enterprise are primarily user-driven. The upside is productivity and speed of spinning them up. The downside is identity sprawl with two classic failure modes: **rights inflation** (the agent effectively inherits a user's broad entitlements in an always-on way) and **attribution failure** (you can see the agent acted, but can't reliably tie the action back to *which* human authorized it). Aembit's answer is to bind **Agent Identity + User Context** into a single authorization decision, so access is evaluated as **"Agent X acting for User Y"**, and constrained to what *both* the agent and that specific user are allowed to do. In practice, that means the agent can be prevented from exercising privileged actions even if the user has admin rights elsewhere, and the audit trail preserves who initiated the action.



2. **Autonomous Agents (workload-driven):** This is more rare, but increasingly growing with examples like OpenClaw / Moltbolts. These agents behave more like long-running services (server-side workflows, background processes, or tool-using agents, often speaking protocols like MCP) that operate without a human in the loop. Here the dominant risk is not delegation, it's **credential hygiene**. These agents frequently depend on **static and long-lived secrets** (API keys, tokens) embedded in code or config, which become catastrophic if the agent or environment is compromised. Aembit's model is to treat the agent as a **first-class workload identity** and front it with an **identity gateway** (e.g., an MCP identity gateway) that performs **token exchange** and issues **short-lived, just-in-time credentials**. The agent authenticates to the gateway, but never directly holds the "forever credential," shrinking the blast radius and making least-privilege enforcement tractable.



This new reality changes the security equation because agentic identities differ fundamentally from human identities:

- They are dynamic and evolving (vs. static and predictable),
- They rely on tokens and keys (vs. MFA-mediated human sessions),
- They operate at high volume, machine speed (vs. low volume, human speed),
- And their ideal form is short-lived and task-based (vs. long-standing).

In these two classic agentic worlds discussed above, they break those assumptions. An "agent" is a continuously operating actor that translates natural language into tool calls, chains actions across many systems, and often runs across **multiple activation surfaces** (endpoint, browser, SaaS agent platforms, MCP/tool servers) where there is no single IdP choke point. In the second assumption above, an AI agent can run continuously, spawn variants, and request expansive permissions. Further, these operate via embedded secrets and service accounts.

As a result, classic SSO cannot reliably answer the key questions security teams now need:

- What was the agent trying to do (intent),
- What exact capabilities were required (permission blueprint),
- What credentials were minted and where were they injected (delegation chain), and
- What happened end-to-end (auditability).

Instead, enterprises need intent-to-access translation, deterministic policy gating (e.g., OPA), just-in-time short-lived scoped credentials, and a controlled execution boundary where secrets never touch the agent capabilities that sit outside the core design center of traditional IdPs and their session-centric access model.

Additionally, this shift is not gradual, we believe it will be volume-driven. Human identity workforce growth was linear. However, in this new architecture where every human identity has multiple agents, agentic identity populations will scale exponentially. Astrix has a report where they had near-zero to over 15,000 entities in months. This is a machine-speed identity problem.

Challenges Leaders Face in This New Identity Architecture

There are a number of risks and challenges that block secure adoption for many leaders.

Legacy IAM & PAM cannot secure agents comprehensively

Legacy IAM and PAM systems weren't architected for the high-velocity and non-deterministic nature of agents who are capable of chaining actions across systems. As agent adoption accelerates across enterprises, this gap will only grow bigger. Agents expose a structural mismatch in legacy IAM.

The stack was built for deterministic actors (humans) and some infrastructure exists for fairly predictable non-human identities. These agents can now plan, branch, and chain actions across multiple services at machine speed. The crucial point is that because an agent's next step is inherently non-deterministic, traditional IAM and PAM can't express the kind of *runtime-bounded, intent-scoped delegation* agents actually need, nor can they reliably preserve attribution as actions propagate through sub-agents, MCP servers, and downstream APIs.

Further, agents leverage tools like NHIs and multi-hop where in another situation, a single human prompt can trigger sub-agents, MCP servers, and downstream services where identity context degrades. This "chain of custody" becomes difficult to reconstruct. When something destructive happens (e.g., data deletion, privilege changes), accountability blurs across the user, the agent runtime, and the tool/service that executed the action. This leaves enterprises in an "all-or-nothing" access control dilemma as we saw in the early days of ChatGPT. Organizations either (1) let agents inherit full user privileges (maximizing utility but

creating unacceptable blast radius and muddy accountability when something destructive occurs), (2) mint dedicated standing credentials such as long-lived API keys or service accounts (still risky, hard to rotate, and operationally unscalable), or (3) deny access entirely (eliminating value). In other words, agents don't just add another identity type, they break the underlying assumptions of delegation and auditability that modern IAM depends on, forcing enterprises into unsafe privilege models precisely when they need tighter control.

Accountability, Chain-of-custody and Auditability Challenges

A single prompt can trigger cascades across sub-agents, MCP servers, and downstream services, causing identity context to decay. Security teams can't confidently attribute actions, reconstruct intent, or assign responsibility after destructive outcomes. Agent autonomy vs least privilege tension are becoming acute. Many enterprises will have the challenge around the best ways to manifest agents in their enterprise. The reason is that agents work best when they can explore, plan, and chain tasks. This pushes teams to grant broad, anticipatory permissions so the agent can "discover" what it needs at runtime (exactly the opposite of least privilege). The predictable outcome is systemic overpermissioning, growing blast radius, and weak control over what data/tools the agent can touch as tasks evolve. There is also an accountability gap in that where audit trails cannot clearly attribute actions inside agentic chains (e.g., an agent deletes a production database and who is accountable?).

MCP expands risk through new trust boundaries and common implementation pitfalls

Agent adoption is increasing an identity sprawl for enterprises. Most practitioners find that to make agents useful, teams must over-provision access and end up minting or reusing a growing pile of NHIs (API keys, OAuth grants, service accounts, tokens, and connector credentials) often without clear ownership, rotation, or segregation-of-duties controls.

MCP accelerates this because it standardizes how agents plug into tools, but in many real deployments it also normalizes risky secret-handling patterns (credentials stored locally, copied across MCP servers, embedded in configs, or reused across environments), increasing the probability of leakage and unauthorized reuse.

Many research papers such as this one, State of MCP Server Security 2025: 5,200 Servers, Credential Risks, and an Open-Source Fix are gradually showing the increasing risks with MCPs across enterprises. MCP makes tool and resource access easier, but this also means it standardizes a high-leverage integration surface. Real-world deployments introduce protocol-level identity and access control risks (spoofing, confused-deputy/OAuth proxy failures, excessive permissions), plus governance risks like shadow MCP servers and insufficient auditability. In many patterns, secrets end up on endpoints or in local contexts that are easier to leak or compromise, creating direct paths to token theft and unauthorized tool use.

Shadow AI & Agents

This is the key risk we hear from practitioners. The rise of Shadow AI and unmanaged agents represents a massive, largely unmonitored expansion of the identity attack surface. Whereas traditional Shadow IT is mostly about employees adopting unapproved SaaS apps, Shadow AI involves employees and developers standing up autonomous agents and often connecting them to unapproved MCP servers outside centralized visibility, registration, and lifecycle management.

Because these agents are deployed outside policy, the organization can't reliably inventory what exists, who owns it, what it can access, or whether the underlying credentials are expiring and governed. In practice, shadow agents frequently lack clear ownership attribution, so when a "credential time bomb" (non-expiring keys, orphaned tokens, shared secrets) is discovered, there is no accountable team to rotate or decommission it. The net effect is fast-growing credential sprawl and unmanaged agent/tool connectivity, where a single leaked token can translate into broad, automated, cross-system action.

The New Enterprise Stack Emerging Now

In this paradigm, governance shifts from managing a human's **"who"** to managing an agent's "who" and **"why."** Human identity is typically a persistent credential used to open a door; agentic identity should be a dynamic and task-specific permission set that exists only for the duration of discrete intent.

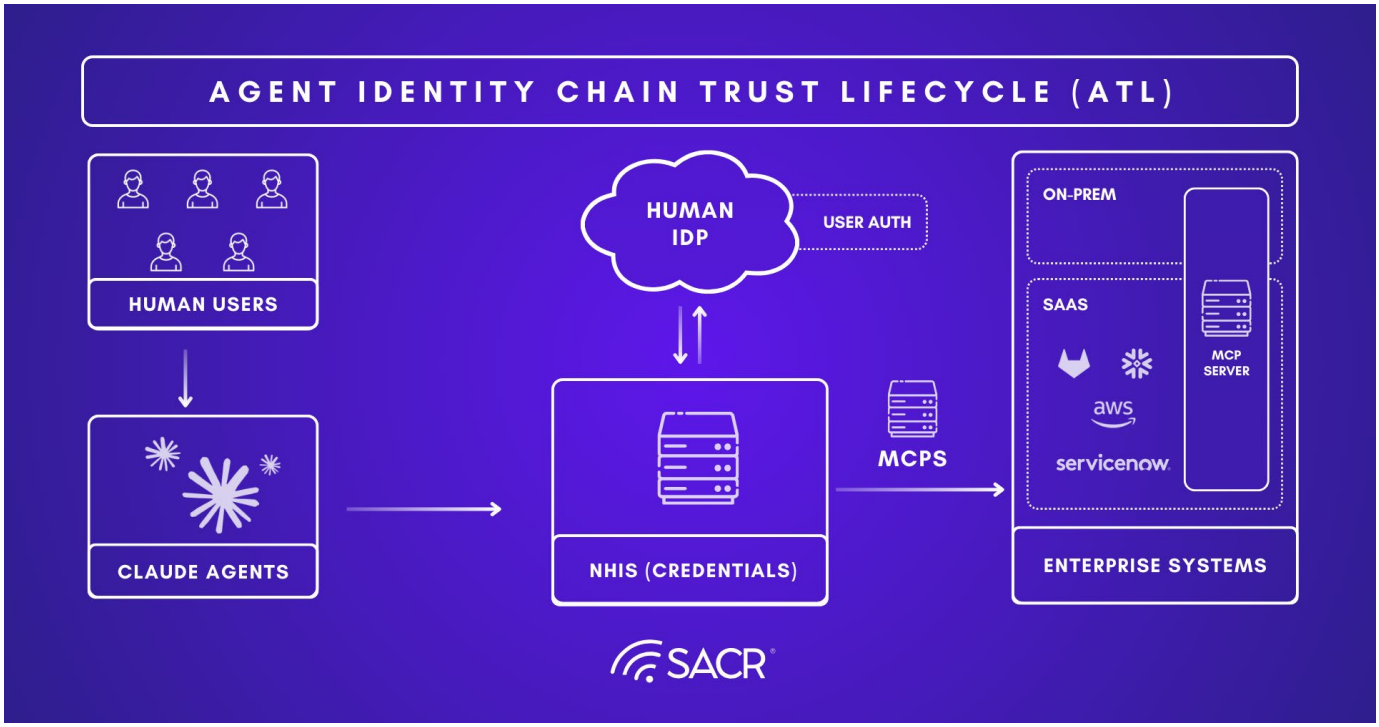
We believe that implementing an Agentic Access Management (AAM) model is the core way to maintain AI-driven velocity without turning the enterprise perimeter into a sieve. This requires transitioning from standing privileges to just-in-time where task-scoped identities exist only while an authorized action is in progress. Crucially, the architecture depends on separating:

