DENIED

MCP DEV SUMMIT NORTH AMERICA · APRIL 2026

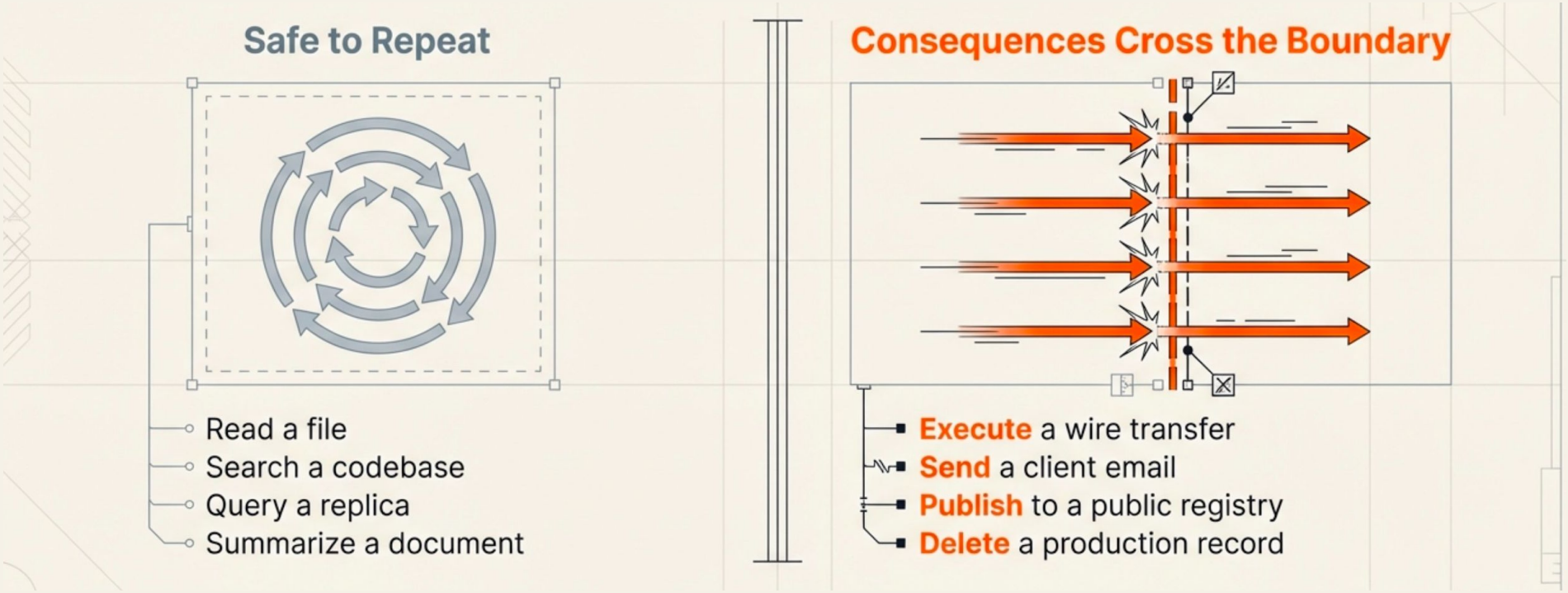
Beyond the Sandbox: Security at the Host Layer

Why the authorization boundary for autonomous agents belongs where context is richest.

Lorenzo Verna & **Pietro Valfrè**
Co-founders, Denied · denied.dev

THE PROBLEM

Agents Walk Through One-Way Doors

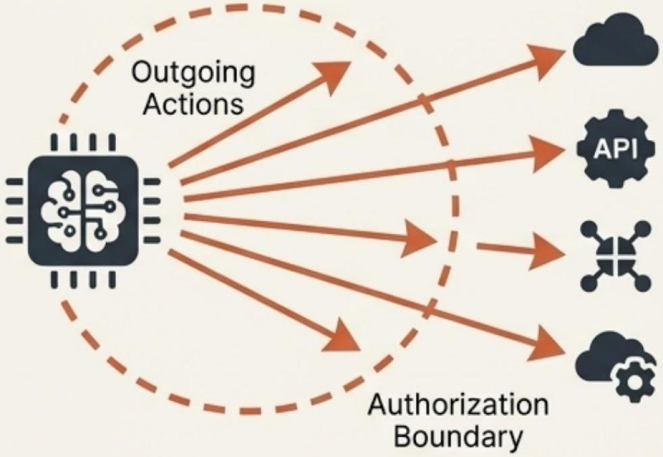


Once the action crosses the boundary, there is no `git revert`.

