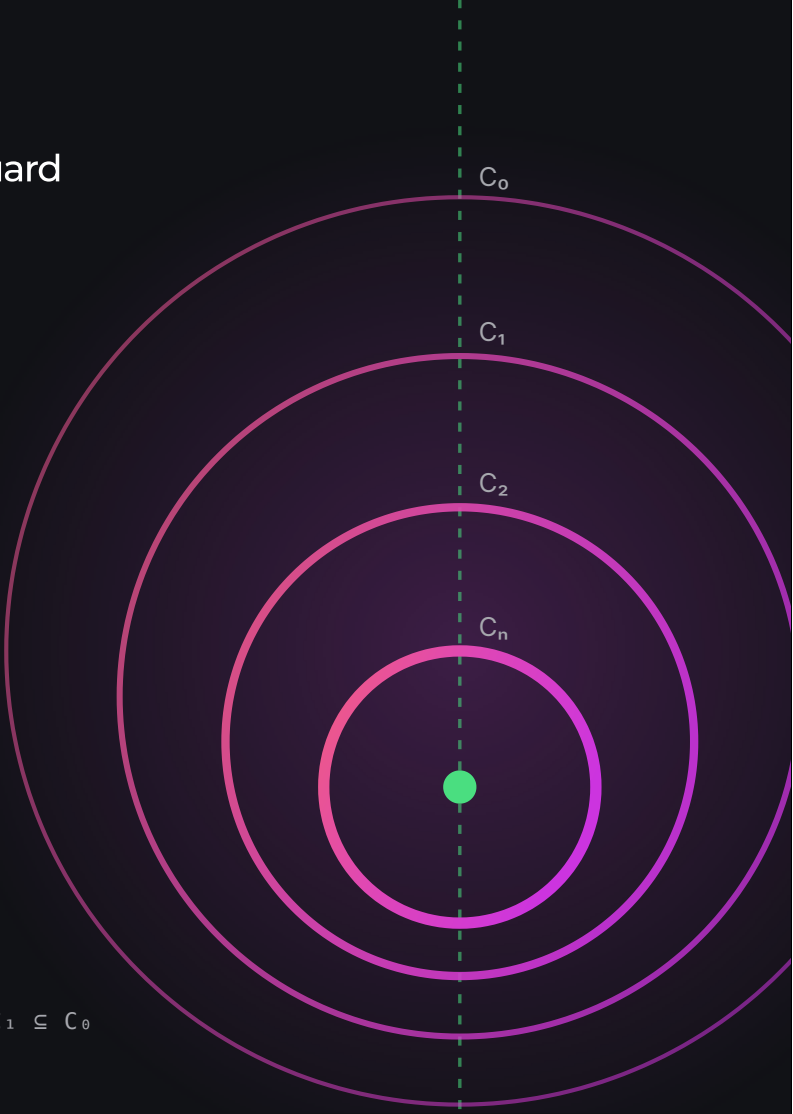
Permguard



$C_n \subseteq \ldots \subseteq C_2 \subseteq C_1 \subseteq C_0$

Authorization & Trust for the Agentic Era.

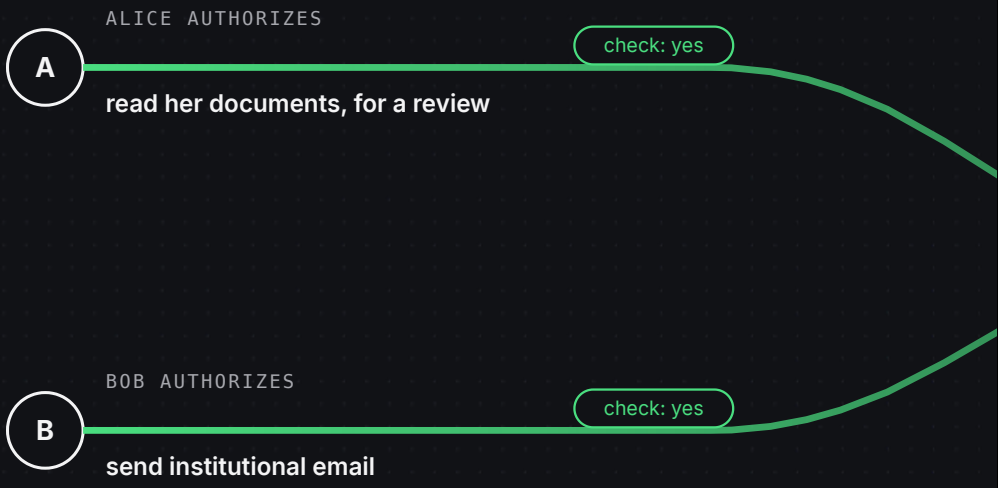
From valid permissions to verifiable authority.