- The **worker** (the agent),
- The **key** (the identity the agent uses to access systems),
- And the **broker layer** that governs the relationship between the two using policy and intent.

New Components of This New Architecture

The report defines three foundational primitives that become the building blocks of agentic identity security: NHIs, MCP, and AI agents (as categories of autonomous actors).

The identity chain: Agent → NHI / MCP / IDPs → Enterprise system



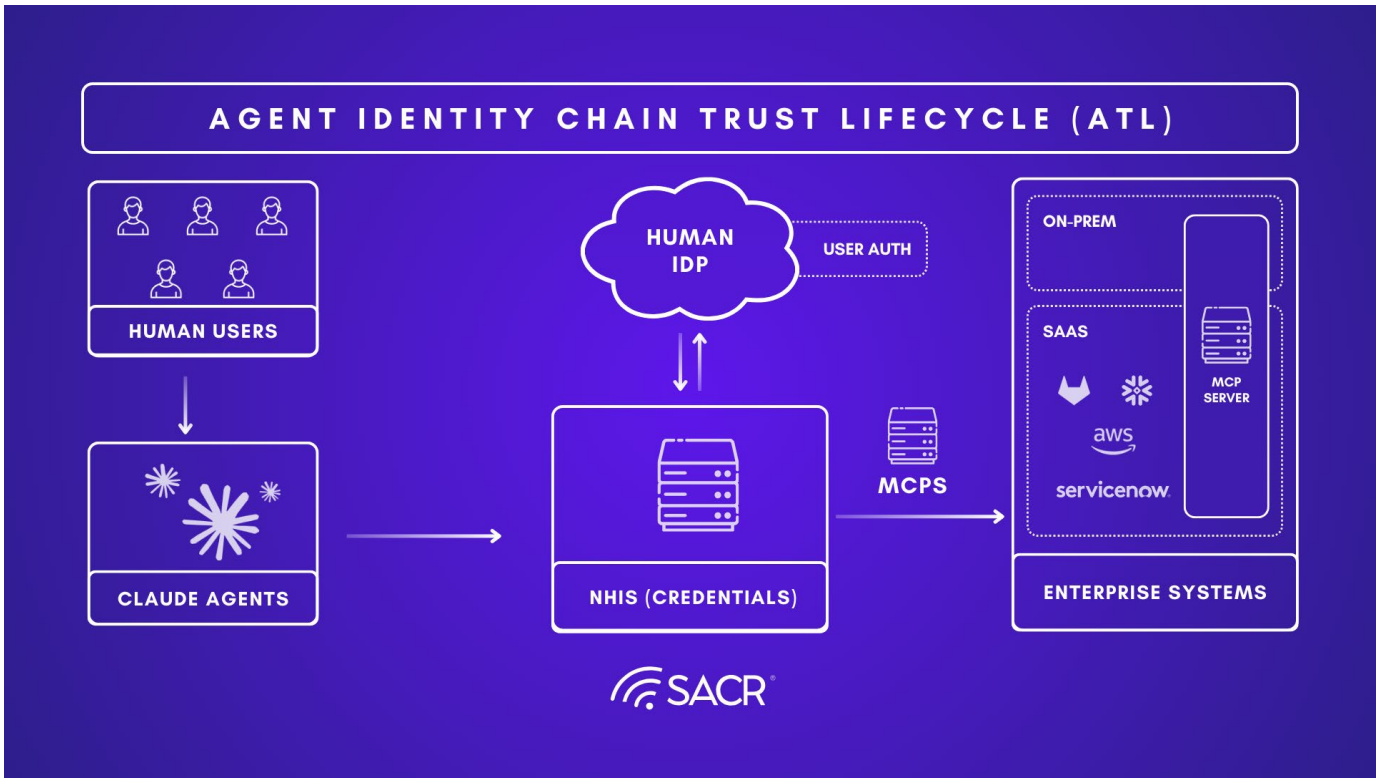
1. **Non-Human Identities (NHIs):** NHIs are the machine credentials that enable software-to-software communication, including OAuth apps, service accounts, API keys, and more. In the agentic era, these credentials become the passports that agents carry to open doors into enterprise systems.

2. **MCP as an integration bridge (and exposure surface):** MCP has as an emerging standard protocol that bridges AI agents to local and remote data sources. This bridge is foundational to agent functionality, but also expands the attack surface as discussed earlier. This is particularly the case when secrets are stored on endpoints to make MCP servers function

3. **New Types of AI agent categories that drive distinct governance patterns.** We've discussed all the various types of agents earlier. The most prominent that exist for enterprises includes delegated access agents that automate a specific user's workload. This is where the agent acts as a proxy for a specific person (e.g., a financial analyst using a Copilot to query portfolio data).

The new architectural diagram is the dependency chain where an agent is not a standalone actor. It is the first link in a chain where security depends on the NHI used to reach enterprise systems. Examples given include a SalesOps agent using an OAuth app to access Salesforce/SendGrid, or a DevOps copilot using a service account or PAT to interact with GitHub/Azure. The root risk is that these NHIs are frequently hard-coded or over-privileged effectively "master keys." Because they are static and long-lived, they create major credential leakage risk. For example, if the endpoint is compromised or the agent behaves unexpectedly, the key becomes an open door into sensitive systems.

Where the role of the IdP (the directory) vs Credential Broker Plays



The IdP

The IdP is the system of record for Authentication. It answers the question: "Who is this user or agent?" Its role is fundamentally to store directory information for humans and static service accounts. It issues the initial proof of identity (like an OIDC token or SAML assertion) used to log in as discussed earlier with Okta and Microsoft Entra ID (Azure AD). Traditional IdPs often lack visibility into non-human identities (NHIs) outside their specific cloud ecosystems (e.g., Azure Foundry misses AWS agents). They typically issue long-lived, static credentials ("forever credentials") that do not expire when an agent completes a task, creating security risks. The new vendors integrate with IDPs to validate the "Identity Lineage" i.e. verifying the human user who initiated the agent before passing control to the broker.

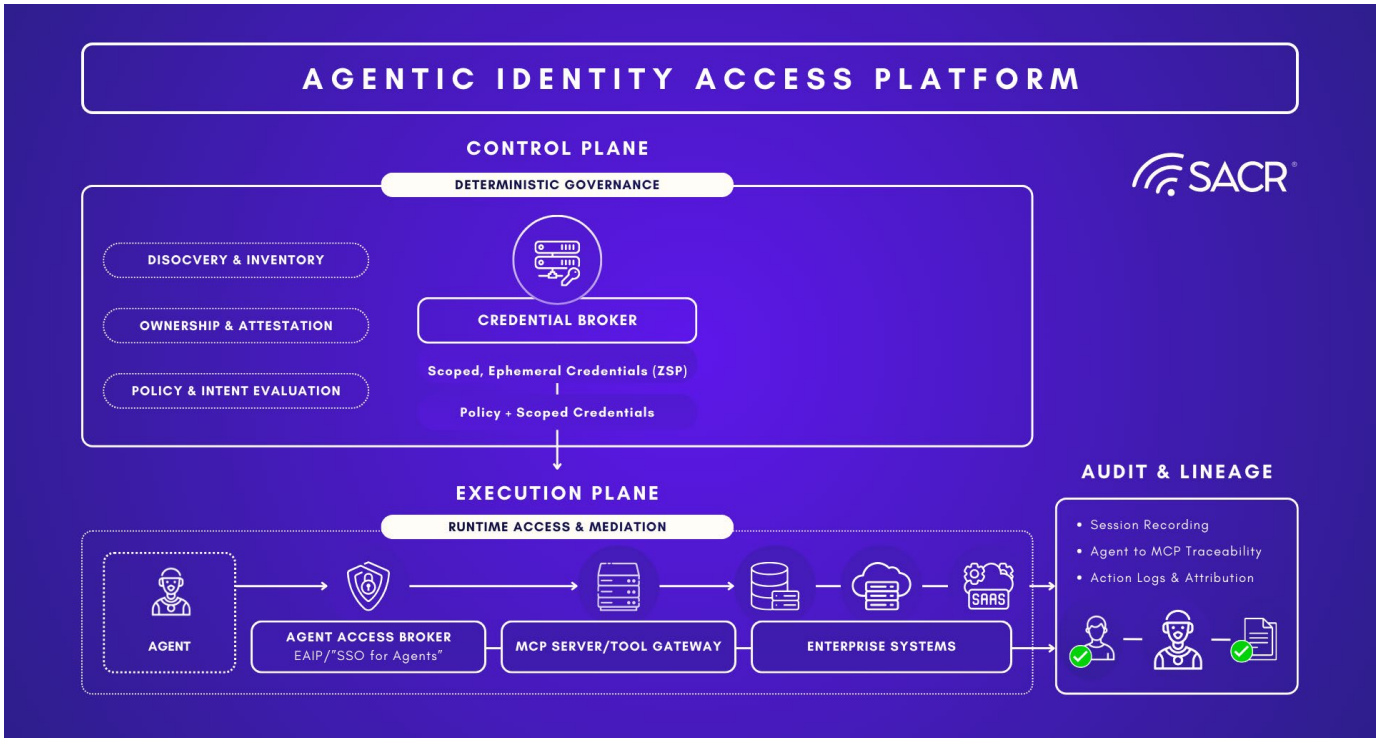
The "Workload" IdP

While the role of the human-centric IdP (like Okta or Entra ID) changes in an autonomous agent world scenario, the function of "an agnostic IdP" remains critical and actually becomes more complex. In a world of autonomous agents interacting with each other, the "Identity Provider" evolves from a directory of humans to a "Root of Trust" for workloads.

