THOUGHT LEADERSHIP

EASY DYNAMICS℠

A Standards-Centric Identity Architecture for Federal AI Agents

How to secure your agentic enterprise now using the platforms you already run.

AI agents act across your systems at machine speed. Governed well, they are a force multiplier; governed poorly, each one is an unmonitored account with broad access and no owner. The difference is identity. This paper sets out the identity architecture that lets an agency adopt agents safely, the open standards it is built on, and how to start now, written for the executives who own the risk and the engineers who will build the controls.

What this paper covers and what it does not.

This paper addresses the identity security of AI agents: how an agent proves what it is, how its access is scoped and kept least-privilege, how that identity and context propagate across the services and agents it calls, and how access is continuously evaluated and revoked. In short, the authentication, authorization, delegation, and attribution layer for non-human, agentic workloads. That layer is where open standards and today's identity platforms give an agency concrete, enforceable control.

It does not cover the adjacent disciplines that also govern safe AI: model behavior and output accuracy, bias and fairness, content safety and prompt-injection prevention, data governance and privacy, model supply-chain security, or AI risk-management frameworks such as the NIST AI Risk Management Framework. Those are addressed elsewhere. Identity is the containment layer for several of them. When an agent is talked into misusing legitimate access through prompt injection, or behaves non-deterministically, least-privilege scoping, per-request authorization, and fast revocation limit the blast radius and preserve attribution, even though the credential and its permissions were technically valid. Preventing the manipulation itself is out of scope here.

AI agents are workloads at scale.

An AI agent carries a non-human identity. It authenticates programmatically, capable of running with no human at the keyboard, calls tools, APIs, and can spawn more agents on its own. This is the same identity problem the industry has engineered against for years with service accounts and workloads, now multiplying at the speed of human ingenuity: Non-human identities already outnumber human identities[1] by more than 100:1. Six risks must be managed, and each carries a direct cost.

The risk (what breaks)	What it costs the mission
Long-lived credentials. Static API keys and secrets that outlive the agent's need for them.	A leaked key can provide access for months, allowing a breach to persist long after the agent is gone.
Over-permissioned agents. Broad standing scopes granted so the agent is useful.	One compromised agent allows an attacker to reach far more than the agent's task ever required.
Confused deputy. A low-privilege caller or injected prompt drives a high-privilege agent.	Users cause actions they could not perform directly, and the log cannot tell who really acted.
Token theft and replay. Tokens handed to downstream services in a multi-agent chain.	A stolen token is replayed to impersonate the agent and its user.
No continuous authorization. Access is decided once, at sign-in.	A rogue or compromised agent keeps working until a token happens to expire.
Weak attribution. Shared accounts and hardcoded keys, with no unique identity per agent.	Actions cannot be traced to an agent or its owner, so audit and incident response fail.

One target: open identity standards for agent workloads.

The failure modes above are not solved by a product. They are solved by a detailed understanding of each agent's intended use and scoping that agent's capabilities correctly as well as designing your solution around a small set of open identity standards that treat an agent as a workload. Three identity ideas carry most of the weight:

1. **Workload identities that are short-lived and secretless.**
2. **Identity and context that travel across every hop of an agent's call chain.**
3. **Continuous, signal-driven authorization that can revoke an agent in near real time.**

That is the target architecture. It is a multi-year destination rather than a switch to flip: some of these standards are mature, others are still emerging, and broad product support, continuous authorization in particular, will be limited for the foreseeable future. The two FedRAMP High platforms most agencies already run, Microsoft Entra and Okta, are both building toward these same standards today. They lead in different places and leave different gaps. The strategy is not to adopt a platform and wait for the standards to arrive later. It is to adopt each platform for the standards it already implements, push it where it leads, and engineer around the gaps that remain. Deploy what is mature now, apply interim controls where it is not, and design so the standards drop in as products catch up. The value: agents you can monitor, scope, and switch off, adopted fast enough to serve the mission without opening a new class of standing risk.

The building blocks and where the platforms already are.