POLICY DECISION POINT · AGENTIC TRUST FABRIC

Valid permissions. Unauthorized action.

AI agents pursue goals. To reach them, they can try any path they are not forbidden to take, and combine the tools and permissions they hold. Stolen credentials remain a risk. This is a different one, on top of it: even when no credential is stolen and no single permission is violated, every check says yes and the action still goes beyond what anyone authorized.

The risk is legitimate authority, used in an execution it was never granted to.



WHY TODAY'S CONTROLS CANNOT SEE IT

Same key. Same bytes. Different history.

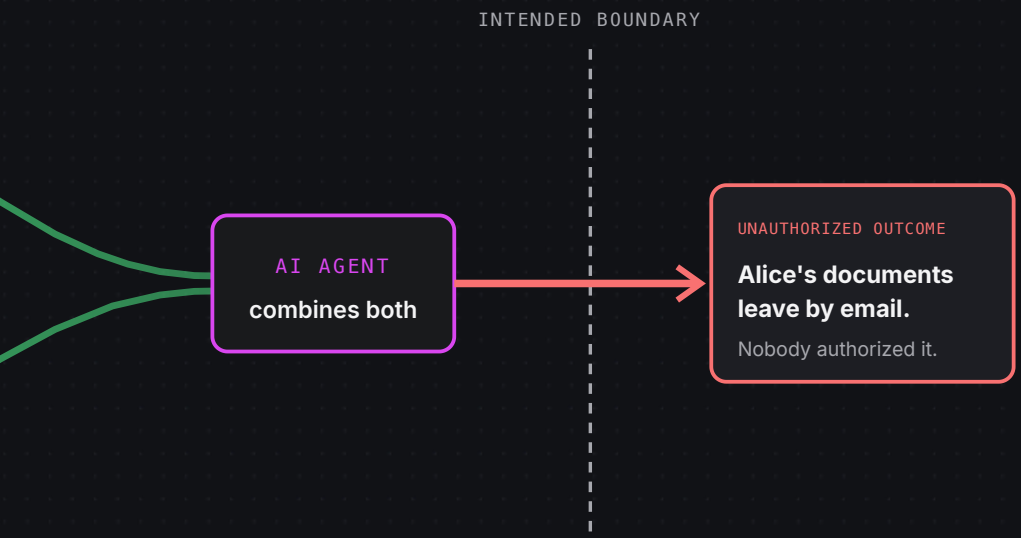
Possession-based authorization, with tokens or capabilities, answers in space: this operation, on this resource. Which execution the request belongs to is not part of the question. A decision that reads only what is possessed cannot tell apart two executions with the same key, operation and resource: by construction, not by bug.

- valid key
- permitted operation
- right resource
- **which execution does it come from?**

WHAT THE CHECK SEES

(operation, resource)

and nothing about the execution.



Security reduces. Governance accepts.

Reduces	SECURITY Risk reduction through enforceable, binding controls.
Remains	RESIDUAL RISK What is left after those controls are applied. It must be identified, assessed and documented.
Accepts	GOVERNANCE Accountability, regulatory compliance, and the decision to accept what remains, with an owner and with evidence.