If Agent A tries to talk to Agent B, Agent B must verify "Who is Agent A?" to decide if it should answer. In autonomous systems, this role is filled by **machine identity providers** using standards like **SPIFFE/SPIRE** or Cloud IAM (e.g., AWS IAM instance profiles). In the workload, IAM world, we know that platforms like Aembit rely on this machine identity token exchange. Even for "Autonomous (Workload-Driven)" agents, the agent must present a valid identity token to the **MCP Identity Gateway** to receive the credentials needed for its task. Without a system to assert *which* software is running, the gateway cannot enforce policy.

Credential Broker: The Gateway / Control Plane

The Credential Broker (often implemented as an Identity Gateway or Control Plane) is responsible for Authorization and Fulfillment. It answers the question: "What specific key does this agent need right now to access this resource?"



It sits between the agent and the target resource. Instead of the agent holding a static password, the broker performs a **token exchange i.e. it** accepts the agent's identity token (from the IdP) and swaps it for a short-lived, downstream credential (e.g., a Salesforce session ID or AWS access key). For example, the following companies are embedded in this agentic workflow:

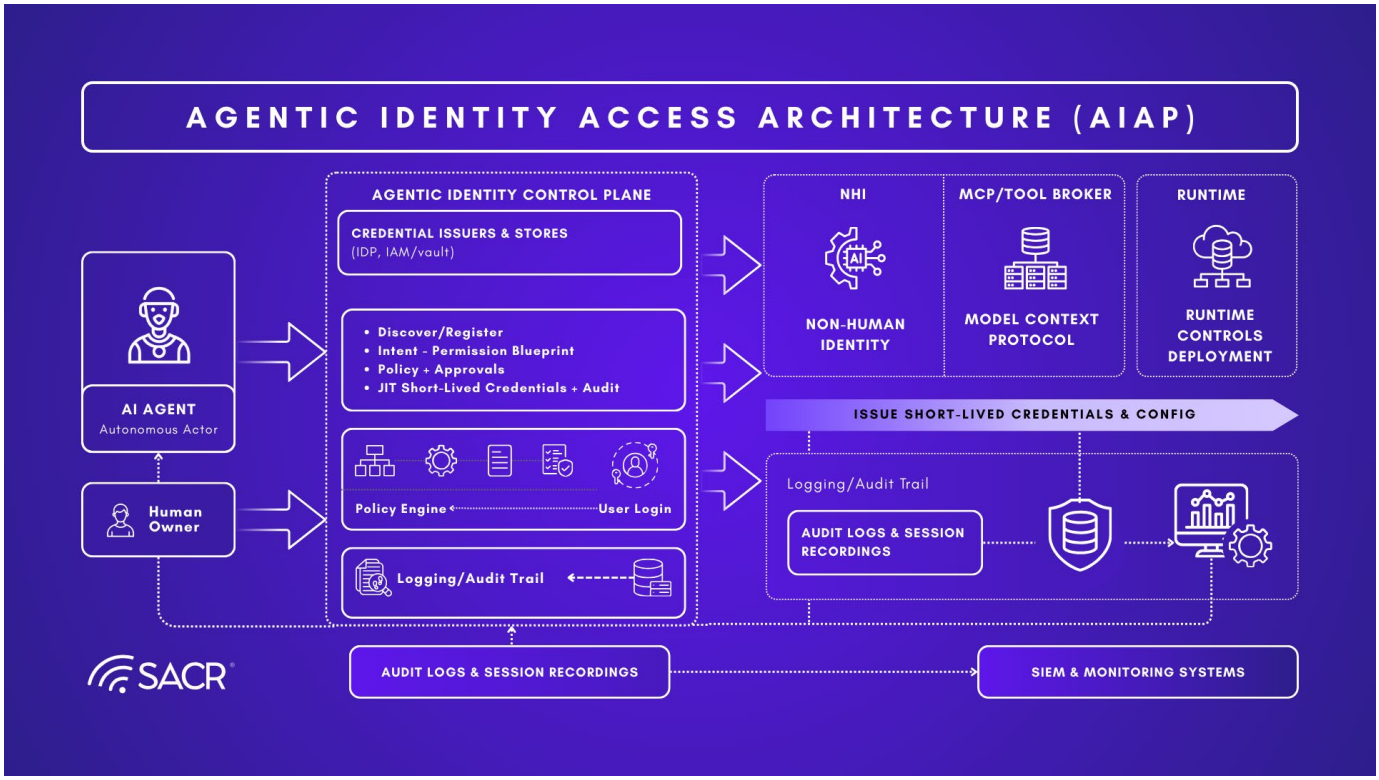
- **Aembit** acts as a broker via its **MCP Identity Gateway**. It intercepts the agent's request, injects the necessary credential, and forwards it to the resource. This ensures the agent *never* actually sees or holds the final secret and eliminating hardcoded keys.
- **Oasis Security** is described as a "broker for credential issuance." It analyzes the agent's intent (e.g., "update leads") and creates a temporary, scoped identity for that specific action, deleting it once the session ends.
- **Astrix Security** uses its **Agent Control Plane (ACP)** to broker the provisioning of secure-by-design agents with short-lived credentials at creation, rather than relying on static IdP service accounts.

The New Framework for Security (AIAP)

The report positions the future solution as an Agentic Identity Access Platform (AIAP) implementing Agentic Access Management (AAM): a “secure-by-design” architecture that governs a “new SSO for Agents.”

They converged around the following:

- Phase 1 discovers and inventories agents to attribute runtime behaviour.
- Phase 2 defines what is allowed, deterministically.
- Phase 3 ensures access exists only ephemerally.
- Phase 4 enforces whether access should continue.



Four strategic principles of Agentic Access Management (AAM)

Principle 1: Visibility, Inventory & Agent Registration (Continuous)

The popular framework across security is that as a first step, you can't manage what you don't know. The goal is to go beyond basic visibility, but to continuously enumerate AI agents + MCP servers + NHIs (API keys/tokens/service accounts/OAuth apps) + secrets + owners + connected resources, including shadow/unregistered artifacts as they spurn up.

The tricky challenge for practitioners is that inventories are fragmented across platforms (IDPs only show their own agents; SaaS and self-built agents get missed). Also, Agents are often "orphaned" at creation, and standard logs show only the service account acting, not the human who prompted it.

EAIPs must include automated discovery mechanisms where you scan endpoints (via EDR integrations or local scans) for running agent processes or config files, querying cloud environments for any app registrations that look like AI agents, pulling in records from SaaS platforms (e.g. checking if any OAuth client was created for an AI tool). The platform then creates an inventory (often an agent directory or identity graph) that lists each agent, its metadata (owner, where it runs, what tokens it uses). This is analogous to a user directory but for agents. Some EAIPs even map relationships, e.g., linking an agent to the secrets it uses and the resources it accesses, which helps visualize potential attack paths.

Principle 2: Authorization - The Intent Policy Layer

This principle focuses on the core control plane by shifting enterprises from role-based access to **intent-based authorization**, where governance occurs *before* any credentials are issued. The fundamental challenge is that agents require broad reasoning latitude ("greedy thinking") while enterprises cannot tolerate broad, static permissions that dramatically expand blast radius.

To resolve this, the Intent policy layer standardizes how agents request access by forcing an **intent declaration** where what the agent intends to do, not how it plans to authenticate. This declared intent is then analyzed and decomposed into a closed, task-specific set of concrete actions, from which a least-privilege permission blueprint is generated.

