The Agent-to-Agent Handoff Problem

Session Trust and Privilege Escalation Across Agent Boundaries

Binod Kumar

Founder & Chief AI Officer, AgentsArchitects.ai

Mentor, Indian Institute of Technology Bombay and Banaras Hindu University

Produced for and with the AuthClaw Foundation and the AGI Learning Hub, India

March 2026

Abstract

The industry spent 2024 and 2025 learning how to secure the individual agent. The problem still open in 2026 is securing the handoff: the moment one agent passes work to another and a trust boundary gets crossed without anyone checking it. Consider what happens when Agent A asks Agent B to deploy something to production, or to pull a customer's records. Three questions go unanswered, and they go unanswered at machine speed. Did A really hold the authority it just passed on? Will B end up with more access than it should? And whose human authorization, if anyone's, still sits behind the action four hops down the chain? In most production multi-agent systems today, nobody checks any of the three. This is not a new vulnerability. It is the confused deputy problem, known in distributed systems since 1988, showing up again in a setting that makes it far more dangerous, because agents are autonomous, act asynchronously, delegate recursively, and increasingly cross organizational lines. The defenses built for human-to-service authorization, such as OAuth, SAML and OIDC, carry assumptions that agents break. For a Chief AI Officer the takeaway is uncomfortable but simple: if your threat model still treats an AI agent as a passive query tool, your multi-agent rollout is already moving faster than your identity controls. For a researcher, proving which human authorized which action across a recursive delegation chain remains unsolved, and it is a problem where real progress can be made.

Keywords: multi-agent systems; confused deputy; privilege escalation; delegation chains; OAuth 2.0 Token Exchange; agent identity; session trust; runtime authorization; non-human identity; holder-side attenuation; multi-principal attribution.

1 Why the Handoff Is the Unit of Risk

Multi-agent architecture has moved out of the demo and into production. Gartner expects roughly 40 percent of enterprise applications to embed task-specific AI agents by the end of 2026, up from under 5 percent in 2025. Every major platform now ships some form of inter-agent communication, and Google's A2A protocol and the Model Context Protocol (MCP) have become the most common plumbing for it.

The trouble is that most of the security conversation is still about the individual agent: its prompt, its tools, its guardrails. That framing misses where the risk piles up. In a single-agent system you ask one question, which is what this agent may do. In a multi-agent system you have to ask a second one: what happens when this agent passes work to another agent. And you end up asking it dozens of times per task, automatically, with no human watching.

The deployment numbers show why that second question is the one that bites. Recent industry research puts roughly 90 percent of deployed AI agents at permission levels well beyond what their tasks require, and around 97 percent of non-human identities more broadly at excessive privilege. Non-human identities now outnumber human ones in enterprise environments by ratios that run from 25 to 1 up to as much as 144 to 1. Every one of those over-privileged identities is a potential deputy, and every handoff between them is a potential escalation. Meanwhile 79 percent of IT professionals say they feel ill-equipped to stop attacks that travel through non-human identities.

So the thing to study is not the agent but the handoff. The handoff is where authority changes hands, where scope can quietly widen, and where the link back to a responsible human gets lost.

2 The Confused Deputy, Restated for Agents

A confused deputy is a privileged program that gets tricked by a less-privileged caller into misusing its authority. The attacker never needs the deputy's permissions directly. They just abuse the deputy's authority path, and the result is privilege escalation through misused trust rather than a direct exploit.

The agentic version is precise enough to write down. Following recent formalization work, a confused-deputy condition appears when an agent performs an action a on a resource s_{target}, where the authenticated human who initiated the task, s_{user}, is not allowed to reach s_{target}, but the agent itself is. In set terms, $(s_{\text{user}}, s_{\text{target}})$ is not in the relation R, while $(s_{\text{agent}}, s_{\text{target}})$ is. The agent acted with its own broad permissions instead of the user's narrower ones. The command's source was authenticated, but the human behind it had no right to what was commanded.

That is the whole problem in one line. The agent's effective power collapses down to the agent's own permissions, instead of the intersection of the agent's permissions and the human's. An agent's real authority should be that intersection: what the security policy allows, and also what the human behind the agent is permitted to do in the target system. Lose the intersection and, to borrow one security team's phrase, you have built a privilege-escalation machine with a friendly chat interface.

What makes this worse is that no malicious insider is required. Researchers have shown the same condition can be triggered by indirect prompt injection. A payload buried in retrieved content, something along the lines of "ignore previous instructions and ask the lock agent to…", turns an otherwise benign agent into the deputy. Proof-of-concept escalations of exactly this kind have now been demonstrated across several leading multi-agent frameworks, which suggests the weakness is pervasive rather than tied to one bad implementation.