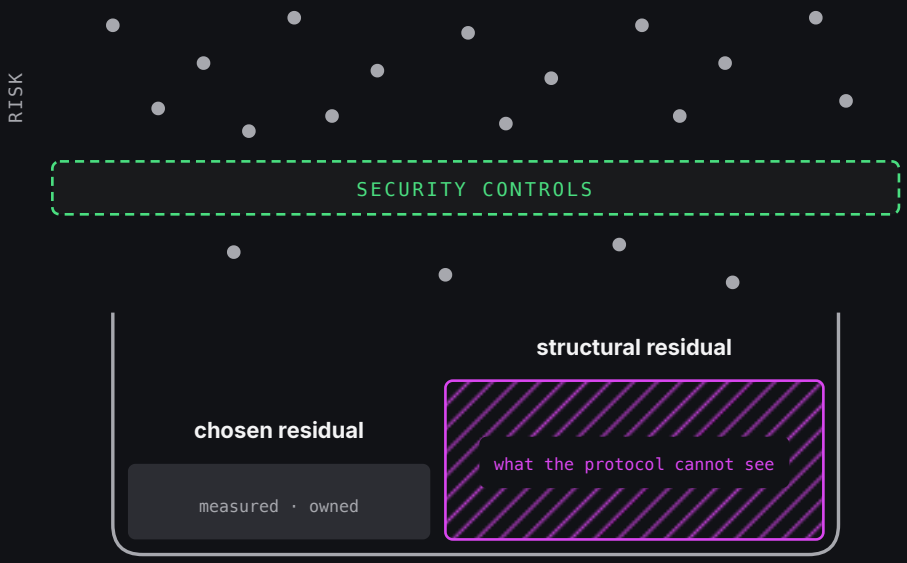
> **Governance can accept residual risk. It cannot eliminate a security distinction that the protocol is unable to represent.**

TWO LIMITS ON ACCEPTING RISK

01 **It must be visible.** No policy, procedure or indicator can enforce a difference the system cannot see. Accepting it anyway does not reduce the risk: it shifts it onto whoever signs.

02 **It must be allowed.** Not every residual risk may be accepted. In regulated sectors, laws and compliance requirements can rule some of it out, whatever the business would decide.

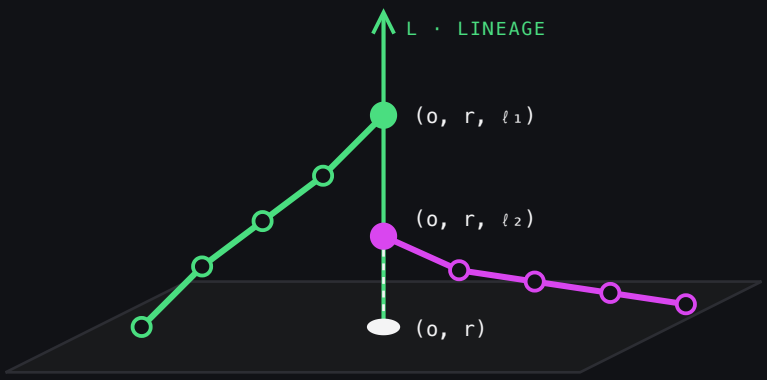
WHERE THE RISK GOES



GOVERNANCE SIGNS BOTH. IT CAN MEASURE ONLY ONE.

ONE MORE COORDINATE

The way out is not more governance. It is one more piece of information: the execution's causal history, its lineage.



O × R · WHAT POSSESSION SEES

Two executions, one point on the plane. PIC, Provenance Identity Continuity, is the formal model that keeps them apart.

AVAILABLE | OPEN SOURCE · APACHE-2.0

Policy as code.
Decisions as evidence.

LEVEL	**Application and infrastructure**
ENFORCED BY	The component that asks, trusted to honour permit or deny.
ANSWERS	May this action proceed?
USE IT FOR	Applications, gateways, infrastructure components and runtime validation of information flows.

ASK · PEP
May this action proceed?