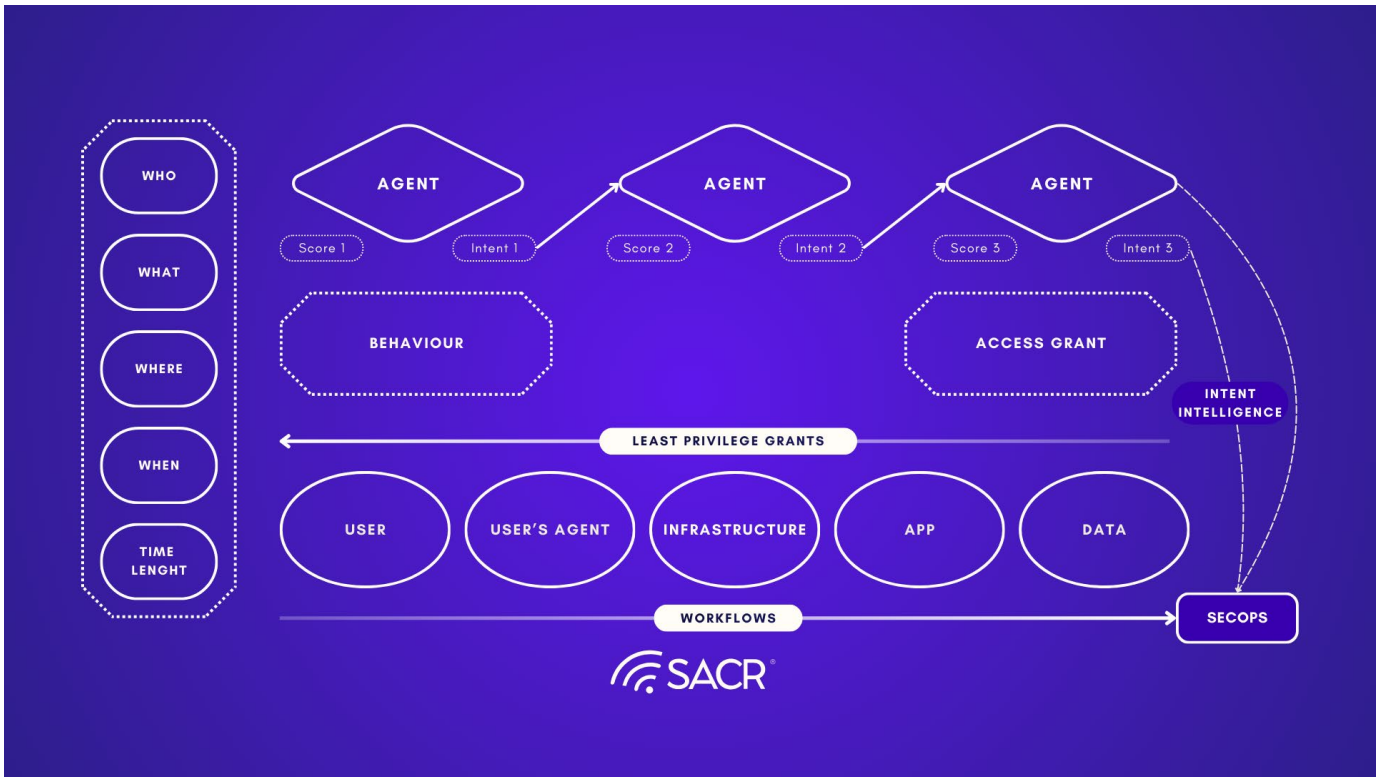
Platforms like Oasis operationalize this through an intent-aware analysis pipeline that translates high-level intent into precise technical permissions and evaluates them using **deterministic policy engines** (e.g., OPA), ensuring decisions are auditable, explainable, and not dependent on probabilistic LLM judgment. Critically, policy is evaluated **prior to credential issuance**, preventing agents from being deployed with standing privileges or "master keys." An Agent Control Plane can further enforce policy-at-creation, ensuring agents are born constrained rather than remediated later. Contextual guardrails extend this model by allowing step-up enforcement for high-risk actions such as bulk data access using human-in-the-loop or AI-in-the-loop approval workflows, as seen in Silverfort's intent-based enforcement approach. By decoupling intent from authentication and inserting deterministic authorization ahead of access, Phase 2 creates a governed, auditable translation layer that safely converts agent intent into narrowly scoped, approved permissions without sacrificing agent autonomy or enterprise control. Critically, Phase 2 stops at authorization: it determines **what access is allowed**, but does not yet issue credentials. That responsibility belongs to the next phase.

Our analyst Lawrence Pingree extensively written from last week. Please see everything in his report here titled: **The Future of Just-in-Time Trust (JIT-TRUST) for AI Users and Agents.**

The Future of Just-in-Time Trust (JIT-TRUST) for AI Users and Agents

Lawrence Pingree

Read full story



Phase 3: Broker & Inject With Intent-Governed Access Fulfillment (Zero Standing Privilege)

Phase 3 operationalizes intent-based authorization by binding deterministic policy decisions directly to credential issuance, enforcing Zero Standing Privilege (ZSP) as an execution model rather than a static posture. The core problem this layer solves is that long-lived credentials (API keys, service account secrets, embedded tokens) remain the dominant attack vector in modern breaches, especially in agent-driven workflows where credentials are frequently hardcoded, over-scoped, and difficult to rotate. In this phase, access is no longer something an agent possesses; it is something temporarily brokered, injected, and revoked per task.

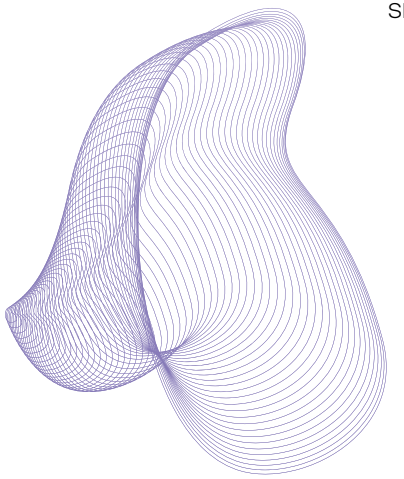
When an agent needs to act, it does not authenticate directly to the target system or hold downstream secrets. Instead, it initiates a handshake with a credential broker or access gateway, presenting its verified identity and previously authorized intent. The broker re-evaluates this request against deterministic, intent-aware policies derived from Phase 2 and, if approved, dynamically mints a short-lived, precisely scoped credential aligned only to the specific action being performed. This may take the form of a five-minute cloud session token, a narrowly scoped OAuth token, or a one-time database credential.

Crucially, this credential is injected at runtime. For example, directly into the API request or ephemeral environment context without ever being exposed to the agent's code or stored on disk. Vendors implement this pattern in different ways, but the outcome is consistent: agents operate with no standing access, credentials exist only for the duration and scope of an approved intent, and privilege is automatically revoked upon task completion or expiration. By tightly coupling intent-aware, auditable policy enforcement with just-in-time credential brokerage, Phase 3 ensures that even if an agent or its execution environment is compromised, stolen credentials cannot be reused, replayed, or leveraged for lateral movement. Policy becomes executable, access becomes ephemeral, and ZSP becomes enforceable at machine speed.

Phase 4: Runtime Enforcement, Audit, and Automated Response

Phase 4 addresses the unavoidable reality of agentic systems: authorization alone is insufficient once execution begins. Even when an agent's intent has been correctly analyzed (Phase 2) and access has been safely brokered through ephemeral credentials (Phase 3), agents remain susceptible to non-deterministic behavior, prompt injection, tool misuse, or logic drift during runtime.

This phase, therefore, focuses on continuous monitoring, enforcement, and rapid termination, ensuring that access remains valid not just at the moment of issuance but throughout execution. The core challenge is that an agent can become dangerous *after* it has been legitimately authorized by deviating from its approved behavior, escalating actions, or accessing data outside its expected baseline.



Phase 4 introduces a runtime threat layer that continuously observes agent activity streams and correlates signals into contextual threat cases rather than isolated alerts. Platforms such as Astrix, Silverfort, and Cyata emphasize correlating usage patterns, historical baselines, and contextual metadata to distinguish between legitimate high-volume automation and malicious or compromised behavior. A central control in this layer is behavioral drift detection: if an agent authorized for read-only operations attempts write actions, begins accessing unfamiliar systems, or rapidly expands its data footprint, the system flags this deviation as anomalous.

Crucially, the effectiveness of Phase 4 depends on the design decisions made earlier because identities and credentials are scoped and time-bound, enforcement becomes decisive. Instead of attempting complex remediation, the system can invoke an immediate kill switch by revoking or simply refusing to renew ephemeral credentials, terminating sessions in real time and cutting off access at machine speed. In parallel, the platform maintains comprehensive audit trails that bind runtime actions back to agent identity, declared intent, approvals, and ownership, supporting forensic analysis, compliance reporting, and governance workflows. Over the agent lifecycle, this layer also automates credential rotation, revocation, and decommissioning, ensuring that dormant or orphaned agents cannot persist as latent risk. Integrated with identity governance processes, Phase 4 enables periodic review of active agents, ownership validation, and evidence-based AI governance for auditors increasingly focused on autonomous system risk. Together, runtime monitoring, scoped and time-based identities, and automated termination transform access from a static grant into a continuously enforced state—ensuring that when agents drift, misbehave, or are compromised, trust is withdrawn instantly rather than discovered after damage is done.

Actionable Recommendations for IAM & Identity Security Leaders

The transition to agentic systems does not require ripping out existing IAM infrastructure, but it *does* require acknowledging that traditional identity controls were never designed to govern the new categories of agents at machine speed. The most successful programs will treat agentic identity as an **overlay control plane**, introduced incrementally, and enforced where it delivers immediate risk reduction.

Phase 1: Legacy Identity Debt Cleanup (Immediate Risk Reduction)

Before agents can be governed safely, legacy identity debt must be addressed. These new categories of agents will inevitably discover and exploit it.

- Identify and rotate unmanaged database users, local accounts, and shared credentials.
- Inventory existing service accounts and API keys to surface orphaned non-human identities.
- Remove hard-coded secrets from endpoints, scripts, and CI/CD pipelines to eliminate easy leakage paths.

This phase is unglamorous but critical. Without it, agentic controls sit on top of a compromised foundation.

Use existing IdP investments

Most enterprises should continue leveraging their existing identity providers (Okta, Entra, Ping) for human authentication, MFA, and step-up controls. However, leaders must be clear-eyed: IdPs are a substrate, not a solution, for agent governance. Enterprises need something new. The agent problem is not authentication, it is delegation, authorization, and runtime control. We recommend treating the IdP as the anchor for user identity, but introduce agent-specific controls above it that manage intent, scope, and execution. Conflating "agent login" with "agent governance" will stall progress.

Phase 2: Governance & Visibility (Make Agents Legible)

Once legacy exposure is reduced, we recommend focusing on visibility and accountability.

1. We have outlined different mechanisms above. We recommend organizations use API / EDR- and endpoint-based discovery to identify shadow agents, local developer tools, and unsanctioned automation.
2. Mandate owner attestation and business justification for every discovered agent.
3. Establish session-level logging or recording for high-risk agent workflows to ensure auditability and post-incident reconstruction.

At this stage, the goal is to ensure no agent operates anonymously.