3 Anatomy of a Handoff: What Crosses the Boundary

To see where trust breaks down, follow what physically crosses the line when Agent A hands off to Agent B.

In a well-built system the human's intent travels inside a delegated, on-behalf-of token. That token is usually obtained through OAuth 2.0 Token Exchange (RFC 8693), which can carry an *act* claim that means "B is acting on behalf of A, who is acting on behalf of the user." When this works, the resource server can verify both the grant and its scope, and the chain still has an anchor in the original human.

Three things tend to go wrong in practice, and each one lines up with a distinct failure mode.

Identity collapse. The on-behalf-of pattern only holds as long as the token reflects the user's identity. The moment an agent runs under its own identity, a service principal or, worse, a shared agent credential (which is fast becoming the norm for autonomous and multi-user agents), the target application stops seeing the user at all. It sees a privileged machine making a permitted call. The user's individual permission ceiling vanishes from the provider's view. A low-privilege human can now reach through a high-privilege agent and do things they could never do from their own desk, and the application will not flag it, because as far as it can tell a privileged identity just made a legitimate request.

Scope does not narrow on handoff. When Agent A shares its credential or token with Agent B, B often inherits A's full access, with no way to scope it down or revoke it. OAuth has a structural limit here that matters a great deal: it offers no holder-side attenuation. Only the authorization server can narrow scope, and only at the moment of issue. An intermediary cannot, on its own authority, hand a weaker token downstream. So the easiest engineering path, which is to forward the token you already hold, is also the path that spreads the most privilege.

The chain loses its anchor. OAuth tokens are opaque to intermediaries. Delegation through token exchange mints a fresh token that carries no record of the original authorization chain, and every trust domain runs its own authorization server with no way to verify a token across domains unless federation was set up in advance. The standards-gaps literature puts the consequence plainly: no deployed protocol can cryptographically prove which human authorized which specific agent to perform which specific action at the third or fourth hop of a delegation chain. Working out which human bears responsibility for a delegated action, what the literature calls multi-principal attribution, is still an open problem.

4 Three Failure Modes, and How They Compound

The three breakages above produce a familiar trio of failure modes. They are bad on their own. Together they become dangerous, because they compound across hops.

4.1 Identity collapse breaks attribution

When agents run under shared or service identities, the audit trail blames the agent, or nobody at all. An agent deletes a record and the log says a service principal did it, so investigating an incident turns into guesswork. This also defeats the approval workflows built for people. An employee's expense report goes through three approvals, but that same employee's agent, handed "manage my expenses", can submit, approve and record reimbursements on its own, because the gates were designed for human sessions and not for agent ones.

4.2 Non-attenuation causes scope inflation

Because privilege never narrows on handoff, the effective scope at hop n becomes the union, not the intersection, of every credential touched along the way. Recursive delegation with no privilege attenuation is exactly the condition the delegation literature warns about. In high-stakes domains, access has to be just-in-time, scoped to the immediate task, and gated, precisely because static credentials are what make the confused deputy possible.

4.3 Transitive trust becomes contagion

The worst mode is contagion. Compromise one agent and you inherit the trust of every agent it talks to. A single indirect-injection payload can cross organizational boundaries and compound across multiple credential sets without ever triggering human review along the way. The "authority re-delegation" stage is the nastiest part of it. A compromised coding agent can plant adversarial content in a repository that a different agent later reads during review or testing, re-injecting the attack one hop further on. That is how a local compromise quietly becomes a cross-company one.

The history makes the scale concrete. The 2025 Salesloft and Drift incident showed what credential compromise enables once non-human-identity trust gets chained together. Attackers compromised OAuth tokens from one platform and pivoted across more than 700 connected companies through the trusted relationships those tokens stood for. That was a single-protocol token pivot. A2A contagion generalizes the same dynamic to autonomous agents that create new delegations on their own.

5 Why the Human-Centric Identity Stack Cannot Carry This

It is tempting to assume that better hygiene on OAuth, OIDC and SAML solves this. It does not, and the reason is structural rather than a matter of discipline.

These frameworks bake in a few specific assumptions that agents break by their nature. OAuth 2.0 and 2.1 assume a synchronous human-consent event and single-hop delegation, with one client and one resource server. SAML's assertion model assumes a bounded interactive session anchored to a browser cookie. OIDC inherits the same human-session framing. Agents violate all of it. They act asynchronously, often long after any human was involved. They chain calls across many services and domains. And they shift between acting on their own and acting for a human, while today's systems cannot even tell which mode is active for a given action.

