empowerID

WHITEPAPER · IDENTITY FABRIC

One decision authority. Every relevant fact.

How EmpowerID Governed Authorization decides access for applications, data, and AI agents with one logical ABAC authority — fed by relationship intelligence, enforced everywhere, explainable to anyone.

July 2026 For security & identity leaders and architects

EXECUTIVE SUMMARY

Enterprise authorization breaks when every application, gateway, graph, and AI agent creates its own truth. EmpowerID Governed Authorization places *one logical ABAC decision authority* at the center of the authorization lifecycle — from business-defined capabilities and live context to enforcement, explanation, and safe change.

Most authorization platforms begin with the policy check. EmpowerID governs what that check means, which facts it sees, every reason the authority exists, and — when the subject is an AI agent — what may be delegated, discovered, and invoked. Relationship intelligence (ReBAC) is a graph-powered super PIP — a policy information point, the standards term for a source of facts the decision engine consults. The graph feeds the decision; it never issues a competing permit. And "one authority" is a

logical claim, not a physical one: policy decision points (PDPs) — the engines that evaluate policy and return the decision — run wherever applications and agents operate, under one governed policy model.

01 — THE PROBLEM

Fragmented authorization truth

Most organizations do not lack authorization technology. They lack a single authorization truth. Roles live in directories and describe broad capability, not the current action. Relationships — ownership, team membership, management scope, delegation — determine real access boundaries but are rarely available where decisions are made. Attributes and risk change what should be allowed right now. And enforcement is scattered across UI code, gateways, backend checks, database filters, and AI-agent tool gateways — each interpreting access on its own terms. When these fragments disagree, three failures follow:

INCONSISTENT DECISIONS

The UI hides a button while the API still permits the action — or an agent path allows what the human path denies.

UNEXPLAINABLE ACCESS

No one can say why a specific user or agent can perform a specific operation on a specific record today.

UNSAFE REVOCATION

Removing one assignment silently destroys access also justified by another legitimate source — or removes nothing at all.

AI agents multiply every one of these gaps: an agent can discover tools, cross APIs, and act through delegated authority faster than session-era, token-frozen entitlements can react. Governed Authorization addresses the root cause — it gives all of those facts one decision authority and connects that authority to every supported enforcement point.

02 — HOW IT WORKS

Meaning, facts, decision, enforcement, proof

Governed Authorization is not a "request in, permit out" chain. Contracts and governed assignments shape policy before any request arrives; policy information points (PIPs) assemble current facts at the moment a request is evaluated; explanation and change evidence remain distinct outputs. It answers

one question, everywhere it matters: *may this subject perform this action on this resource under the current context?*

`PIP` information point — supplies the facts

`PDP` decision point — evaluates policy and decides

`PEP` enforcement point — applies the decision

1

Applications define what access means

An App Authorization Contract declares features, operations, object-level requirements, and required enforcement locations — engine-neutral semantics projected into policy, grants, and tests. Authorization begins here, not at the runtime request.

2

The business composes and binds authority

Bundles group capabilities; Persona Profiles compose them into a business job; environment bindings decide who receives them, for how long, under which approval. For agents, the assignment is a bounded delegation — never a copied human role.

3

PIPs assemble current facts

Role and grant facts, identity and resource attributes, risk and environment — and relationship facts from the graph/ReBAC super PIP: ownership, membership, management scope, sharing, delegation. The graph contributes facts; it never issues a permit.

4

One logical ABAC authority decides

A policy enforcement point (PEP) sends a standards-based OpenID AuthZEN request; the serving PDP instance evaluates the governed policy model and returns the decision and its context.

5

PEPs enforce; the system explains

UI, gateway, backend, search, and agent PEPs enforce the same decision and may fail closed. The platform explains why access exists now and records who changed authorization configuration, under which authority, with what impact.

Figure 1 — The authorization lifecycle: business-defined meaning through decision, enforcement, and explanation.

> **The normative rule.** PIPs contribute facts. The PDP authorizes. PEPs enforce and may further narrow or deny. No downstream component may widen authority granted by the PDP. A role may establish the capability to attempt an operation — the ABAC authority still determines whether *this* subject may perform it on *this* resource, through these relationships, under these conditions.