Leverage Idp to connect attribution and ownership

Before introducing new policy engines or credential brokers, organizations must establish clear attribution. Every agent, whether an internal copilot, automation bot, or customer-facing AI workflow must have a human owner, an explicit business purpose, and a defined blast radius. This is non-negotiable. Without ownership, security teams cannot govern lifecycle, enforce policy, or respond to incidents. Prioritize user-driven agents first (internal copilots, support bots, productivity agents), where blended identity provides immediate value by separating user permissions from agent permissions while preserving accountability. This alone reduces over-privilege and simplifies incident response.

Phase 3: Intent-Aware Autonomy (Controlled Execution)

With visibility and ownership in place, organizations can safely enable autonomy.

- Migrate new AI projects to a brokered access model, where agents authenticate through a centralized control plane rather than directly to target systems.
- Enforce policy-driven, intent-aware authorization that translates declared tasks into minimal, auditable permission scopes.
- Issue short-lived, task-specific credentials and phase out standing privileges entirely.

This is where Zero Standing Privilege becomes operational, not theoretical. Agents gain autonomy without inheriting permanent access.

Future Trends & Predictions

This category is rapidly evolving and faster than we can imagine, so it's hard to make a guess on the future. However, one thing we can certainly say is that autonomous systems will force identity security to evolve faster than any prior shift in enterprise IT. The central issue is not that agents are "new users," but that they execute actions at machine speed with non-deterministic behavior and delegated authority. This is a combination that breaks the assumptions behind legacy IAM, IGA, and even modern PAM. As a result, the market is converging on three structural shifts that will define the next generation of identity and access control.

1) The rise of the centralized identity broker: "SSO for Agents"

Enterprises will increasingly stop allowing agents to connect directly to SaaS APIs, cloud consoles, and infrastructure endpoints with embedded credentials. Direct integration is operationally convenient, but security-incoherent: it creates a sprawl of long-lived tokens, unclear accountability, and inconsistent policy enforcement across every agent toolchain. The emerging replacement is a centralized broker - almost similar to an "SSO for Agents", but one that becomes the default entry point for agent access. This broker does more than authenticate; it standardizes *how agents request access,* translates declared intent into minimal permissions, mints short-lived credentials, and enforces policy in one place. The pattern mirrors what happened with SSO for humans: enterprises eventually centralized authentication because distributed login created chaos.

The difference now is that centralized brokering must extend beyond login into intent, delegation, and runtime governance, because agents are not just authenticating, they are executing workflows. In the next phase of maturity, this broker becomes a control plane that security teams can operate like infrastructure: measurable, enforceable, auditable, and resilient.

2) Agent-to-Agent (A2A) protocols become a first-class governance problem (near-term future risk)

Today most controls assume a human is the initiating actor and the agent is an assistant. That assumption will not hold. The more agentic systems mature, the more work will shift from "human → agent" to agent → agent: planners delegating to executors, copilots spawning sub-agents, and workflows chaining across tools and organizations. This creates a new trust boundary: how agents authenticate and verify each other, how authority is delegated, and how intent is propagated across multi-step autonomous workflows. Security architects will need to govern A2A communication the way they once governed service-to-service communication except now the transactions are probabilistic, goal-driven, and often mediated through tool protocols rather than APIs alone.

Expect enterprise policy to evolve into "workflow governance": what types of agent delegation are allowed, what verification must occur before an agent can call another agent, and how downstream actions remain constrained to an upstream-approved scope. A2A trust will become a major design constraint for both product builders and defenders.

3) A unified access layer collapses silos between NHI, workload identity, and agentic identity

The distinction between “non-human identity,” “workload identity,” and “agent identity” will not survive at scale. These categories exist today because they are managed by different teams, governed by different tools, and used in different contexts. But agentic systems blur the boundaries: agents run as workloads, consume NHIs, delegate from users, and operate across SaaS, cloud, and endpoint surfaces.

The inevitable outcome is convergence into a single dynamic access layer where identity becomes a temporary state granted continuously based on context, validated intent, and bounded execution. In that future, “identity” stops being a directory object and becomes an ephemeral control primitive: a short-lived authority token representing who is acting, on whose behalf, for what purpose, and within what scope. This unification will also reshape governance. Instead of periodic access reviews and static entitlement models, organizations will move toward continuous enforcement: access is granted for an objective, not assigned as a standing condition.

What this means for CISOs and IAM leaders

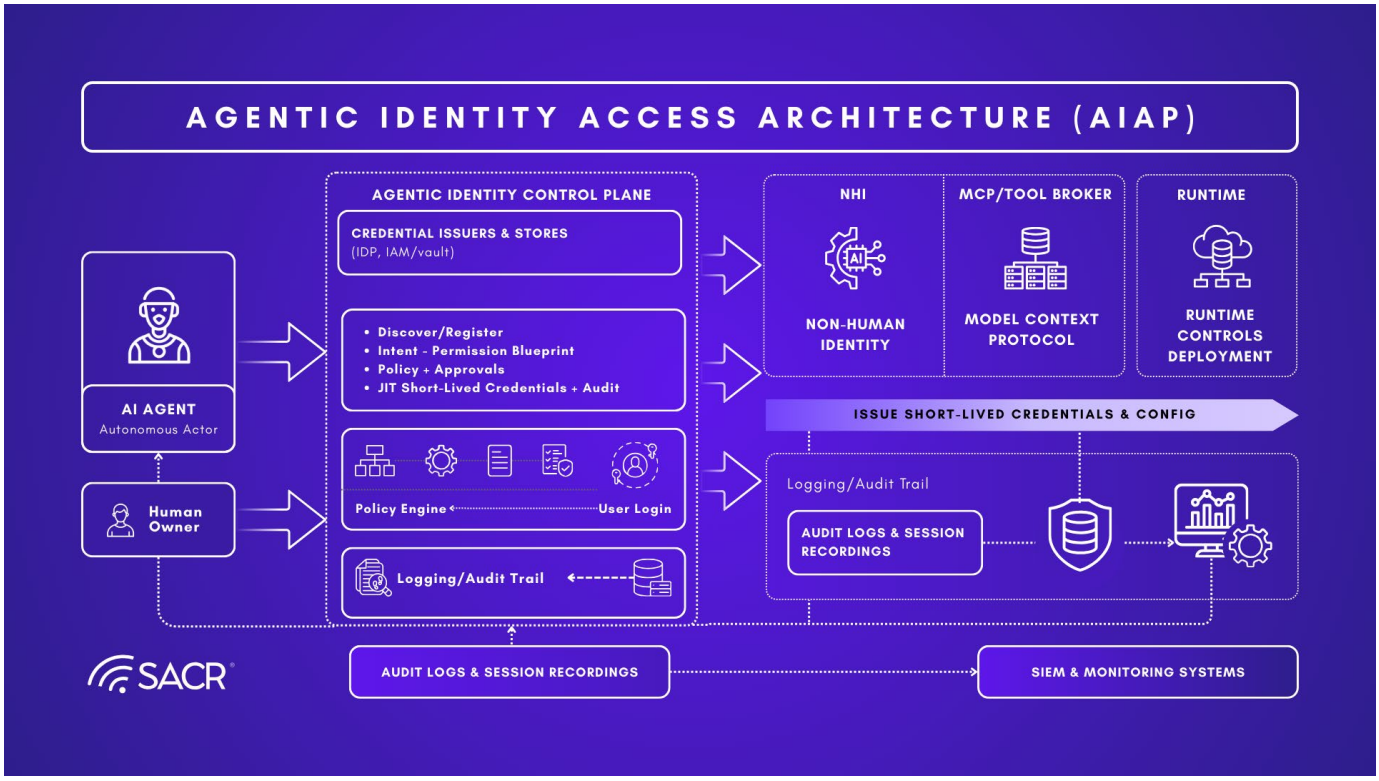
These trends point to a simple conclusion: identity is becoming the control plane for autonomy. The security goal is no longer just to prevent unauthorized login; it is to ensure that autonomous action remains attributable, bounded, and terminable in real time. Organizations that win will treat agentic identity as an engineered system: inventory first, deterministic policy next, brokered ephemeral access by default, and runtime kill-switch enforcement as a standard operating capability. The ones that lose will treat agents as "just another app integration" and will discover, after the first incident, that machine-speed execution makes legacy identity failures catastrophic.

Key Vendors Solving This Risk

The remainder of the report grounds this architecture in real-world implementation patterns through case studies and representative vendors. We've partnered with 5 vendors who are leading innovation in this new ecosystem. They include:

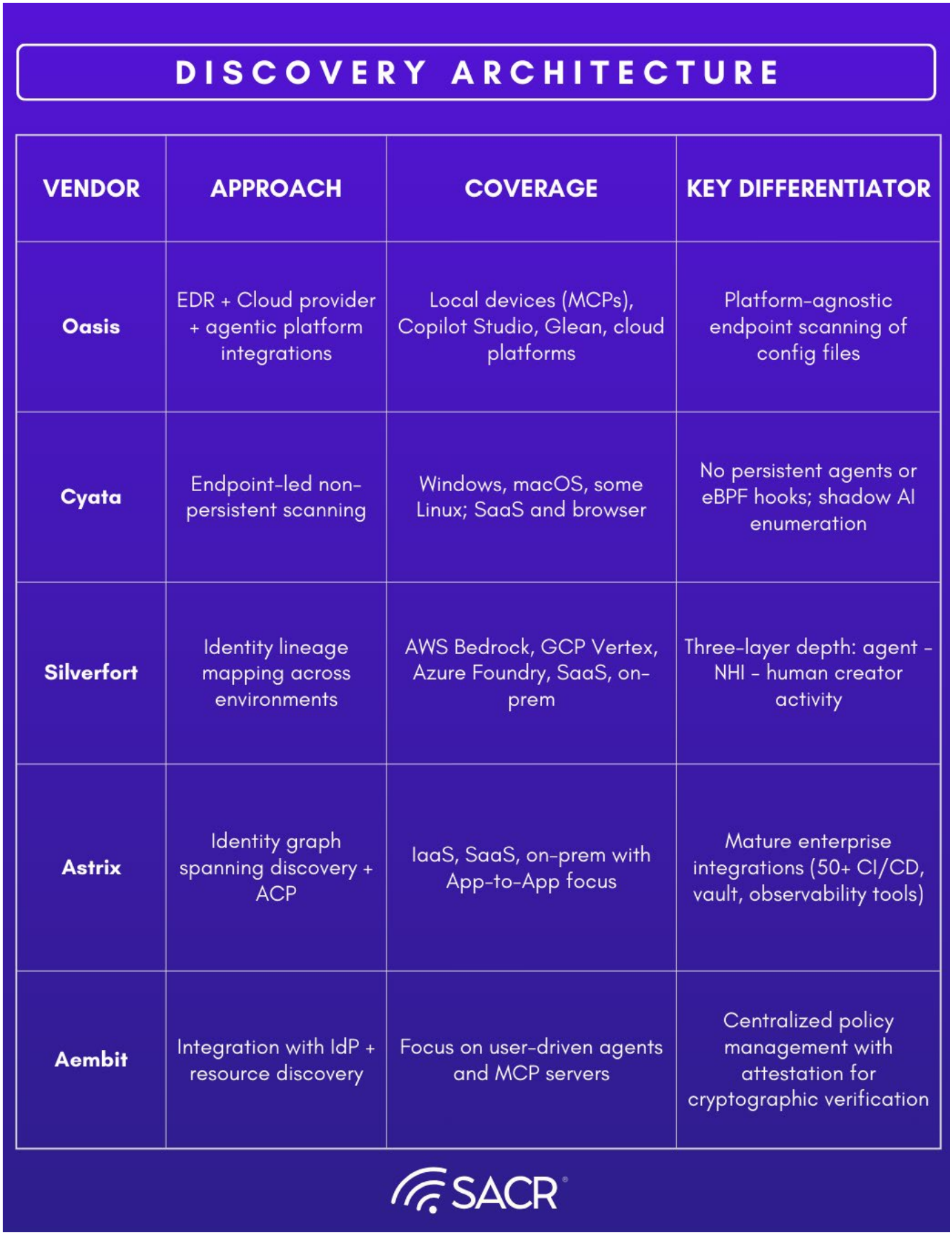
1. Astrix
2. Oasis
3. Aembit
4. Cyata
5. Silverfort

Each vendor is analyzed based on where it fits across the four phases: 1) visibility and inventory, 2) intent-to-policy translation, 3) credential brokerage and secretless injection, and 4) runtime enforcement and response, highlighting both strengths and gaps. The goal is not to declare a single "winner," but to provide security and IAM leaders with a practical lens for evaluating how the market is converging on a unified access control plane for autonomous systems, and how to assemble an end-to-end program that is deployable today while aligned with the inevitable future: centralized brokering, agent-to-agent governance, and identity as a temporary, continuously enforced state.



Vendor Overview Differences

Appendix



DISCOVERY ARCHITECTURE

VENDOR	APPROACH	COVERAGE	KEY DIFFERENTIATOR
Oasis	EDR + Cloud provider + agentic platform integrations	Local devices (MCPs), Copilot Studio, Glean, cloud platforms	Platform-agnostic endpoint scanning of config files
Cyata	Endpoint-led non-persistent scanning	Windows, macOS, some Linux; SaaS and browser	No persistent agents or eBPF hooks; shadow AI enumeration
Silverfort	Identity lineage mapping across environments	AWS Bedrock, GCP Vertex, Azure Foundry, SaaS, on-prem	Three-layer depth: agent - NHI - human creator activity
Astrix	Identity graph spanning discovery + ACP	IaaS, SaaS, on-prem with App-to-App focus	Mature enterprise integrations (50+ CI/CD, vault, observability tools)
Aembit	Integration with IdP + resource discovery	Focus on user-driven agents and MCP servers	Centralized policy management with attestation for cryptographic verification

SACR

Appendix 2

VENDOR	CORE MOTHODOLOGY	KEY STRENGTH	TARGET USE CASE
Astrix Security	NHI-Centric	Deep discovery of API keys/OAuth apps across SaaS/IaaS; identifies hard-coded credentials.	Large enterprises needing to map complex identity dependencies.
Oasis	Intent-Aware	"Agent-centric view" linking agents to the NHIs they leverage; end-to-end session recording.	Organizations requiring high auditability and human-reviewable intent analysis.
Silverfort	Unified Identity	Seamless integration with existing legacy IAM (AD, Okta, Entra ID) stacks.	Bridging the gap between traditional and agentic identity for hybrid environments.
Aembit	Workload Identity	Managing secure access between cloud workloads and microservices.	Developer-heavy environments focused on CI/CD and workload-to-workload security.
Cyata	Agent-Centric	Pioneers in reimagining the agentic identity control plane from the ground up.	Organizations seeking a purpose-built architecture for autonomous agent governance

SACR


Astrix

Astrix Security: Identity-First Platform for AI Agent Security

Astrix Security is emerging as a key platform at the convergence of AI agent security and non-human identity (NHI) security. As AI agents increasingly operate as intelligent, LLM-based services with direct access to enterprise systems and data, non-human identities become the core mechanism that enables what those agents can access and affect across the organization. They were built on the foundational insight that AI agents cannot be secured without securing the non-human (machine) identities that power them.

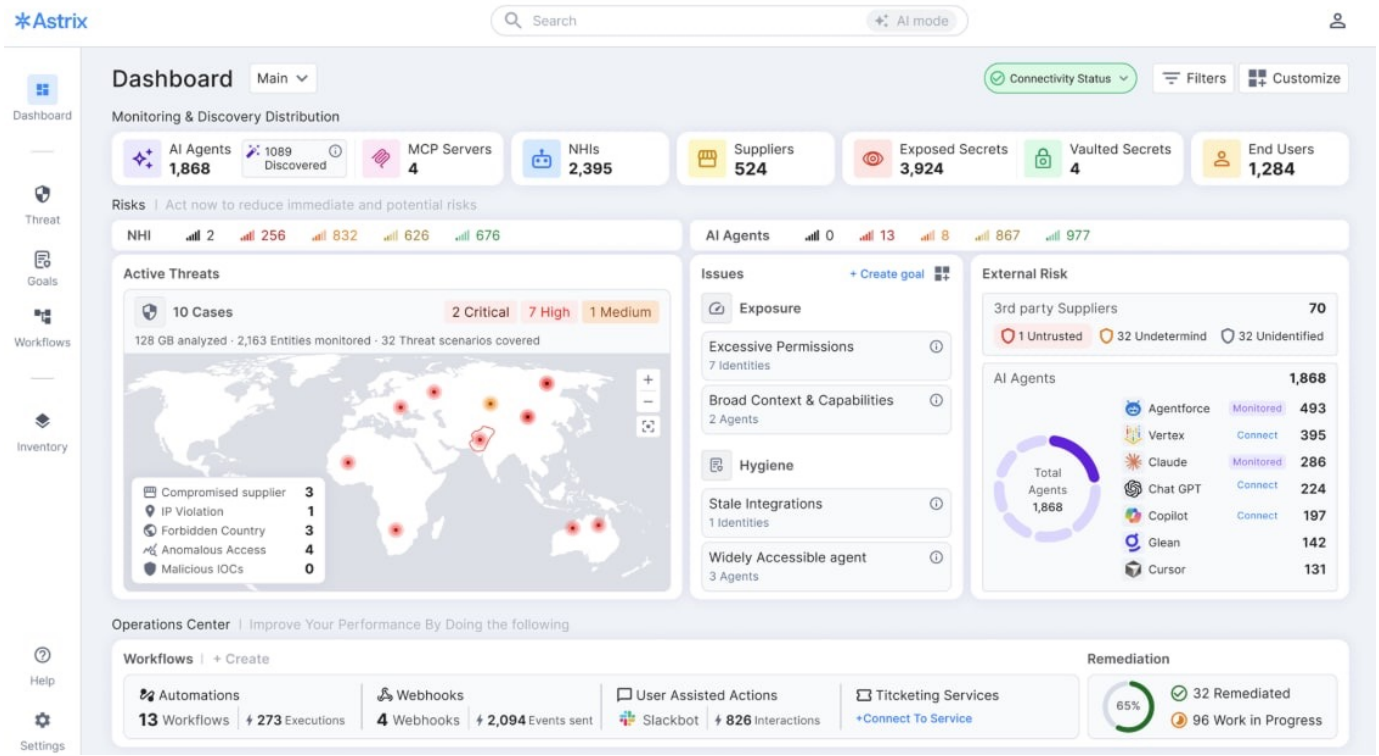
Founded in 2021 and now with $85 million in funding from leading firms including BVP, Menlo Ventures, and Anthropic, Astrix pioneered the NHI security category and coined the term at RSA Conference 2023, where the company was an RSA Innovation Sandbox finalist. Astrix also leads the OWASP NHI Top 10 project and, in collaboration with CIS, the AI Agent Cybersecurity Matrix, contributing to industry guidance for securing AI agent environments. The company has evolved its platform to become the first amongst its peers to deliver an end-to-end approach to AI agent security through Astrix's Agent Control Plane (ACP) and a unified Discover-Secure-Deploy framework.

Astrix's platform applies a threat-driven, continuously monitored security approach to AI agents and non-human identities through three integrated capabilities:

1. Discover
2. Secure
3. Deploy

Discover & Inventory AI Agents

The Discover capability delivers continuous, real-time inventory of AI agents, NHIs powering them, as well as MCP servers across cloud (AWS, Azure, GCP), SaaS (Salesforce, GitHub, Slack, Workday), on-premises



The Astrix Platform Dashboard

systems (Active Directory, SAP), and AI platforms (OpenAI, Copilot, Gemini, Anthropic, Agentforce), with the ability to extend coverage to additional platforms through a bring-your-own-source (BYOS) model. This inventory is enriched with contextual intelligence that maps ownership, permissions, accessed resources, and usage patterns, enabling organizations to assess risk and prioritize remediation efforts.