The Protection Direction Inverts

Traditional: Protect internal resources from incoming requests.

Agentic: Authorize outgoing actions before they hit the external world.



Three Properties Change:		
1. Enforcement Point: Moves to where the agent produces actions.	**2. Evaluation Depth:** Must inspect unstructured content, not just metadata.	**3. Visibility:** Requires a cross-tool view of all outbound behavior.

THE STATUS QUO

Four Common Approaches. Each Has a Blind Spot.

Sandboxing

Isolate the runtime. Block ports, restrict filesystem, filter network access.

> The moment the agent needs to reach an external system, the sandbox must open. Once open, it controls environment, not behavior.

Resource Authorization (Tokens, Scopes, RBAC)

Grant credentials with specific permissions. Scope access via OAuth, JWT, roles.

> Scopes define what the agent can reach, not what it does with that access. A valid send_email scope allows sending anything to anyone.

Prompt Instructions

Rules in the agent instructions.
“Do not do X. Always check before Y.”

> Not inspectable. Not auditable. Not testable. Can be overridden by prompt injection or ignored by the model.

Server-Side / Gateway Authorization

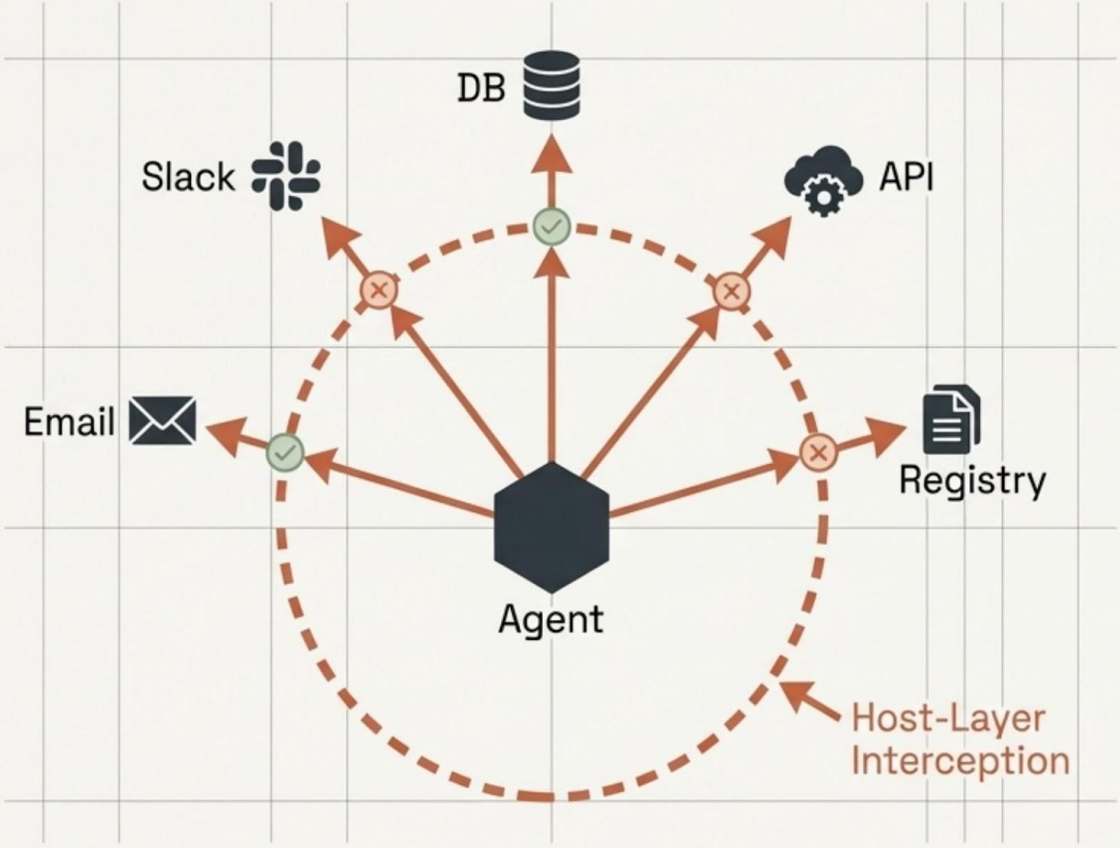
Enforce authorization at the tool server or API gateway.

