Govern what your agents do — not just what they say.

An **Agent Security Kernel (ASK)** runs **inside** your agent and checks every action — each model call, tool call, data read, and code run — against the full path of the task so far, in your own environment, before it executes. Allow, warn, or block in under a millisecond. **Every agent has to ASK before it acts.**

THE RISK YOUR STACK CAN'T SEE

Identity, content guardrails, and AI gateways each judge one action in isolation. The breaches that matter are **legitimate actions in a dangerous sequence** — read a customer record, then send it outside the org. Every step passes every stateless check. **The path is the breach.**

```
policies on paths — same action, two histories

# clean path — nothing sensitive was read
step.model    POST                          ALLOW
step.resource GET product/info              ALLOW
step.message  POST                          ALLOW

# exfiltration path — PII read, then external send
step.model    POST                          ALLOW
step.resource GET crm/customer [pii]        ALLOW
step.message  POST                          BLOCK
```

The same outbound message — allowed on one path, blocked on the other. A stateless, per-call check can't do this: it doesn't know what came before. Path-dependence is the whole point.

THREE VERDICTS, ONE LAYER

- **ALLOW** — Proceeds — recorded as a tamper-evident log entry.
- **WARN** — Proceeds, but flags the step for review.
- **BLOCK** — Stopped **before** it executes, deterministically.

Human approval isn't a fourth verdict — it's a **gate step** a policy can require on the path: e.g. `code execution requires a preceding gate`.

WHAT MAKES IT AN ASK

Complete mediation
Every action is checked before it runs — not a sampled subset.

Path-dependent
Decisions use the full task history, not the single step. A PDP that remembers.

Deterministic
Policy, not an LLM judging an LLM. Same input, same verdict, every time.

In-process & sovereign
No proxy. Only the policy crosses the boundary; prompts and data stay put.

WORKS WITH THE FRAMEWORKS YOU ALREADY RUN

Python SDK | LangChain | LangGraph | Claude Code | REST / local engine | Custom templates | + any runtime

< 1 ms	In use	Tamper-evident	AI Act evidence
per step — 296µs p99, 100 policies & 50-step history	NL financial services, insurance & healthcare	hash-chained log of every action & verdict	runtime records supporting Art. 9 · 12 · 14 · 15

Free proof-of-value in your environment.

We deploy the kernel alongside one of your agents, write policies on paths following your internal rules and AI Act obligations, and hand you **a runnable enforcement layer plus a report for your CISO, legal team, or auditor**. Your infrastructure, your data, no further commercial commitment.

jeroen@kyvvu.com

PARTNER-DELIVERED · YOUR DATA STAYS WITH YOU

BUILT ON

Runtime Governance for AI Agents: Policies on Paths — Maurits Kaptein, full professor Eindhoven University of Technology; co-author *AI Agents at Work*.

INSPIRED BY

Zero Trust for AI Agents — the framework Anthropic published for securing autonomous agents (May 2026). The Agent Security Kernel is a privacy-first, in-process realization of it. claude.com/blog/zero-trust-for-ai-agents

No tool makes an agent perfectly safe — that's the "useless-agent" corner. Safety is a frontier between usefulness and risk; an ASK moves your frontier outward, along the one axis nothing else covers: the behavioral path.

arXiv 2603.16586

kyvvu.com

The kernel runs in your process.

Policy evaluation is pure CPU — **no network calls, no database queries, no I/O on the hot path.** Policies are pulled from your own git and cached; the engine keeps enforcing on cached policies even if the control plane is unreachable. The behavioral trace is shipped to your SIEM platform; agent identities come from your IAM.

`Your agent code`	`@kv.step` · LangChain handler · REST — emits each action as a behavior (`step_type, scope, verb`)
`kyvvu SDK`	Translates events to behaviors via your YAML templates; **pulls policy manifests from your git.**
`kyvvu-engine`	**In your process.** Evaluates the action against the full task history — allow / warn / block — in sub-ms, before the step runs
`Your platform / SIEM`	Hash-chained trace flushed on task completion via `KV_LOG_LOCATION` · `KV_LOG_FORMAT` — OTLP to Datadog & the like, or HTTP / file, or `metadata_only` to redact.

IN-PROCESS

The engine runs **inside the agent**, not on the wire. No proxy, no added hop, no perceptible latency.

POLICY-AS-CODE, IN GIT

Policies are **YAML manifests in your repo** — versioned, reviewed, assigned per agent.

LOGS TO YOUR PLATFORM

The trace lands in **your** SIEM / observability stack. Nothing routes through us.

DATA STAYS WITH YOU

Only the **policy** crosses the boundary. Prompts, tool I/O, and customer data never leave.

OPEN CORE

SDK & integrations **Apache 2.0**; engine **BSL 1.1**, source-available. Auditable, no black box.

AI ACT EVIDENCE

The **hash-chained trace** is a tamper-evident record of every action and verdict, ready for audit.

The SELinux for AI agents.

SELinux is the minimal, trusted layer that enforces policy on every operation a process attempts. An ASK is the same idea for agents: a small, verifiable, deterministic core that mediates every action and **bounds the space the agent is allowed to operate in**. The security kernel decides the space; your platform and the OS jointly enforce it. This is a runtime implementation of the core idea of **Zero Trust for AI Agents** — the kernel-level discipline of SELinux applied to autonomous, machine-speed action.

HARDENED FOR PRODUCTION

FAIL-CLOSED

High-risk agents **block every action** when no policy is loaded — fail-closed, never fail-open.

`KV_POLICY_FAIL_MODE=closed`

SURVIVES OUTAGES

Policies are **cached to disk** and keep enforcing through a control-plane outage or restart.

`KV_POLICY_CACHE_PATH`

TAMPER-PROOF POLICY

HMAC-signed fetch — the engine rejects any policy not signed by your control plane.

`KV_POLICY_HMAC_SECRET`

Agents ASK before they even begin. At startup, each agent must declare its purpose, tools, and risk class — registration is validated before the first step is allowed to run.

BUILT FOR THE EU AI ACT

Every action is intercepted, recorded, and enforced at the step level — so the kernel produces the runtime **evidence** the Act asks for. A mechanism, not a document.

ART. 9
Risk management
Per-agent risk class; policies act as continuous runtime controls.

ART. 12
Record-keeping
The hash-chained trace logs every action automatically.

ART. 14
Human oversight
Gate steps put a human on the path; warn / block surface for triage.

ART. 15
Accuracy & robustness
Deterministic enforcement, fail-closed, signed policy integrity.