Lumos

Role Models: From Chaos to Intelligence

From point-in-time projects to continuous, agent-run role mining.

The role model paradox.

Role-Based Access Control is the access model every IAM program wants to land on. Clean roles, mapped to job functions, with one place to grant access, audit it, and remove it. In theory, RBAC retires the chaos of one-off entitlements and gives you an access posture you can hand to an auditor without flinching.

The trouble starts with how the work gets framed. Role mining is often treated as a point-in-time, "hand clap, we did it" exercise. A consultant comes in, runs workshops, ships a spreadsheet, and someone painstakingly maps the roles into the IdP. The day the project closes, drift begins. New apps come online. Two business units merge. A contractor program triples. A few roles get patched. Most stay exactly as they were on day one.

You can tell a role model has gone stale because the access request queue tells you. People stop requesting roles. They request individual entitlements. Each request makes the role model less useful and the entitlement sprawl harder to govern. The original design becomes a thing on a slide, not a thing in the IdP.

Role mining gets treated as a point-in-time, "hand clap, we did it" exercise, but the **work rarely stays done**.

What stale roles actually cost

Access requests stop being role-based and become one-off entitlement grants.

Provisioning slows. Every new hire becomes a bespoke setup.

Access reviews lose their teeth. Reviewers are handed loose entitlements instead of role assignments, and certifying loose entitlements is what got you here.

The original RBAC investment quietly stops paying back. Auditors notice. Boards notice. The team that built it knows first.

Treating role design as a project guarantees drift the moment the project ends. The work has to be set up as a practice if the role model is going to keep up.

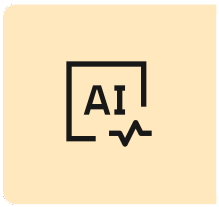
Three approaches that don't keep up.

The market has tried to solve role drift three ways. Each one solves part of the problem. None of them solve the part that compounds: keeping the model current as the business moves.

Approach	Why it stops short
Manual role engineering →	Expensive, slow, and obsolete on day one. The output is correct for the moment it was built and steadily wrong from there. Most teams who go this route do it once.
Legacy IGA role management →	Designed to administer roles that already exist. Solid for maintenance, weak for discovery, blind to drift. You know what your roles are; you do not know whether they still match the business.
Standalone role mining tools →	Capable of crunching usage data and recommending roles, but the output is a report. Implementing the recommendations is still a manual project, and by the time it lands the underlying data has moved on.

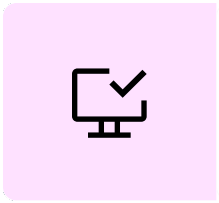
What is missing in all three is the loop. Mining roles once and acting on them once produces a snapshot. A working role model has to be wired into a system that watches usage as it changes, recommends updates as patterns shift, and pushes those updates into the tools your team already uses to provision and review access.

A consultant can deliver a snapshot, but then maintenance has to be provided somewhere else. What your team actually needs is a role model that maintains itself.



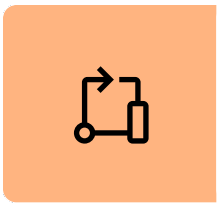
1. AI-powered role discovery

Surfaces candidate roles grounded in how access is actually being used, not how it was modeled three years ago. Patterns the team would never spot manually become named, reviewable proposals.



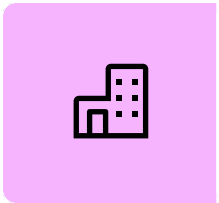
2. Continuous role intelligence

Monitors the role model against real usage and flags drift as it accumulates. New BUs, M&A activity, and contractor cycles show up as role changes the team can act on, not as failures the team has to explain.



3. From roles to action

Connects recommendations directly to provisioning, access requests, and reviews. Approving a role update means the next new hire gets the right access on day one, not a week of one-off requests.



4. Enterprise-ready

Built for the orgs where simple does not apply: multiple business units, contractor populations the IdP barely knows about, mergers that doubled the entitlement footprint last quarter.

Each cycle of usage feedback makes the model sharper. Each new app onboards into a model that knows how to absorb it. Roles stop being a thing you redo every few years and become a thing your access governance runs on.

What the loop actually does

A marketing team adopts a new content tool. Twelve people request access in the first month. Lumos sees the pattern, notes that the affected users already share most of their other entitlements, and proposes adding the new tool to the existing Marketing role. The IAM lead approves the change once. Every marketer hired after that gets the new tool on day one without filing a request.

The same loop runs in the other direction. When a business unit divests or a team is reorganized, role members stop touching whole categories of tools. The role they shared becomes an outlier against actual usage, and Lumos flags it as a candidate for revision or retirement. Access shrinks at the rate the business is changing, not at the rate annual reviews catch up to it.

Agentic role mining: the way forward.

Role mining needs to become a continuous exercise to keep up with the pace of modern identity. That requires agents that can run the loop. Enterprise environments are changing too fast for a human-driven cadence to track. At quarterly intervals the team falls behind. At weekly intervals the work outpaces what any reasonably sized IAM team can absorb. The only way to close the gap is to move the watching and the proposing out of the team's queue and into an agent that runs continuously in the background.

The agent's job is bounded: watch usage against the role model, generate role recommendations with the evidence and confidence needed to act on them, learn from each accept and reject, and push approved changes into the provisioning and access request workflows the team already runs.

Watches drift

Continuously evaluates the role model against how access is actually being used. Surfaces divergence the moment it crosses a threshold, not when the next review starts.

Proposes with evidence

Each recommendation arrives with the usage data, peer comparisons, and a confidence score the team needs to approve or dismiss in seconds, not days.

Learns from your team

Every accept or reject shapes the next round. The agent gets sharper with each pass, and the model converges on what your business actually needs.

Agentic role mining can't be a separate platform. It has to be baked into the models and tools the team is already using — the role data, the provisioning workflows, the access request and review systems — because that's the only way an agent's recommendations become governed changes instead of another dashboard of findings.

Every recommendation touches a role that governs access for a population of employees. Because the workflow is that high-impact, it has to be built on progressive trust. Start with human oversight to guide the agents and see their recommendations. Delegate as the team builds confidence. Calibration happens in the first session, not a separate training cycle: every accept or reject shapes the next round, and the agent converges on what the team needs in days. This is the same model Lumos built Albus and the Identity Security Agents on.

Where this shows up.

Moving from one-and-done to continuous to agentic runs on the role model the team already owns, fed by usage and entitlement data Lumos is already collecting, and connected to the workflows the team is already running.

The practical effect

- Access requests trend toward roles and away from one-off entitlements.
- New hire provisioning gets faster and more consistent, because the role that fits the job actually fits the job.
- Access reviews finally have something meaningful to review: role assignments, not loose bundles of entitlements.
- The role model stops being a perennial project on the roadmap and starts being a working system in the IdP.
- Drift gets caught and surfaced by an agent watching continuously, instead of accumulating until the next review cycle.

The question worth asking

If your team is weighing a role mining project right now, the question worth asking is not whether the project will produce a clean role model. It will. The question is whether the model will still be clean a quarter after the project ends. The honest answer is no, unless something is running the loop after the consultants leave.

See it in action

Lumos brings continuous role mining, agentic role intelligence, and full IGA into one platform, so your role model stops being a historical artifact and starts running with the business. Request a personalized walkthrough at lumos.com.

Book a Demo