> Each server sees only its own calls. No server sees the full behavioral pattern across all integrations.

Each is necessary. None evaluates the agent’s outbound behavior as a whole.

THE ARCHITECTURE

Put the Boundary Around the Agent



1. Global Visibility:

Sees every tool call across all integrations, not just isolated servers.

2. Content-Aware:

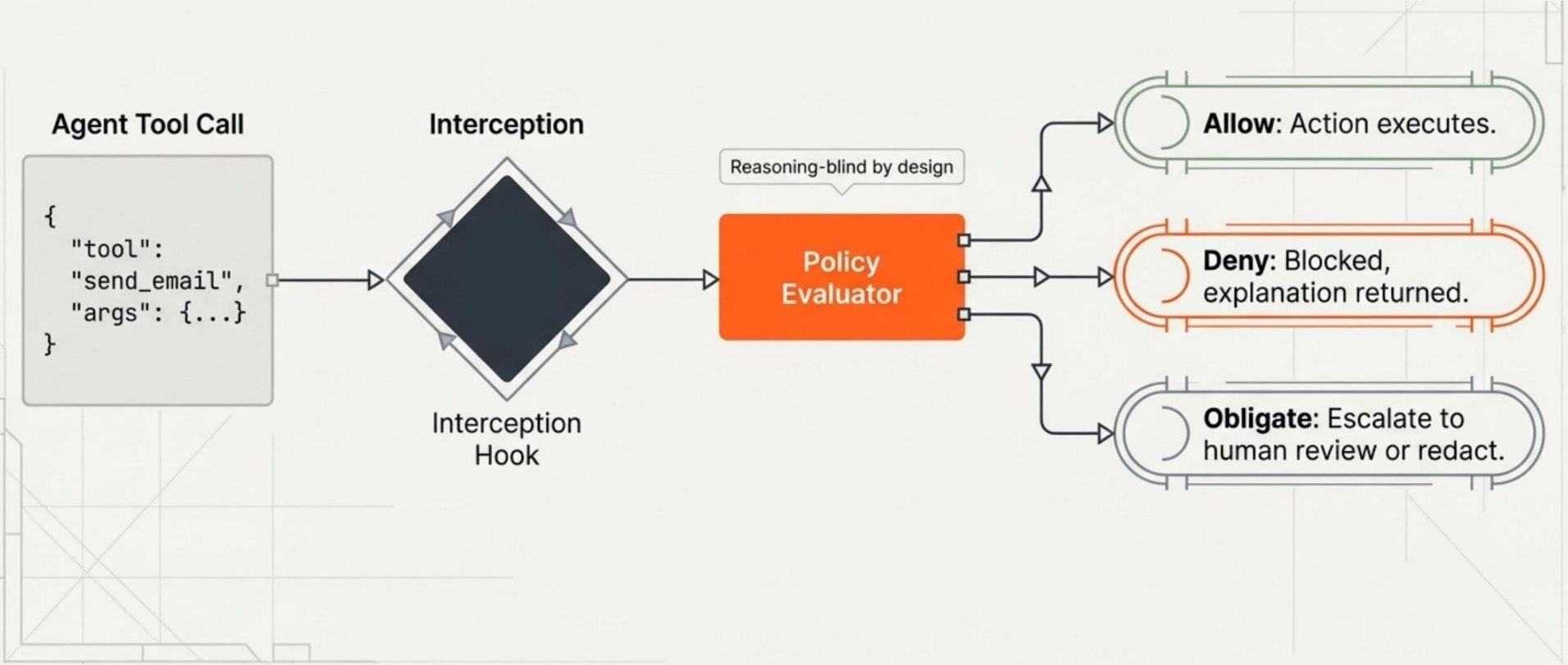
Evaluates parameters, payload, and content, going beyond identity tokens.

3. Cognitive Isolation:

Operates completely independently from the agent's own reasoning to prevent self-persuasion and prompt overrides.