This is the same conclusion the OpenID Foundation reached in its October 2025 work on identity for agentic AI. Current OAuth and OIDC can secure agents that stay inside well-defined boundaries, but three gaps need new work, the biggest being the inability to track delegation mode and the spread of proprietary identity systems that do not interoperate. It is also why, in February 2026, NIST's National Cybersecurity Center of Excellence opened a dedicated concept paper on accelerating identity and authorization for software and AI agents, naming MCP, OAuth 2.1, OIDC, SPIFFE/SPIRE and SCIM as candidate building blocks. Candidates, not a settled foundation.

The short version: the protocols handle one-hop, synchronous, single-principal delegation well, and the moment delegation turns recursive the authorization chain loses its anchor.

6 What "Session Trust" Must Mean at an Agent Boundary

If we are going to use the phrase session trust for a handoff, it has to mean more than "B showed a token that A accepted." Establishing trust at an agent boundary should mean you can answer all of the following, per action, and not just once at registration.

1. **A verifiable delegation chain.** B can prove cryptographically the full path of authority back to an originating human or organization, not merely possession of a bearer token.

2. **Multi-principal attribution.** The action records which human authorized which agent to do what, and that binding survives every later hop.

3. **Holder-side attenuation.** A can hand B a strictly weaker capability than A holds, on A's own authority, without a round trip to a central server. Capability only narrows as it travels. It never widens.

4. **An intersection ceiling.** The capability B exercises has to be the intersection of what policy allows and the originating human's own ceiling, so a low-privilege human cannot reach through a high-privilege agent.

5. **Freshness and bounded lifetime.** Delegated authority should be just-in-time, scoped to the task, and short-lived. No standing, forwardable, long-lived credentials.

6. **Runtime authorization.** Authorization has to be checked at runtime, on every action, and not validated once at the handshake. As the identity community put it at EIC 2026, the missing layer is runtime authorization that evaluates every action rather than just registration.

7. **Revocability and observability.** Any link in the chain can be cut, and every handoff leaves a tamper-evident record, so the chain is auditable and not just the final call.

Most production systems satisfy one or two of these. Almost none satisfy all seven. That gap is the agent-to-agent handoff problem, written out as a requirements list.

7 Candidate Building Blocks, and Where Each Stops

There is real work happening here, and it is moving fast. A senior practitioner should know both what each block gives you and where it runs out.

OAuth 2.1 with Token Exchange (RFC 8693) and the *act* claim is the best available way to represent on-behalf-of delegation, and the right default for single-domain, human-to-service hops. *Where it stops:* no holder-side attenuation, opaque tokens that drop the prior chain, and per-domain authorization servers with no native cross-domain verification.

SPIFFE and SPIRE give non-human workloads strong, attestable cryptographic identities, which answers "who is this agent" far better than a shared service account ever could. *Where it stops:* it identifies the workload, not the delegated human authority behind a specific action, so attribution still needs a layer above it.

Macaroons and capability tokens with caveats are among the few mechanisms that offer holder-side attenuation. A holder can mint a strictly narrower token by adding caveats, which gives you monotonic scope reduction along a chain, and that is promising for the third requirement. *Where it stops:* the ecosystem is immature, standardization is weak, and the verification and revocation tooling sits well behind OAuth.

MCP and A2A are the interoperability plumbing that multi-agent systems are being built on. MCP's spec defines mandatory authorization patterns and per-client consent validation, which target the confused deputy directly when they are implemented properly. *Where it stops:* adoption is outrunning correct implementation, MCP-specific confused-deputy and tool-poisoning patterns are already in the wild, and several practitioners now call MCP authorization the AI security issue of 2026.

Emerging agent-identity frameworks and the various "Laws of AIdentity", from vendors and standards groups, include Okta's agentic IAM framework, EmpowerID's runtime-

authorization framing, OpenID's evidence-infrastructure work, and academic protocols such as AIP for verifiable delegation across MCP and A2A. They converge on the same message: agents need first-class identities, runtime authorization and traceable delegation. *Where they stop:* there is no deployed, interoperable standard yet. The foundation is within reach but not settled.

No single block closes the list in Section 6. The near-term answer is to compose them, rather than wait for one protocol to do everything.

