Nick Clark · Published July 29, 2026 · PDF

Okta Auth for GenAI: Agent Identity and Origin Equivalence

An agent that meters refusals per assertion can be pushed into withholding by one adversary presenting many fresh identities, because the cost of another identity is lower than the cost of the refusal it buys. Okta Auth for GenAI, as publicly described, addresses the adjacent problem of giving AI applications and agents a governed way to authenticate and to act on a user's behalf. U.S. Provisional Application No. 64/117,812 discloses something structurally different: the agent derives origin-equivalence classes from its own append-only lineage field and its own counterparty identity records, meters per source rather than per assertion, and treats the resulting class identifier as strictly local. This article describes both, fairly, and shows where they sit relative to each other.

When one adversary arrives as forty callers

An agent that can refuse has to count refusals, or a single counterparty wears it down by repetition. So it keeps a counter, sets a threshold, and withholds action when the rate of refusals in a window suggests its own picture of the world is unreliable.

That design has a cheap failure mode in an open network. The filed specification names the condition directly: increments accumulate on both refusal paths, the rejected determination (124) produced for a false conduct evaluation artifact (116) and the not-

determinable determination (126) produced for an artifact unresolvable against the append-only lineage field (104). Volume alone therefore drives the authorization gate (300) into the withheld state (310), and, in the words of the filing, the volume available to an adverse party is limited only by the cost of presenting further identities.

Where presenting a further identity is cheap, that bound is weak. A counter charging per assertion charges once for each artifact, so one origin presenting many identities that look mutually unrelated accumulates one increment apiece. The artifacts need not be sophisticated, only numerous and unresolvable against the lineage field. The filing works this through, by way of illustration, at forty asserting parties sharing no declared relation type.

One instinct is to fix this at the identity layer: make the callers prove who they are, and the duplicates collapse. That is why the identity-platform category is the right thing to describe first.

The problems an identity platform is built to solve

Okta is, as publicly described, an identity and access management platform. Its core business is authenticating principals, issuing and validating tokens, applying access policy, and giving an organization one governed place to manage who and what may reach a resource. That is standards-centered work, and it is hard work.

Okta Auth for GenAI, as publicly described, extends that platform toward AI applications and agents. Its stated concerns are the ones teams shipping agentic software run into: an AI application needs to authenticate the user in front of it; an agent needs to call third-party APIs on that user's behalf without the application hoarding raw credentials; some agent actions should not proceed without a human in the loop, including when the agent runs asynchronously and the human is not at the

keyboard; and authorization decisions often need to be finer-grained than a single role check, particularly where retrieval touches documents the user may not be entitled to see.

Those are well-scoped problems and the category answer is a good one. Delegated authorization is the right primitive for an agent acting for a principal, and centralizing token handling under governance beats scattering secrets through application code. Nothing below argues that any of it is insufficient. The filed subject matter answers a different question.

Deriving an origin class from the agent's own records

An identity platform answers **who is this**. The filed mechanism starts from a question its own metering forces on it: **how many sources of pressure is the agent facing**.

Under the specification, the refusal counter (304) of the semantic agent (100) is incremented per origin-equivalence class (200) rather than per conduct evaluation artifact (116). A source is a set of asserting parties (118) between which the agent demonstrates a relation from its own records.

Assignment follows an ordered procedure. On receipt of an artifact, the agent retrieves the counterparty identity record (114) of the asserting party, or instantiates one where none exists. It then retrieves from the signed policy object (112) in force an enumeration of declared relation types, each specifying a class of lineage entry and a matching condition over such entries. For every party already assigned to a class, in the current window or a preceding one, it evaluates each declared relation type against the present party.

The filing gives three declared relation types by name:

- A **shared dispatch lineage (202)**, evidenced where the lineage field holds an entry recording a dispatch to the present asserting party and an entry recording a dispatch to the compared asserting party, both recording a common parent dispatch entry as their immediate antecedent.
- A **co-signature (204)**, evidenced where a single lineage entry bears a signature verifiable against an identity primitive of the present asserting party and a signature verifiable against an identity primitive of the compared party.
- A **common introduction path (206)**, evidenced where both counterparty identity records record an introducing party and the recorded introducing parties are identical.