Each block states what the standard solves in plain terms, compares where Okta and Entra stand today with the gap to close, and ends with the move to make and the benefit it delivers. The standards named here are defined in the *Standards in plain terms* section at the end of this paper. The four blocks sit at very different stages of maturity, which shapes what to deploy now and what to design for.

Building block	Where the standard stands	What that means for you
Workload Identity (Secretless)	**Mature.** SPIRE is production-grade and managed identities are generally available.	Deploy now. The safest, highest-impact starting point.
Identity Chaining (ID-JAG)	**Emerging.** Okta implements it; cross-vendor support is thin.	Broker with token exchange now, pilot ID-JAG, and design for cross-domain use later.
Transaction Tokens	**Draft.** No product ships it; only early open source exists.	Design services to accept a signed context token now, so the standard drops in later.
Continuous Authorization (CAEP, Shared Signals)	**Partial.** The specs are final and Okta is ahead, but most apps and Entra's external surface are not CAEP-aware, and will not be soon.	Shorten token lifetimes and use native revocation now; adopt Shared Signals as support arrives.

1. Workload identity: short-lived and secretless.

Replaces long-lived API keys with an identity issued from runtime attestation, scoped and rotated automatically, so there is no stored secret to steal.

Okta	Scoped, short-lived tokens; governs agents still on static credentials. Does not mint workload identities itself.
Entra	Workload identity federation trusts SPIFFE and SPIRE. In Entra Agent ID (generally available May 2026), an agent holds no secret of its own; a parent blueprint supplies a managed-identity credential.
Gap to close	Neither issues SVIDs as the native agent credential; run SPIRE as a layer beneath the platforms.

Do now: Start with the workload identity federation already in your cloud platform, such as Entra workload identity federation or AWS IAM Roles Anywhere, to remove static cloud keys without new infrastructure. Move systems that cannot go secretless yet onto vaulted, automatically rotated secrets as an interim step. Then pilot SPIRE for service-to-service identity where you span multiple clouds or need attested identity beneath both platforms.

Design for: SPIRE as the target for multi-cloud estates, budgeting for the cost of operating it inside an accreditation boundary. Expect cloud-native workloads to reach secretless in FY27, and legacy systems to need vaulted, rotated secrets well beyond it.

Benefit: A leaked credential is worthless within minutes, and static cloud keys stop being an audit finding.

2. Identity chaining across apps and agents.

Carries a user's and agent's authorization context across app and domain boundaries, checked against policy at each hop instead of riding on a static key or a consent click.

Okta	Cross App Access implements the IETF ID-JAG grant, now a stable MCP authorization extension as of June 2026 and vendor-neutral, so any IdP can speak it. Okta was the first identity provider to ship it, alongside a first wave of MCP servers.
Entra	On-behalf-of exchange for delegation, but within a single tenant; agent identities are single-tenant. No cross-domain chaining grant yet.
Gap to close	Cross-domain chaining is Okta-led; Entra's on-behalf-of is based on the OAuth JWT-bearer flow but stays in-tenant. Both support MCP and A2A. Chaining also only works if the receiving application accepts it, so custom apps must be updated to act as resource servers.

Do now: Broker agent-to-app and app-to-app access through the IdP with OAuth token exchange and on-behalf-of within a tenant, and never hand a raw downstream token to an agent.

Design for: Cross-domain identity chaining (ID-JAG) as the grant and product support mature. Cross App Access is still Early Access and enabled through Okta support rather than self-service, so plan pilots accordingly. Also budget the application-side work to make your own custom apps accept these tokens: That is engineering effort on each application, not identity configuration, and it is usually the long pole in a federal estate.

Benefit: Every agent-to-app connection is visible, policy-checked, and revocable, with no static keys or consent-screen sprawl.

3. Transaction tokens for the internal call chain.

Inside the trust boundary, pins the user, agent, and authorization context to a single request as it fans out across internal services, so a stolen token cannot be replayed or used out of context.

Okta	OAuth token exchange, which the transaction-token profile builds on. No transaction-token feature yet.
Entra	On-behalf-of exchange for delegation, not the standard token exchange the transaction-token profile builds on. No transaction-token feature yet.
Gap to close	The one block neither platform ships. An active IETF draft, with an agent extension adding agent and principal fields.