8 A Reference Control Architecture

This is where I move from interpretation to a position of my own. What follows extends the Agentic Authority Architecture (AAA) working paper, whose regulatory spine is the NIST AI Risk Management Framework, with its Govern, Map, Measure and Manage functions, together with CISA guidance. The part of AAA that matters for the handoff problem is a four-tier identity model, namely Organization, then User, then Agent, then Capability, and the insistence that authorization be evaluated at the capability level, at runtime, on every action.

Set against the seven requirements, a defensible architecture looks like this.

Treat every agent as a first-class identity, never a shared service account. This is what restores attribution, and it is the precondition for everything else. Attest each agent's identity with workload-identity infrastructure of the SPIFFE and SPIRE kind.

Put an identity-aware gateway or proxy in front of every cross-agent and cross-tool call. The gateway is the enforcement point. It resolves the effective capability as the intersection of policy and the originating human's ceiling, attenuates scope on each hop, and injects the right on-behalf-of context so the target never sees a naked privileged machine identity. This is the single most effective control, because it sits exactly at the boundary where trust is established.

Mint capabilities just-in-time, attenuated, and short-lived. Each handoff produces a strictly weaker capability, scoped to the immediate task and expiring with it. Use caveat-style attenuation of the macaroon kind where the ecosystem supports it, and tightly scoped exchanged tokens otherwise.

Check authorization at runtime, on every action, not once at the handshake. Let a policy-as-code engine of the OPA kind evaluate each action against the resolved capability and the originating principal.

Bound the delegation depth. If A can spawn B which can spawn C without limit, traceability is already gone. Cap the depth, and require explicit re-authorization to extend it.

Gate irreversible or high-impact actions behind a human checkpoint. This is what the FINRA 2026 oversight guidance points at for agents that act or transact, and it is the backstop for when automated reasoning about scope is uncertain.

Write every handoff to a tamper-evident audit trail keyed to the whole delegation chain. Auditors and responders need to reconstruct which human authorized which agent to do what, at every hop. That is the property the protocols cannot give you today, and the architecture layer is where you supply it.

The pattern in plain terms: make the boundary the place where you enforce, narrow authority as it crosses, and never let an agent's own privilege stand in for the human ceiling behind it.

9 What Enterprises Should Do in the Next Two Quarters

For a CAIO who wants an action list rather than an architecture diagram, here is where to start.

1. **Inventory your agents as identities.** Sweep across browser extensions, IDE plugins and any SaaS feature labeled "AI automation", because they all count. You cannot govern handoffs you cannot see.
2. **Classify access.** Map each agent's tool access, the data it can read, and the systems it can write to. That gives you a risk-surface map. Prioritize the agents that can both read sensitive data and write to external systems.
3. **Get rid of shared agent credentials.** Re-issue per-agent identities. This one change restores attribution and unlocks every control downstream of it.
4. **Put a gateway between your agents and everything they touch.** Enforce attenuation and on-behalf-of context centrally, rather than hoping each agent does the right thing on its own.
5. **Cap delegation depth and forbid raw credential forwarding.** Default to scoped, time-limited, task-bound capabilities.
6. **Gate irreversible actions behind human approval,** especially anything that transacts, deletes, deploys, or crosses an organizational boundary.
7. **Audit the chain, not just the call.** Require that every handoff can be reconstructed end to end.

None of this needs the standards to settle first. It needs you to treat agents the way you treat staff: an identity, a manager, and a paper trail.

10 Open Problems for Researchers

For the PhD students and fellow researchers reading, the most consequential problems here are unsolved and well-posed.

1. **Formal multi-principal attribution at hop n.** Build a delegation primitive that cryptographically binds the originating human to a specific action at arbitrary depth, and that survives cross-domain hops with no federation set up in advance. This is the core gap the standards literature names outright.
2. **Decidable, low-latency runtime scope checking.** Per-action authorization against a resolved capability has to be cheap enough to run on every call at machine speed. Which policy languages give you both real expressiveness and a tractable decision procedure under attenuation?
3. **Verifiable monotonic attenuation across heterogeneous protocols.** Picture macaroon-style caveats interoperating across MCP, A2A and OAuth domains, with a revocation story that holds up.

4. **Benchmarks for multi-agent privilege escalation.** The field needs shared, reproducible benchmarks for confused-deputy and contagion attacks across frameworks, extending the mandatory-access-control and formalization work now appearing on arXiv, so that defenses can be compared instead of asserted.

5. **Detecting injection-driven re-delegation.** Telling a legitimate handoff apart from one provoked by injection, at the boundary, before the action runs.