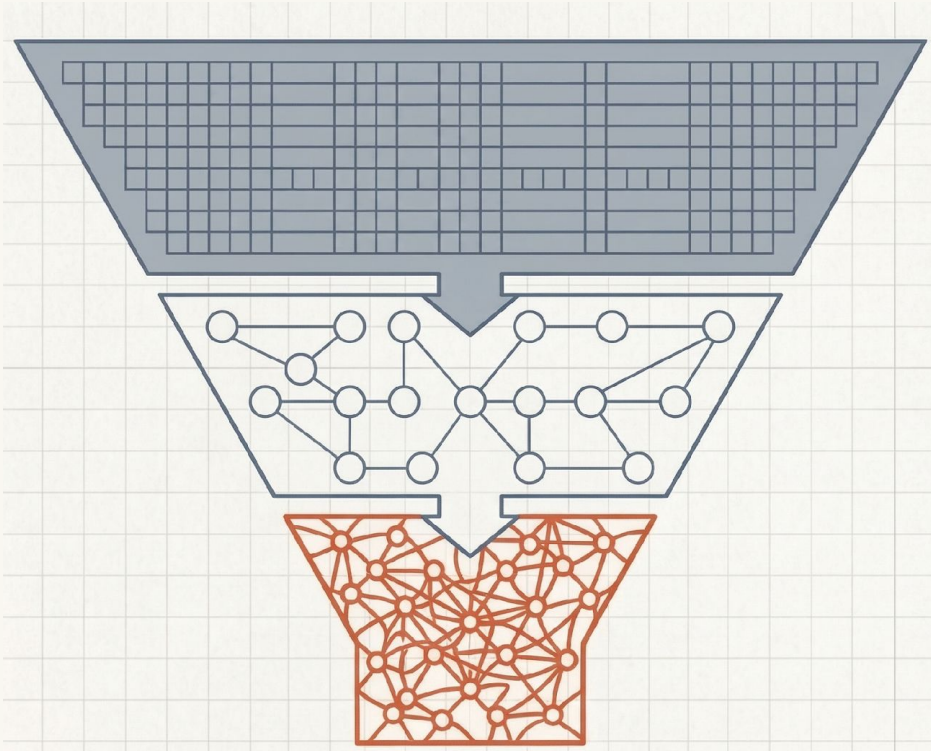
THE MECHANISM

Intercept. Evaluate. Decide.



EVALUATION

Rules Where You Can. Models Where You Must.



Tier 1: Deterministic Rules (Rego)

Evaluates structured parameters (domains, paths, tool names). Fast, predictable, sub-millisecond resolution.

Status: In production.

Tier 2: LLM Content Classification

Inspects the unstructured payload meaning. Catches what strict rules miss (e.g., sensitive data in an email body).

Status: In production.

Tier 3: Specialized Policy Models

Handles complex ambiguity. Evaluates structured natural language policies using specialized models (e.g., gpt-oss-safeguard).

Status: In development.

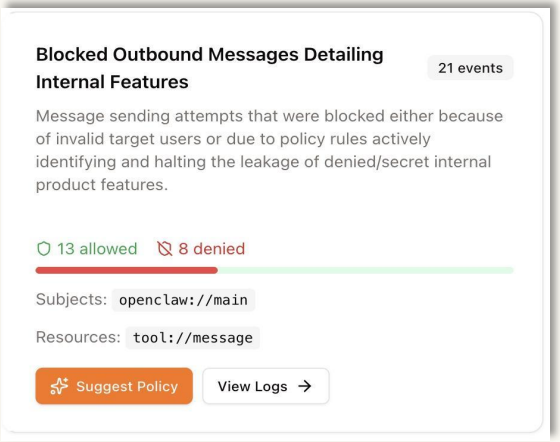
Deterministic rules handle the volume. Model-based evaluation handles the ambiguity.

