The Seventh Question

Who authorized the agent, and is that authority still standing?

Detection tells you an agent acted. It does not tell you who let it.

The advice reaching security leaders this year is consistent: AI agents are arriving, so learn to detect them, score them, and decide in real time whether to trust the session. That advice is sound. It also stops at the front door.

Behind the door, every organization already runs hundreds or thousands of non-human actors of its own. Service accounts. App registrations. Workload identities. Automation runners. Vendor connectors. And now, agents shipped inside products already licensed. Each one holds standing authority. Each one was, at some point, allowed by someone.

The question nobody is asking about those actors is not "is this one risky right now?" It is **who authorized it, is that person still in a position to have done so, and did the actor stay inside what was granted?** That is the seventh question. The answer is not in the traffic. It is in the record of who granted what, and whether that record still holds.

Six questions everyone asks. One nobody does.

The agent-trust playbooks now circulating all list the same six questions to answer before allowing a sensitive agent action:

1. Is this a human or an agent?
2. Which agent is it?
3. Who is it acting for?
4. Is it allowed to do this?
5. Is the agent, or its principal, risky?
6. Should this action be trusted right now?

Good questions. All six are answered from the session: the traffic, the device, the policy as it stands at the moment of the call, a risk model. The seventh is answered from the lifetime of the grant:

7. **Who authorized this actor, is that authority still standing, and did execution stay inside it?**

An organization can answer all six and still have no idea that the service account behind its nightly batch was approved in 2021 by a manager who left in 2023, was never re-affirmed, and picked up two extra roles along the way that nobody signed for.

Question 4 is the one most often mistaken for question 7. "Is it allowed?" checks the policy today. It does not ask whether that policy was ever explicitly granted by someone with standing to grant it, or whether that person still has it. Permissions accumulate. Authority is granted, and it either holds or it lapses.

Standing, not score

A trust score is a probability. Standing is a fact. An actor's authority has standing when four things are true at the moment it acts:

- **The grant is explicit.** Someone recorded that this actor may do this thing. Authority inherited from a group, a default, or a vendor's installer is permission without a grantor.
- **The grantor had standing to grant it.** The approver held, at the time, a role that could approve it, recorded as an immutable identity, not a username.
- **The grantor still has standing, or the grant was re-affirmed.** When the approver leaves or changes roles, the authority they granted does not silently survive them. It is unattested until someone with current standing re-affirms it or revokes it.
- **Execution stayed inside the grant.** What the actor actually did, evidenced by receipts from the systems it touched, reconciles against what it was allowed to do.

None of these is a judgment call. Each is a check against a record, which is what makes the seventh question provable in a way the first six are not. An auditor can verify the answer offline, without trusting anyone's word or anyone's model.

> The doctrine in one line: **explicit only, implicit never counts.**

Detection, identity verification, policy enforcement, and trust scoring answer the first six questions, and you should run them. Authority Provenance answers the seventh, and it consumes the other four as evidence. They enforce or infer authority. We prove who granted it, whether it was still valid, and whether execution stayed inside it.

Start with a number

The first authority-exposure number any organization needs is the share of grants held by its non-human actors that have no authorization on record. It is usually the number that gets a program funded, and it is usually not known.

Cerydora's free, in-tenant reach sheet is where that number starts, with read-only directory access. It lists who holds privileged roles, user accounts and service principals alike, which consents are tenant-wide, and what a privileged session can reach. Nothing leaves your tenant. The paid product adds the declared-authority column: for every grant, whether an authorization is on record and whether its authorizer is still standing.

The criteria are published, and so is the evidence format. Any auditor, and any competitor, can check the result.

- Open specification and reference verifier: github.com/csingcloud/authority-provenance
- *Authority Provenance bundle v0.1: specification, field tables and test vectors.* doi:10.5281/zenodo.23601519
- *Authority Provenance: Standing of Delegated Authority Behind Non-Person Actor Actions*, v0.3. doi:10.5281/zenodo.23105657
- *Authority Exposure: Standing Authority Reachable by Undeclared Input*, v0.1. doi:10.5281/zenodo.23318689