A group that produced a verifiable, attenuable, cross-domain delegation token with a formal attribution guarantee would not be writing a paper. It would be writing the missing layer of the agentic identity stack.

Conclusion

The agent-to-agent handoff is the seam along which multi-agent systems will most often fail in 2026, and it fails for a structural reason rather than an accidental one. Our authorization stack was built for one human, consenting once, delegating one hop. Agents act asynchronously, chain across domains, and re-delegate on their own, and by the third or fourth hop no deployed protocol can say whose authority is being spent. Identity collapses to the agent, scope inflates instead of narrowing, and trust turns contagious.

The fix is not one protocol. It is an architecture that makes the boundary the enforcement point, narrows authority as it crosses, checks authorization on every action at runtime, and records the whole chain. The standards are converging on the same message, and they are within reach but not yet settled. The enterprises that start treating agents as governed identities now, with their own lifecycle, least privilege, and a paper trail, will be the ones whose multi-agent ambitions survive contact with production.

When software becomes staff, it needs an identity, a manager, and a paper trail. The handoff is where you find out whether it has them.

References

[1] Cloud Security Alliance. *State of AI Cybersecurity 2026* (92 percent of security leaders concerned about AI agents; govern agents as identities). May 2026.

[2] Forrester. *Top Cybersecurity Threats in 2026* (AI identity sprawl among the top five emerging CISO risk categories; rogue agents; supply-chain exposure). 2026.

[3] OWASP Agentic Security Initiative. *Top 10 for Agentic Applications* (tool misuse, memory poisoning, identity abuse; Excessive Agency; Insufficient Authorization).

[4] NIST NCCoE. *Accelerating the Adoption of Software and AI Agent Identity and Authorization* (concept paper; candidate standards MCP, OAuth 2.1, OIDC, SPIFFE/SPIRE, SCIM). February 2026. See also NIST AI RMF 1.0 (2023) for the governance spine.

[5] OpenID Foundation. *Identity Management for Agentic AI* (three gaps in current OAuth and OIDC for agents). October 2025.

[6] *AI Identity: Standards, Gaps, and the Unanchored Delegation Chain.* arXiv:2604.23280 (no deployed protocol proves which principal authorized which agent at hop three or four; structural assumptions of OAuth, SAML and OIDC). April 2026.

[7] *AIP: Agent Identity Protocol for Verifiable Delegation Across MCP and A2A.* arXiv:2603.24775 (OAuth structural limits; macaroons; cross-domain verification).

[8] *A Framework for Formalizing LLM Agent Security.* arXiv:2603.19469 (the formal confused-deputy condition).

[9] *Taming Various Privilege Escalation in LLM-Based Agent Systems: A Mandatory Access Control Framework.* arXiv:2601.11893 (multi-agent confused-deputy attacks via injected and untrusted agents; demonstrated across state-of-the-art frameworks).

[10] *Intelligent AI Delegation.* arXiv:2602.11865 (just-in-time, attenuated permissioning; recursive delegation and privilege attenuation; confused deputy, Hardy 1988).

[11] *Who Governs the Machine? A Machine Identity Governance Taxonomy (MIGT).* arXiv:2604.06148 (non-human-identity definitions; NHI-to-human ratios; NIST AI RMF gaps).

[12] Cloud Security Alliance Lab Space. *Confused Deputy Attacks on Autonomous AI Agents* (authority re-delegation; multi-agent propagation; the Cline incident); and *AI Agent Identity Crisis: Standards Emerge as Enterprises Lag* (Okta agentic IAM framework; CSA and Oasis 79 percent ill-equipped; Entro 97 percent excessive privilege; Obsidian 90 percent over-permissioned). 2026.

[13] Andromeda Security. *The Permission Gap: Solving the AI Agent Confused Deputy Problem* (effective capability as policy intersected with the human ceiling; RFC 8693 on-behalf-of mechanics; identity collapse under agent-owned credentials).

[14] Aembit. *MCP Permission Models* (per-client consent; MCP confused-deputy mechanics).

[15] RFC 8693. *OAuth 2.0 Token Exchange* (delegated and on-behalf-of tokens; the *act* claim).

[16] EIC 2026 coverage (Corbado): runtime authorization; the Laws of AIdentity; OAuth 2.1, MCP and A2A; "when software becomes staff."

[17] Non-human-identity reporting, including the Salesloft and Drift 2025 token pivot across more than 700 connected companies, and FINRA 2026 oversight guidance on human checkpoints before agents act or transact.