MCP Servers, Secrets, and Shadow AI Coverage

The platform extends discovery to MCP servers and secrets across the entire technology estate spanning secret vaults (HashiCorp, AWS Secrets Manager), identity providers (Okta, Entra ID), and even endpoint devices via EDR integrations (CrowdStrike, SentinelOne, Microsoft Defender). It extends to official and unofficial MCP servers, whether deployed locally on developer machines or remotely in cloud environments, capturing both sanctioned implementations and home-grown integrations that often proliferate without security oversight. Astrix also applies advanced fingerprinting (without a direct integration) techniques to detect shadow AI agents including the use of unauthorized AI tools like ChatGPT, Claude Code, Cursor, or custom-built agents that bypass IT approval processes.

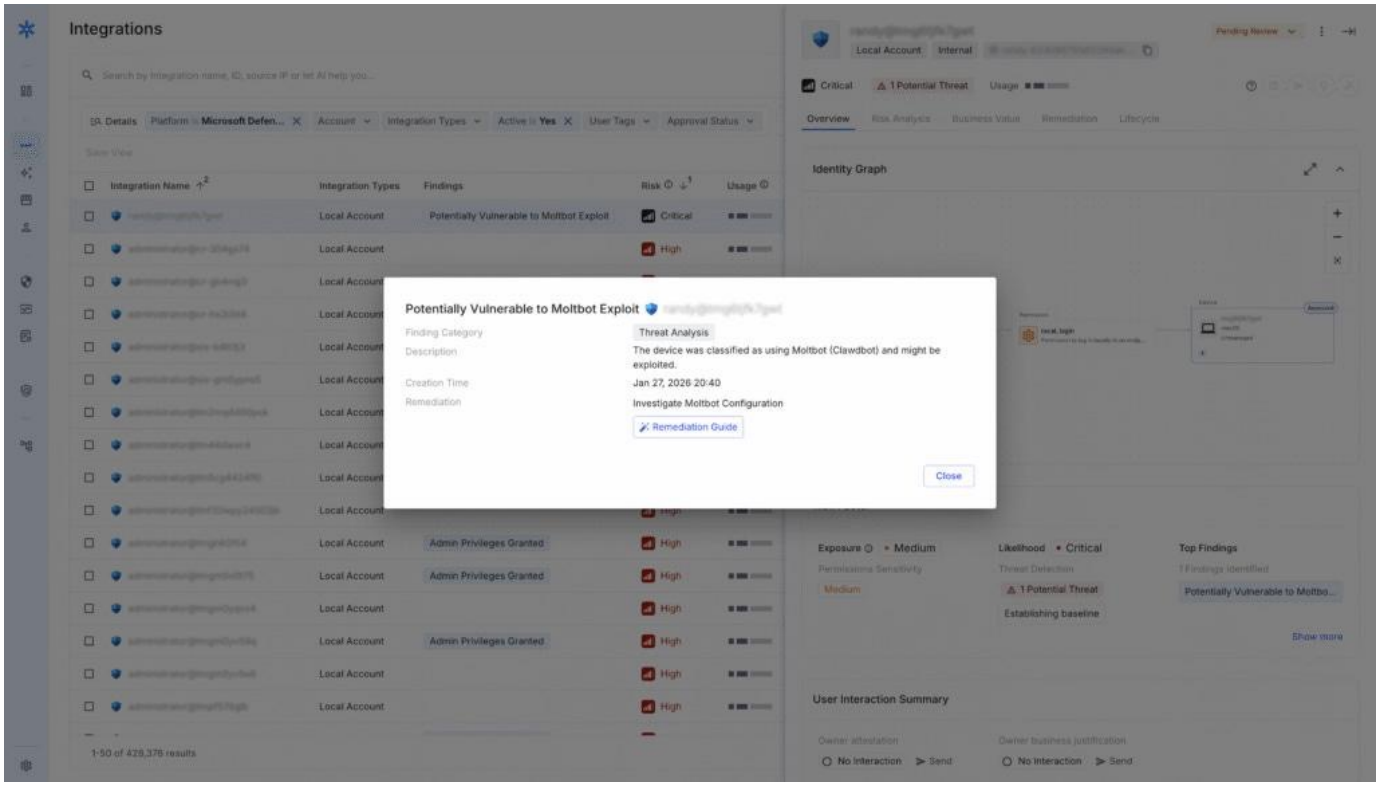
As Astrix operates at the non-human identity layer, with visibility into credentials, service accounts, API keys, OAuth applications, and MCP-related identities, the platform is positioned closer to the underlying access mechanisms AI agents rely on. This proximity enables fingerprinting methods that are difficult to replicate for tools operating only at the AI application or prompt layer.

Fingerprinting is further informed by forensic behavioral analysis, combining static identity attributes with dynamic usage signals and behavioral patterns over time. This allows Astrix to detect shadow AI activity based on how agents access systems and data, rather than relying solely on static configuration or tool identification.

Astrix enriches every discovered AI agent and identity with contextual attributes such as Owners & consumers (human attribution) or such as permissions (actual vs granted).

Secure: Posture Management & Threat Detection

The next step we saw from their product is that it has the capability to address posture management (identifying excessive privileges, configuration weaknesses, hard-coded credentials, and



OpenClaw/Moltbot exploit alert as flagged in the Astrix platform

ownership gaps), while also detecting anomalous behavior and compromised identities through Astrix's Agentic Detection & Response (ADR).

An important aspect of AI agent security addressed by Astrix is third-party and supply-chain risk. As enterprises increasingly adopt AI through procured platforms, SaaS integrations, agentic extensions, and MCP servers, a growing portion of agent access and execution originates outside direct organizational control. These external dependencies introduce high-impact risk, as compromised vendors, integrations, or extensions can operate with privileged non-human identities inside enterprise environments.

This risk was illustrated just recently in Astrix's analysis of MoltBot, one of the first widely observed autonomous AI agents, which demonstrated how agentic extensions and external integrations can evolve rapidly, accumulate privileged access, and operate beyond the visibility of traditional security controls, creating meaningful enterprise exposure.

OpenClaw/Moltbot exploit alert as flagged in the Astrix platform

Astrix has demonstrated a pattern of proactive identification and mitigation of such risks before public disclosure or exploitation. In multiple cases, the platform identified compromised or misused third-party non-human identities, including incidents where affected vendors were notified directly.

In another example, Astrix identified malicious AI-powered Chrome extensions that were subsequently blocked by a majority of customers before being publicly disclosed as malicious. This focus on third-party identity exposure allows organizations to reduce supply-chain risk tied to AI agents rather than relying solely on vendor attestations or delayed disclosures.

Astrix's Secure capability addresses the endemic hygiene problems that plague non-human identities and AI agents through comprehensive posture management that identifies and remediates excessive privileges, configuration weaknesses, credential exposure, and ownership gaps across the entire agentic ecosystem, where identity misconfiguration directly translates into enterprise-level risk.

The platform performs least-privilege analysis by comparing permissions granted to each NHI against actual API usage over a configurable baseline period (typically 90 days), surfacing over-permissioned accounts where granted entitlements far exceed functional requirements expanding the blast radius of potential compromise. For example, flagging a GitHub service account with repository admin rights across all repos that only performs read operations on three specific repositories for privilege reduction. Configuration assessments detect vulnerable setups including agents with write/delete permissions that should be read-only, NHIs exposed to the public internet without IP restrictions, service accounts shared across multiple applications, and secrets stored in non-rotatable formats or embedded in application code rather than retrieved from vaults, as seen in recent real-world agentic incidents.

Astrix automatically assigns ownership by tracing the creating user from audit logs and system metadata, then enforces attestation workflows where owners must periodically confirm the continued business need for each agent and NHI under their responsibility. This creates a forcing function for lifecycle hygiene where agents that cannot be attested (because the owner departed or no longer recognizes the entity) are flagged for decommissioning. Organizations can define and continuously audit policy compliance across dimensions such as "all production agents must use short-lived credentials from vaults," "no service accounts may have write access to HR systems," or "agents processing PII must have designated DPO ownership." These controls are applied continuously rather than through periodic review cycles, allowing security and IAM teams to identify risk as access and ownership change, not weeks or months later.

These posture findings, third-party exposures, and real-world agentic incidents are unified through Astrix's Threat Center, which provides a single operational view of AI agent and non-human identity threats and is continuously informed by Astrix's ongoing research and incident analysis.

Behavioral Threat Detection & Automated Response

Beyond static posture hygiene, Astrix provides behavioral threat detection through its Agentic Detection & Response (ADR) capability, which establishes baseline behavioral profiles for each agent and NHI, then alerts on anomalous activity that deviates from normal patterns. They address the critical blind spot where traditional security tools monitor human user behavior but lack visibility into non-human identity activity. Behavioral baselines are evaluated in the context of identity usage and access paths, allowing Astrix to detect risk when agents begin accessing new systems, expanding permissions, or operating outside expected environments.

Deploy Securely with the Agent Control Plane (ACP)

Deploy represents the stage where Astrix applies its full platform capabilities once organizations have visibility and security controls across their AI agents, MCPs, and NHIs.

Deploy brings together these capabilities together through Astrix's Agent Control Plane (ACP), which enables secure-by-design agent provisioning through just-in-time, short-lived, precisely-scoped credentials issued via API, eliminating the need for developers to embed long-lived secrets in configuration files while enforcing policy guardrails, ownership attribution, and maintaining complete audit trails for compliance.

Conceptually, Agent Control Plane functions as a Zero Trust access layer for AI agents, applying identity-aware, least-privilege controls at the moment access is requested rather than relying on static credentials.