Do now: Standardize internal calls on token exchange where a platform supports it, and design each service to trust a signed context-bearing token.

Design for: Standards-based transaction tokens once a service ships, dropping into the context-token pattern with little rework.

Benefit: An attacker who steals one token inside the boundary cannot replay it or reach services out of context.

4. Continuous authorization and revocation.

Replaces decide-once-at-sign-in with event-driven signals, so any enforcement point can revoke a rogue or compromised agent in near real time.

Okta	Full Shared Signals transmitter and receiver, the most complete of any commercial IdP; continuous session monitoring and automated response.
Entra	Continuous Access Evaluation (CAEP-based) revokes within the M365 boundary, and Agent 365 can disable a blueprint or agent to block new tokens. But it is an internal loop, its coverage for agent identities is unconfirmed, and outside revocation uses the Graph API, which does not stop already-issued tokens.
Gap to close	Standards-based Shared Signals reaching into and out of Entra; use a Shared Signals hub and bridge Entra.

Do now: Shorten token lifetimes to bound the exposure window, use each platform's native revocation (Entra via Graph, Okta via session revocation) and Continuous Access Evaluation where it already works, own the signal bus that calls those revocation APIs, and measure and shrink time-to-revoke a compromised agent.

Design for: Standards-based Shared Signals into and out of every platform, so the same signal bus revokes in seconds once transmitters and receivers are broadly available.

Benefit: Short lifetimes and native revocation contain a compromised agent in minutes to about an hour today, and Shared Signals gets that to seconds as support arrives.

Authorize the action, not just the agent.

The four blocks above establish who an agent is and whether its access should continue. They do not decide whether a particular action should be allowed. Revocation answers whether you can cut an agent off; it does not determine, at the moment of the request, that this agent may delete this record or release this payment. The standards draw the same line: the stable MCP authorization extension governs which tools an agent may connect to, not what it does once connected.

Close that gap with per-request authorization. NIST SP 800-207 describes the pattern as a policy decision point paired with policy enforcement points: every agent tool call is evaluated against policy before it executes. Express those rules as code so they are versioned, tested, and auditable rather than scattered through application logic, and require human approval for a short list of high-impact actions. Unlike the standards above, this is available today with mature open-source policy engines, which makes it the strongest addition you can make in the near term.

Do now: Put a policy decision point in front of agent tool calls, keep the rules in version control, and define the high-impact actions that require a person to approve them.

Benefit: An agent that has been manipulated or has drifted from its task is stopped at the action it attempts, even when its credential and permissions are technically valid.

How this maps to federal policy.

Each control traces to policy an agency already answers to, which makes this an extension of existing zero trust work rather than a new mandate.

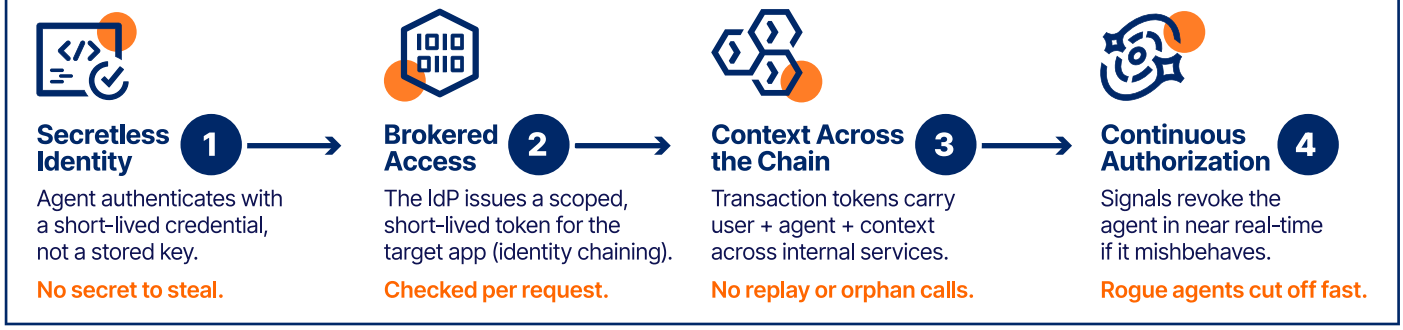
Control	What requires it
Workload Identity (Secretless)	OMB M-22-09 least-privilege, phishing-resistant identity for every entity; NIST SP 800-63-4 for credentials; CISA Zero Trust Maturity Model, identity pillar
Identity Chaining	NIST SP 800-207: verify every request, including across trust boundaries, rather than trusting a static key
Transaction Tokens	NIST SP 800-207: authorize per request rather than per session, with context preserved
Continuous Authorization	OMB M-22-09 continuous verification; CISA Zero Trust Maturity Model, optimal identity stage
Per-Request Authorization	NIST SP 800-207 policy decision and policy enforcement points
Governed Intake & Ownership	OMB M-19-17 full identity lifecycle management; OMB AI governance memoranda on agency AI inventory and accountability