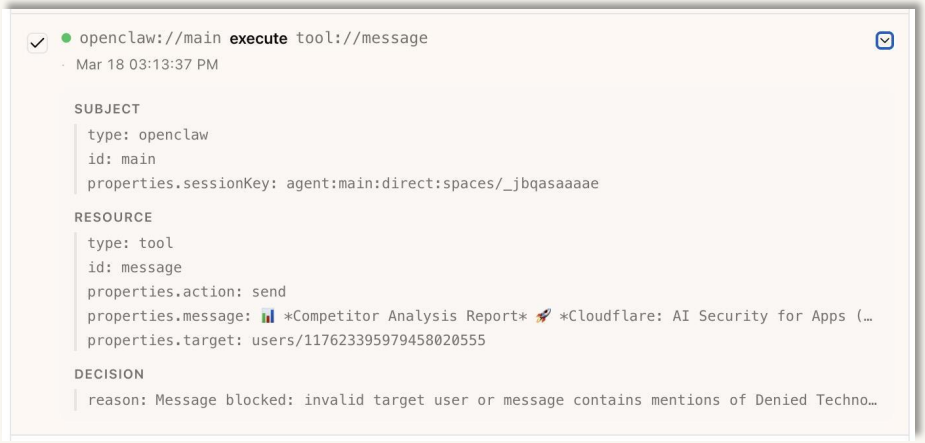
OPERATIONS

Build Policies From Real Behavior

1	2	3	4	5
Observe	**Log**	**Analyze**	**Define**	**Enforce**
Agent operates, tool calls intercepted	Every decision logged with full context	Review logs, identify patterns	Create or refine policies, deploy	All calls evaluated against policies

↻ Iterate from real behavior, not hypothetical threat models.





CONVERGENCE

A Shared Architectural Pattern Is Emerging

ANTHROPIC: AUTO MODE FOR CLAUDE CODE

Released March 2026. Multi-tier authorization pipeline evaluates every tool call before execution. Deterministic rules handle safe actions, a model-based classifier handles everything else.

ROOST: OSPREY + GPT-OSS-SAFEGUARD

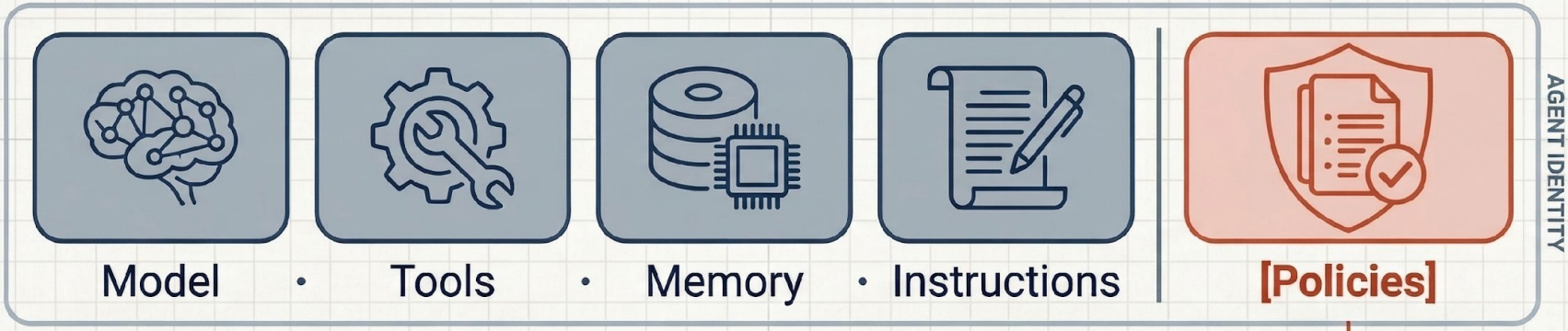
Osprey: real-time rules engine, built at Discord, in production at Bluesky and Matrix.org. gpt-oss-safeguard: open-weight model from OpenAI for policy-based classification with a “bring your own policy” approach.

- Intercept at the action level.
- Evaluate before execution.
- Layer deterministic rules with model-based evaluation.
- Keep the evaluator independent from the actor.

We are building it as a horizontal authorization layer, designed to work across agent frameworks.

OUR VISION

Policies Belong in the Agent's Definition



What the agent can and cannot do. Explicit. Inspectable. Testable.

DENIED

Authorization for Agent Behavior. Built at the Host Layer.

> Open integration: Claude Code, OpenClaw and any agent framework.

ALPHA PROGRAM OPEN

Onboarding available for early users and design partners.

`alpha.denied.dev | Use code: MCP-NYC`

Lorenzo Verna & **Pietro Valfrè**
Co-founders, denied.dev