Agent Control Plane (ACP) enables organizations to provision AI agents that are secure-by-design from creation, significantly reducing the risk exposure window associated with agent deployment. Traditional agent deployment requires developers to provision long-lived credentials (API keys, service account passwords, OAuth tokens) with broad permissions, then embed those credentials in configuration files, environment variables, or code repositories where they remain static for months or years, creating a persistent attack surface. This model creates a persistent attack surface that scales as AI agent adoption increases.

Voice of Customer

Astrix serves Fortune 1000 enterprises including Autodesk, NetApp, Stanford University, Databricks, Workday, HubSpot, and Ecolab, achieving 0% customer churn and 140% net revenue retention metrics that reflect deep product-market fit in a category the company defined.

Leading Travel Company (Customer Experience)

To validate market dynamics, we spoke with a senior security leader at a Fortune 500 travel company using Astrix to govern AI agents while maintaining aggressive AI adoption velocity. The organization employs approximately 600 developers out of 1,500 employees and operates under a CEO mandate to "embrace AI" and "use AI as much as possible with maximum speed." The core challenge was securing AI agent deployments without creating friction that would drive shadow IT.

The Value of Astrix in Practice: Visibility, Detection, and Control

This company's approach centers on detective controls rather than preventive gates that would slow development velocity. Astrix enables this balance by providing complete visibility into the organization's AI agent ecosystem delivering a comprehensive inventory of all GPTs, their associated connectors, and access permissions.

Astrix provided a complete inventory of GPTs and their connectors," the customer noted, allowing the security team to understand exactly which agents exist, what data they access, and who controls them. This foundational visibility extended to identifying GPTs open to all employees, those with confidential data, and agents with over-permissive API keys: blind spots that previously existed in their environment despite governance policies and committee approvals, which the customer characterizes as "soft controls."

The operational integration of Astrix into the travels company's existing security infrastructure demonstrates the platform's maturity. Alerts flow into the enterprise SIEM and are processed through established SOC workflows, eliminating the need for standalone monitoring. "Astrix alerts feed into the SOC operations process via SIEM," he noted, where analysts investigate with resource owners and can mark alerts with "valid justification" to close the loop. The platform prioritizes a defined set of 8-9 high-risk scenarios including HR employees creating GPTs with confidential employee data, DBAs connecting ChatGPT to BigQuery with over-privileged API keys, and agents with write access to resources that may not be appropriate.

This detective approach allows security to intervene when necessary while maintaining development velocity: developers can create agents without notification if they don't meet high-risk criteria (customer-facing, processing PII, or material impact), relying on Astrix to surface policy violations when risk emerges.

A particularly valuable capability for the organization is continuous permission monitoring and least-privilege enforcement. Astrix enables the security team to see what access each agent has and identify agents with permissions beyond read-only read-only. "[We] can see what access each agent has and identify agents with more than read-only permissions," the customer highlights. Most alerts relate to agents with unnecessary write access to applications, enabling least-privilege enforcement without manual audits and addressing the common pattern of agents provisioned with broad permissions "just in case" and never right-sized.

Outcomes, Limitations, and Market Reality

Adoption of Astrix for AI agent security was a natural extension of an existing relationship. The organization was already using Astrix for non-human identity security before expanding into AI use cases, creating immediate operational synergy. "[We were] already using Astrix for identity security and non-human identity management before the AI use case," he explains. Two to three engineers were already actively working with Astrix on IAM-related initiatives, and the platform's strong integration with Google Workspace, critical for the organization being a "Google shop", made it a natural fit alongside Okta within the organization's best-of-

breed identity strategy. The customer emphasized that Astrix complements rather than replaces, fitting into their best-of-breed strategy where they use the best tool for each capability rather than pursuing single-vendor consolidation.

Competitive Advantage Emerges Around The Convergence Thesis (Anchoring AI Agent Security in Non-Human Identity Control)

Astrix's unique competitive advantage stems from being one of the few emerging identity platforms that addresses both legacy NHI remediation and agent AI deployment within a single integrated solution. This approach anchors AI agent security in non-human identity control, treating AI agents as enterprise identities whose risk is defined by the access, permissions, and downstream impact they hold across systems.

Underpinning this approach is Astrix's identity-first, threat-driven, and enterprise-ready security philosophy. Astrix views AI agent risk as a function of what agents can access and impact across the enterprise, rather than how models behave in isolation. As a result, the platform emphasizes real access paths, behavioral drift, and downstream impact over static policy assumptions or predefined rules, while operating at enterprise scale across cloud, SaaS, and hybrid environments. This framing positions Astrix to support AI agent adoption without introducing friction, embedding security controls into existing workflows rather than relying on standalone enforcement points.

The broader competitive landscape generally spans two areas: pure-play NHI security specialists that excel at discovering and securing existing machine identities but lack deployment capabilities, and emerging AI security vendors that focus on prompt injection, model security, or data governance but don't address the identity/access layer.

Astrix uniquely spans both areas by enabling organizations to use the Discover–Secure workflow to gain visibility and control over all AI agents, including sanctioned and shadow agents, MCP servers, and the thousands of existing NHIs that power them across enterprise environments, while leveraging the Agent Control Plane (ACP) to ensure new agents are deployed following secure-by-design patterns from day one. The technical integration between Astrix's discovery, security, and deployment (powered by ACP) capabilities is reinforced through the Identity Graph, which allows deployment decisions to be informed by existing identity context and risk signals. When a developer requests credentials via ACP for a new agent, Astrix evaluates the risk context of similar existing agents: Are there other agents accessing this same system? What permission levels do they have? Have any of them been flagged for security findings?

Conversely, when the discovery platform identifies a risky NHI (e.g., a long-lived service account with excessive permissions and no recent activity), Astrix determines whether that identity is actively managed through Agent Control Plane or represents a legacy shadow identity that bypassed secure provisioning, and prioritize remediation accordingly. This bidirectional intelligence flow between discovery, security, and deployment is architecturally impossible for point solutions that only address one side of the lifecycle.

Astrix's Identity Graph is a purpose-built graph database representing entities, such as AI agents, MCP servers, NHIs, secrets, systems, users, along with their relationships (uses, accesses, owns, authenticates-with) as first-class objects with rich metadata. Security is fundamentally about understanding relationships and context, not just cataloging assets. A service account with admin privileges might be high-risk or acceptable depending on what uses it (a sanctioned DevOps pipeline vs. a shadow agent), who owns it (active employee vs. departed contractor), where it's used (production vs. development), and what it accesses (public data vs. PII). The Identity Graph makes these contextual questions first-class queries rather than requiring manual joins across multiple tools. This contextual intelligence powers Astrix's differentiated risk scoring (incorporating deployment posture, usage patterns, and ownership status rather than just static entitlements) and policy learning (identifying patterns like "service accounts created by Team A have 3x more security findings than Team B" and codifying preventive policies).

Identity Becomes the Control Plane for Autonomy (SACR Sees This as a 2026 Category)

Agentic AI doesn't simply introduce a new application risk; it introduces a new class of actors that are agentic systems that execute actions at a scale, speed, and variability humans cannot match. That shift breaks the implicit assumptions behind legacy identity architectures. IAM and SSO were built to identify who is logging in; IGA was built to review access on a human timescale; PAM was built to protect privileged pathways that were rare, explicit, and operationally bounded. Agents invert those assumptions. They operate continuously, chain tools, spawn workflows, and frequently rely on non-human identities and secrets that are fragmented across environments. In that world, "secure access" is no longer a one-time decision at login. It is a continuously enforced state tied to intent, scope, and runtime behaviour.

This is why **SACR is unusually bold in framing Agentic Identity and Access as a major security category heading into 2026**—not as an incremental extension of IAM or PAM, but as a control-plane shift driven by machine-speed autonomy. The market isn't merely adding another point product; it is being forced to re-platform how authority is granted, fulfilled, and revoked when the actor is non-deterministic and operates faster than human supervision.

This report frames the emerging solution as the **Agentic Identity Access Platform (AIAP)**: a practical blueprint for building the new "SSO for Agents" that governs agent access end-to-end. AIAP is not a single feature; it is a control plane that standardizes how agents are onboarded, how they request authority, how credentials are fulfilled, and how trust is withdrawn when execution drifts. The four-phase model captures what "good" looks like operationally:

1. **Discover & Register** the agentic identity graph
2. **Translate & Authorize** by converting intent into deterministic, auditable policy decisions
3. **Broker & Inject** to enforce Zero Standing Privilege through short-lived, task-scoped credential exchange
4. **Watch & Terminate** to detect behavioral drift, build actionable threat cases, and trigger kill switches by revoking ephemeral access in real time.

Together, these phases convert an organization into a governed system where access is attributable, minimal by design, and terminable at machine speed.

The vendor landscape reflects a market still early but rapidly converging. Different approaches emphasize different phases from visibility breadth, intent-to-policy translation, brokered credential injection, or runtime enforcement. The common direction is clear: enterprises are moving toward centralized brokering, deterministic authorization, ephemeral access, and runtime consequence. Over time, the silos between non-human identity, workload identity, and agentic identity will collapse into a unified, dynamic access layer where identity is not just a directory object but a "temporary authority state" granted for a purpose, bounded by policy, and continuously validated through execution.

For IAM and security leaders, the path forward is pragmatic rather than theoretical. Start with ownership and attribution. Clean up identity debt and eliminate hard-coded secrets. Use existing IdP investments as the substrate for human authentication, but build agent governance above it: intent policies, brokered access, and runtime kill switches. The organizations that succeed will not be the ones with the most agents: they'll be the ones that can prove, for every autonomous action, who initiated it, what was intended, what was allowed, what credentials were used, what happened during execution, and how quickly access was withdrawn when things went wrong. That is the new bar for trust in the age of autonomy, and it is exactly why SACR is calling this one of the defining security platform battles of 2026.

There will be many more reports from our firm unpacking this category later in 2026.

Software Analyst®
Cyber Research

business personal

softwareanalyst.io

SACR®

Trusted research. Sharp insights. Real conversation.

CISO	VENDOR
SECURITY TEAMS	INVESTORS