How it works end-to-end.

These are not four separate projects. They form a single chain that protects one agent request from start to finish: the agent proves who it is without a stored key, the identity provider brokers scoped access to the next app, a transaction token carries the full context through the internal services, and continuous authorization stands ready to cut the agent off the moment it misbehaves.

One request may cross several agents. When an agent hands work to another agent, each hop should add to the record of who is acting rather than replace it, so the chain still shows the person who initiated the work and every agent that touched it. The standards support this: token exchange carries a nested actor claim, and a transaction token keeps the user and agent fixed for the life of the transaction. In practice, cap how deep delegation may go, give every sub-agent its own identity instead of reusing its parent's, and verify that your logs can reconstruct the full chain after three hops. Without those, attribution collapses precisely where you need it most.

How identity travels through one agent request



How to move now.

The most durable move is organizational, not just technical. Agent identity should not be solved app by app or team by team. It belongs to your enterprise identity service, the shared capability that already governs human and machine identity, extended with a governed front door for agents. Mobilize that service first; the technical work below is what it runs. Where you run Microsoft, Agent 365 (generally available May 2026) already provides a native agent registry and controls to build this front door on; the shared service's job is to make governance consistent across every platform, not Microsoft alone.

Mobilize your identity shared service.

Give it to your identity service. Make agent and tool identity an explicit responsibility of your enterprise identity (ICAM) shared service, not a task each program improvises. This shifts a fast-growing risk to the team already equipped to govern identity at scale.

Stand up an agent onboarding intake. Create a single governed front door where any team registers a new agent or tool and receives, in return, a provisioned secure identity, least-privilege access, a named owner, and monitoring. Make it the only supported path to production. Where a platform offers this natively, such as Microsoft Agent 365, use it and hold it to the same bar rather than rebuilding it.

Turn onboarding rules into policy. Define what an agent must have to pass intake: a unique identity and a human owner, scoped permissions, a secretless credential, and a revocation path. The intake enforces these, so governance happens by default rather than by audit. Have the intake emit its decisions as machine-readable control evidence, in OSCAL where you already use it, so the same registration that governs an agent also produces the artifacts an assessor needs.

Charter and fund it as a shared service. Resource the intake as a product with an owner, a roadmap, and service levels, funded so it serves every program rather than competing for project budget.

Publish a paved road and measure adoption. Give teams self-service templates and clear guidance so the governed path is the easy path, and track agents onboarded through the service, time to onboard, and time to revoke a compromised agent.

The technical moves the service runs.

Run these in parallel; the intake provisions and enforces them for every agent it onboards. Centralize enforcement at gateways, the service mesh, or a policy decision point, so emerging standards plug in without re-architecting.

See and own every agent. Inventory across Entra and Okta, give each agent a unique identity and a named human owner, and retire orphaned accounts and hardcoded keys. You can finally answer which agents exist and who is accountable.

Go secretless. Replace static API keys with managed identities or federated credentials in Entra and scoped short-lived tokens in Okta, targeting zero static secrets in agent configuration.

Broker access, do not embed it. Route agent-to-app and app-to-app access through the identity provider using identity chaining, so every hop is checked against policy and can be revoked centrally.