Where any declared relation type is evidenced, the present party joins the compared party's class; where more than one class is identified, the classes merge; where none is evidenced, the party gets a new class. The assignment, the relation types evaluated, the entries relied upon, and the resulting class identifier are appended to the lineage field, and the identifier is written into each member's counterparty identity record. That identifier records the class to which the party is assigned and is not a determination concerning that party's conduct.

Each declared relation type is independently sufficient. The evaluation runs on the agent's own lineage entries and counterparty identity records, without reference to a centralized registry, without query to a directory, without participation in a consensus procedure, and without coordination with a further execution node.

Two properties matter for the comparison. First, the class is local. A second semantic agent holding a different lineage field and different counterparty identity records derives, from the same population, a partition that need not agree, and no procedure reconciles the two: no class identifier is transmitted between them, and neither agent admits a class identifier derived by the other. The class identifier is therefore not an identity attested by a third party and confers no portable standing. Second, normalization governs metering alone. An artifact from a party in a class already

recorded in the per-class increment register still produces a determination and is appended to the lineage field; it applies no increment. The filing states that normalization suppresses no determination and withholds no adjudication.

The filing also addresses identities the agent has never met, where no relation is evidenced. Under the severance-survival test, each constituent edge of a class resolves to a reason-type, and a class all of whose edges resolve not-typeable is designated an **untested class**. Where such a class is designated and the recorded introduction paths of its members converge upon a common ancestor entry, within a traversal depth declared in the signed policy object, a cost multiplier declared in that policy object is applied to the class's contribution. The multiplier is greater than zero and less than unity. Both conditions are required: a class designated untested whose introduction paths do not converge contributes at full weight. The multiplier, the rate threshold, the depth, and the continuation interval are policy-declared, and the specification states no values for them outside a worked trace given by way of illustration.

Attested identity versus local partition

An identity platform establishes standing that travels; the filed class is defined so that it cannot travel. Portability is the point of federated identity: an assertion from a trusted issuer is meant to be presentable, verifiable by a party that never met the subject, and consistent across relying parties.

Origin-equivalence normalization rests on the opposite commitment. The class is computed by one agent from that agent's own records, is never transmitted, is never admitted from another agent, and confers no standing on the party it describes. The specification distinguishes this from a Sybil-resistance scheme that resolves identity against a shared registry or a coordinating authority, and states that no shared authority is required or consulted.

The positioning is complementary rather than competing. One layer establishes standing a relying party can verify; the other derives a partition that, by construction, no party but the deriving agent can use. The filed procedure requires nothing of the caller and nothing of a third party because its inputs are limited to entries the agent already holds.

Layering the two, and the limits the filing states

A deployment can carry both layers. The identity layer authenticates the human, obtains delegated authorization for the agent to act, handles tokens for downstream calls, and applies policy at the boundary of an administered domain. The agent-local layer, as disclosed, operates only on what it already holds: the declared relation types are evaluated over entries of the append-only lineage field (104) and over counterparty identity records (114), and the specification states the evaluation proceeds without reference to a centralized registry, without query to a directory, without participation in a consensus procedure, and without coordination with a further execution node. Whatever a boundary establishes about a caller therefore reaches the metering only insofar as it is recorded as such an entry. The filing describes no interface for importing it, and the comparison here should not be read as one.

Several limits of the disclosed architecture follow from that scope, and none of them are gaps it sets out to close:

- It does not authenticate anyone. There is no credential issuance, token validation, session management, or login flow in the mechanism.
- It does not produce a reputation or trust level. The class identifier is not a determination concerning conduct.
- It gives the counterparty nothing usable. A party cannot present its class elsewhere or carry it to a second agent.

- It does not eliminate cost from unrelated new callers. Where no declared relation type is evidenced, each party forms its own class, and the multiplier reduces that class's contribution only where the class is designated untested and its introduction paths converge.
- It states no values. The rate threshold, the multiplier, the introduction-path depth, and the continuation interval are declared in the signed policy object, and a deployment must set them.