03 — WHAT MAKES IT DIFFERENT

A governed authority, not another policy engine

Externalized PDPs, relationship graphs, real-time context, and agent tool filtering all exist across the market. The differentiation is the combination — and the discipline of where each capability sits:

Dimension	Typical authorization platform	EmpowerID Governed Authorization
Starting point	A runtime request arrives	The application defines what access means, before any request
Role of the graph	ReBAC issues its own permit — a second truth to reconcile	Graph super PIP contributes relationship facts to one ABAC authority
Decision topology	One central PDP server — or divergent local engines	One governed policy model; distributed PDP instances with identical semantics
Revocation	"Delete the grant" — and hope nothing else depended on it	Contribution-aware: remove one reason, preserve every other legitimate reason
AI agents	Gateway traffic filtering; agents inherit human tokens	Delegation, discovery, and invocation governed as three distinct moments
Interface	Proprietary SDK per enforcement point	OpenID AuthZEN between every PEP and the PDP — interop-tested December 2024
Explanation	An audit log, reconstructed after the fact	Present-state "why allowed / why denied" kept distinct from change history

04 — ARCHITECTURE

One logical authority, distributed decisioning

One logical ABAC decision authority does not mean one centralized PDP server. It is an architectural invariant about who may decide, not a constraint on where decisions run. A governed control plane distributes policies, schemas, and versions to a fabric of PDP instances across regions, clusters, and deployment boundaries; each serving instance applies the same policy model, canonical PIP contracts, and decision semantics, and every decision lands on a unified evidence plane.

`one policy model` not one server

`one decision model` not one network endpoint

`central governance` not necessarily central execution

`distributed PDPs` consistent semantics and evidence

Three authorization patterns cover the practical integration surface: evaluation (may this subject perform this action on this resource?), batch evaluation (which of these actions or resources are permitted? — UI capability loading), and authorized search (which records may this subject see?). For search, authorization predicates are pushed into the backing query where supported, and results, totals, and pagination are filtered consistently — a result count never leaks what a record-level check would deny.

05 — BUILT FOR THE AGENT ERA

Authorization before cognition.

Reauthorization before action.

EmpowerID does not give an AI agent a copy of a human's permissions. Before an agent plans, the platform narrows the tools available for the current user and delegation. When the agent invokes a tool, the authorization pipeline checks again with current policy and context. The same logical ABAC authority governs three distinct moments:

1 · Delegation

May this user delegate this tool to this agent? Delegatable tools = agent capability ceiling ∩ owner delegation authority. The right to perform is not the right to delegate.

2 · Discovery

Which tools are effective for this user, agent, and active delegation? The agent sees a policy-scoped tool surface before it plans — not its full registered toolbelt.

3 · Invocation

May this agent use this tool on this resource now? Current delegation, graph, identity, trust, risk, and data-scope facts are evaluated again at action time.

> **Every stage can narrow. No stage can widen.** Effective tools = persisted agent definition ∩ active delegation ∩ live tool catalog ∩ runtime policy. An agent never gains a tool because it appears in a prompt, request payload, stale definition, or untrusted header — and enforcement fails closed: unknown agents are rejected, direct invocation cannot bypass policy-scoped discovery, and absent an active delegation the tool surface collapses to a minimal documented baseline.

The graph super PIP matters most here, because an agent request is relational by nature: who owns the agent, whether an active delegation connects this human and agent, what the delegation contains, its expiry and consent state, and how the delegator relates to the target resource. The graph understands the delegation chain; one logical ABAC authority decides whether the chain authorizes this action now.

06 — GOVERNANCE AND AUDIT

Every reason recorded. Every removal safe.

One effective capability can be justified by several active sources — a bundle, a persona, an import, a governed direct assignment. The Grant Contribution Registry records each reason separately and materializes the effective grant from their union. Removing one contribution does not destroy access still justified by another; access disappears when the final valid contribution disappears. Revocation is not "delete the grant" — it is "remove this reason, preserve every other legitimate reason, and explain what remains."

Two evidence questions stay deliberately distinct: the active authorization state answers *why allowed or denied now*; the configuration change record answers *who changed what, under which authority, with which approval, and with what before-and-after impact*. An audit log alone cannot serve as

current authorization truth, and a current-state graph cannot serve as change history. EmpowerID keeps both.

07 — EVALUATING PLATFORMS

Five questions to ask any authorization platform

1. Does the graph issue a second permit — or contribute facts to one policy authority?
2. Does authorization begin with application-defined semantics, or only with a runtime request?
3. Can it explain every active contribution to a subject's access right now — and remove one reason while preserving the others?
4. Does agent authorization govern delegation, discovery, and invocation as three distinct moments — or only gateway traffic?
5. Can UI, API, backend, search, and agent enforcement points share one decision authority?

Governed Authorization is designed so the answer to each is the strong one. It works alongside the rest of the Identity Fabric: continuous access updates the facts the next decision sees; agent identity and federation establish the verified identity authorization consumes; Credential Control governs secrets after a permit; Governed Execution corroborates the external effect. Each is covered in its own EmpowerID publication, available on request.

KEY TAKEAWAYS

One decision authority. Every relevant fact. No parallel authorization truth.

One governed policy model — decided wherever your applications and agents run.

Authorization before cognition. Reauthorization before action.

Declare the capability. Bound the scope. Gate the action. Prove the chain.

See one authority decide, everywhere

Watch an application capability, a human persona, and a delegated agent resolve to the same governed decision — with the explanation to prove it.

Request a demo

info@empowerID.com · empowerid.com

empowerID