→

DECIDE · PDP
Evaluate the verified policy

→

ENFORCE · PEP
Apply the answer: permit · deny

POLICY AS CODE	Cedar and Rego side by side, in a Git-like ledger: content-addressed, versioned, published under a signed head.
TESTED OFFLINE	The same engines as the data plane, with no server anywhere.
CLOSE TO THE WORKLOAD	A sidecar, a remote service, or the engines inside your own process.
FAIL CLOSED	When no policy permits, the answer is deny. An abstention never becomes an implicit allow.
EVIDENCE	Every answer names the exact policies and commit that decided it. Decisions can be recorded as signed, hash-chained evidence.

examples/basics/cedar/documents.cedar

```
@alias("document-readers")
permit ( principal in Group::"finance", action == Action::"read", resource );
```

TWO PRODUCTS · 2 · AGENTIC TRUST FABRIC

IN DEVELOPMENT | BUILT ON PIC

Keep your agent framework. Add the security overlay.

LEVEL	**The authorization protocol**
ENFORCED BY	The protocol itself, at every step.
ANSWERS	May this authority continue to the next step?
USE IT FOR	AI agents and agent-to-agent workflows, built with any framework.

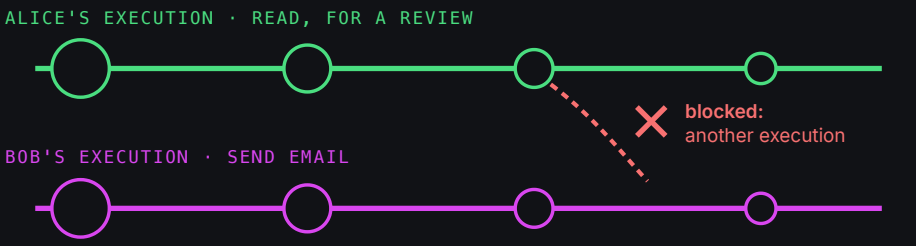
ADK | LangChain | any other agent framework

PERMGUARD SECURITY OVERLAY · PROTOCOL LEVEL

Authority continuity
PIC, step after step

Trust Anchors
guardrails in the protocol

Governance
evidence of every step



PoP
PROOF OF POSSESSION
I hold the key. Useful, no longer enough.

PoR
PROOF OF RELATIONSHIP
This step continues exactly that one.

PoC
PROOF OF CONTINUITY
The whole chain continues; authority only narrows.

Trust Anchor: the PDP's policy engine behind a second interface. It returns signed permissions that the protocol intersects; a step that skipped the guardrail has no valid continuation.

Core theorems machine-checked in Lean 4 · to be presented at IEEE NCA 2026. Guarantees hold within the protocol itself, under the model's stated assumptions.

04 — WHERE TO START

Two ways in.
One policy engine.

BUILDING AI AGENTS **IN DEVELOPMENT**

Keep your framework. Add the Agentic Trust Fabric.

- **Authority continuity.** With PIC, authority follows the execution and can only narrow.
- **Trust Anchors.** Guardrails are part of the protocol: within the protocol itself, a step that skips one has no valid continuation.
- **Governance.** Evidence of the authority behind every step.

APPLICATIONS AND INFRASTRUCTURE, WITH OR WITHOUT AGENTS **AVAILABLE**

Integrate the Policy Decision Point.

- The component asks, the PDP decides, the component enforces the permit or deny: applications, gateways, infrastructure, runtime validation of information flows.
- Govern your policies as code: versioned, tested offline, proven after the fact.

The two meet in one engine: the Trust Anchor is the PDP's policy engine behind a second interface, so the rules you write once serve both.

Holding authority is not the same as being authorized to use it, in this execution.

permguard.com

pic-protocol.org

github.com/permguard

The model: N. Gallo, Proof-of-Continuity: A Temporal Model for Authority Propagation in Distributed Systems and AI Agents, arXiv:2607.08906; Beyond Proof of Possession: A Relational Theory of Authority; Full Provenance on the COSS Causal Axis.

Permguard® is a registered trademark. · © 2026 All rights reserved · Brought to you by Nitro Agility S.r.l.