Stated structurally: an identity platform establishes and conveys standing about a caller, while the filed mechanism derives, locally and without conveying anything, how much a caller's refusals weigh.

Disclosure Scope

This article describes subject matter disclosed in U.S. Provisional Application No. 64/117,812, a pending application. Statements about the disclosed architecture are drawn from that filing and use its own mechanism names and outcome terms. Nothing here expands its scope or should be read as a claim construction.

References to Okta Auth for GenAI are to public materials and are used for comparison only; no relationship, endorsement, or infringement is asserted.

Descriptions of the named subject are qualitative and reflect its publicly documented purpose and category. Product capabilities change, and readers evaluating a deployment should consult the vendor's current documentation rather than this comparison. Statements about what the disclosed architecture requires are statements about the filing, not an audit of any other product.

Origin-Equivalence Normalization (/origin-equivalence)

All 49 steps → (/inventive-steps)

Metering that can't be gamed by splitting one accuser into many.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

- Origin-Equivalence Normalization: Metering Refusal Per Source, Not Per Assertion (/articles/origin-equivalence/origin-equivalence-normalization)

SECONDARY TECHNICAL

- Content-Blind Encounter Commitment and the Paired Encounter Receipt (/articles/origin-equivalence/content-blind-encounter-commitment)
- Cross-Receipt Equivocation Detection: Catching an Agent That Tells Two Stories (/articles/origin-equivalence/cross-receipt-equivocation-detection)
- The Untested-Origin Saturation Bound: Pricing a Flood of Freshly Minted Identities (/articles/origin-equivalence/untested-origin-weighted-saturation-bound)
- Counterparty-Side Fork Detection: Recognizing a Split Identity From Outside (/articles/origin-equivalence/counterparty-side-fork-detection)
- The Attestation Origin-Equivalence Gate: An Anti-Collusion Bound (/articles/origin-equivalence/attestation-anti-collusion-bound)
- The Bridging Party and the Deferred Merge: Charging One Windowed Increment Against Every Origin-Equivalence Class a Single Asserting Party Bridges (/articles/origin-equivalence/bridging-party-deferred-merge)
- The Common-Execution-Node Relation Type: Charging Co-Hosted Asserting Parties as One Origin (/articles/origin-equivalence/common-execution-node-relation-type)
- Receiver-Side First-Encounter Budget Decrement: Pricing a Stranger's Arrival Per Presented-Material Origin Class (/articles/origin-equivalence/receiver-side-first-encounter-budget-decrement)
- Mutual First Encounter Without Cross-Budget Coordination: Two Strangers, Two Matched Pairs, Two Separate Budget Decrements (/articles/origin-equivalence/mutual-first-encounter-no-cross-budget)

APPLICATIONS · GENERAL

- Sybil-Resistant Agent Networks: When One Accuser Becomes Many (/articles/origin-equivalence/sybil-resistant-agent-networks)

- Fraud Rings in Autonomous Commerce: Metering Per Source, Not Per Claim (/articles/origin-equivalence/fraud-rings-in-autonomous-commerce)

APPLICATIONS · SPECIFIC

- W3C Decentralized Identifiers and Verifiable Credentials: Verifying an Identifier Versus Metering Its Origin (/articles/origin-equivalence/w3c-decentralized-identifiers)
- **Okta Auth for GenAI: Agent Identity and Origin Equivalence (/articles/origin-equivalence/okta-auth-for-genai)**
- WorkOS: Agent Identity Infrastructure and Source Metering (/articles/origin-equivalence/workos-agent-identity)
- Entra Agent ID: Directory Identity and Derived Origin Classes (/articles/origin-equivalence/microsoft-entra-agent-id)
- SPIFFE and SPIRE: Workload Identity and Origin Equivalence (/articles/origin-equivalence/spiffe-spire)
- Cloudflare Bot Management: Blocking Agents and Metering Sources (/articles/origin-equivalence/cloudflare-bot-management)

TERMINOLOGY

- asserting party (/glossary#asserting-party)
- origin-equivalence class (/glossary#origin-equivalence-class)

Origin-Equivalence Normalization overview → (/origin-equivalence)