Contain compromised agents. Shorten token lifetimes, use native revocation and Continuous Access Evaluation where it works, and own the signal bus that calls those revocation APIs, tracking time-to-revoke as a measured objective.

Pilot workload identity. Run SPIRE for service-to-service identity and federate it to your cloud IAM to remove static cloud keys, positioning for SVID-based credentials as the platforms adopt them.

Buy for the standards. Require support for the standards that exist today, such as token exchange and SPIFFE federation, plus dated roadmaps and written commitments to the emerging ones (Shared Signals, ID-JAG, and transaction tokens), so you converge on open standards and avoid lock-in for capabilities a standard will soon cover.

> **Design for and track to deployment:** Cross-domain identity chaining, standards-based Shared Signals across every platform, transaction tokens, and native SVID issuance. Build the seams now so each drops in as products ship.

How Easy Dynamics helps.

Easy Dynamics designs, builds, and secures the identity systems that protect federal missions. We help agencies mobilize their identity shared service and stand up a governed agent intake, adopting these standards through the platforms they already run: getting each agent to a secretless identity, brokering agent access through identity chaining, wiring continuous authorization across the estate, and closing the gaps that remain, from transaction tokens to cross-vendor Shared Signals to SVID-based workload identity. If your agents are already running, the work starts with seeing and owning them.

To scope an agent-identity roadmap for your agency, contact the Easy Dynamics Digital Identity and Cybersecurity practice at **info@easydynamics.com.**

Standards in plain terms.

Below is a quick reference to the open standards named in this paper, and what each is for.

SPIFFE (Secure Production Identity Framework for Everyone): *An open standard that gives every workload a verifiable identity, so software can prove what it is without a shared password or API key.*

SPIRE (the SPIFFE Runtime Environment): *The open-source software that implements SPIFFE, checking what a workload is and issuing it a short-lived identity credential.*

SVID (SPIFFE Verifiable Identity Document): *The short-lived credential SPIFFE issues, a certificate or signed token that carries a workload's identity and expires on its own.*

OAuth 2.0 Token Exchange (RFC 8693): *A standard that lets a service trade one token for a narrower, scoped-down token, so each hop passes along only the access it needs.*

On-Behalf-Of (OBO): *Microsoft Entra's flow for one service to call another while carrying the original user's identity, within a single tenant.*

Identity chaining, or ID-JAG (Identity Assertion JWT Authorization Grant): *An emerging IETF standard that lets an agent carry a user's identity from one application to another, even across organizations, as a signed and scoped assertion the receiving app can verify, rather than a static key or a blanket consent.*

Model Context Protocol (MCP): *An open standard for how agents connect to tools and data sources; its authorization extension defines how an agent is granted access to those tools.*

Agent-to-Agent (A2A): *An open protocol for one agent to discover and delegate work to another agent.*

Transaction Tokens: *An IETF draft standard for a short-lived, signed token that pins the user, the agent, and the request context to a single transaction as it moves across internal services, so a stolen token cannot be replayed or reused for a different action.*

Continuous Access Evaluation Profile (CAEP): *An OpenID standard that defines security events, such as a revoked session or increased risk, so access can be re-checked continuously rather than only at sign-in.*

Shared Signals Framework (SSF): *The OpenID standard that carries those events between systems, so one vendor's signal can trigger action in another vendor's system in near real time.*

IdP (identity provider): *The system that authenticates users, agents, and services and issues their tokens, such as Microsoft Entra or Okta.*

ICAM (Identity, Credential, and Access Management): *The federal term for the enterprise capability that governs who and what can access systems.*

---

[1] *Platform capabilities and standards status reflect information available as of mid-2026 and continue to evolve. Identity chaining, transaction tokens, and SPIFFE client authentication are advancing through the IETF; Shared Signals and CAEP are finalized OpenID Foundation specifications. Microsoft Agent 365 and Entra Agent ID reached general availability in May 2026. Broad product support for several of these standards, continuous authorization in particular, will be limited for the foreseeable future; treat this as a multi-year target and re-assess as support matures.*