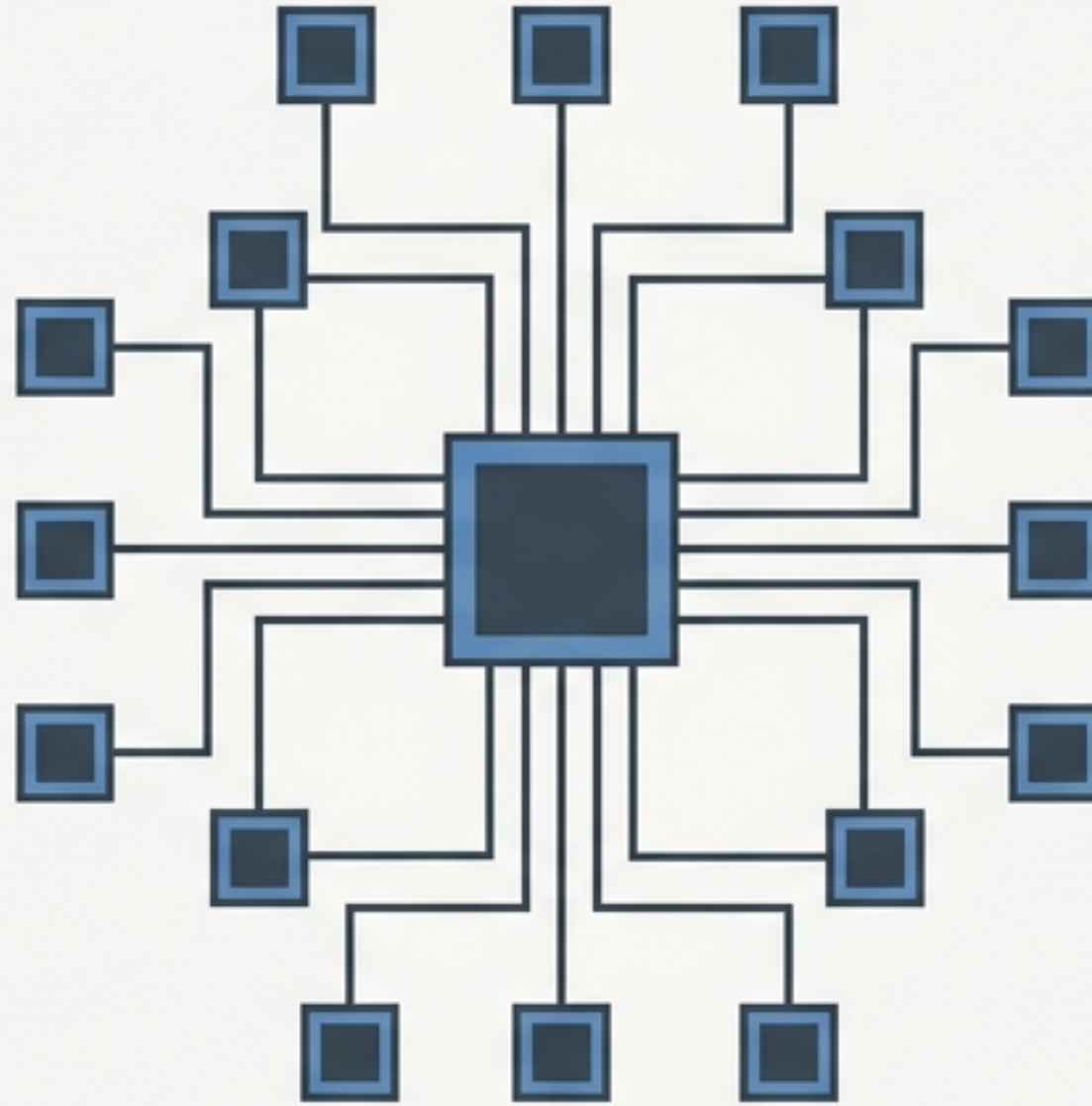


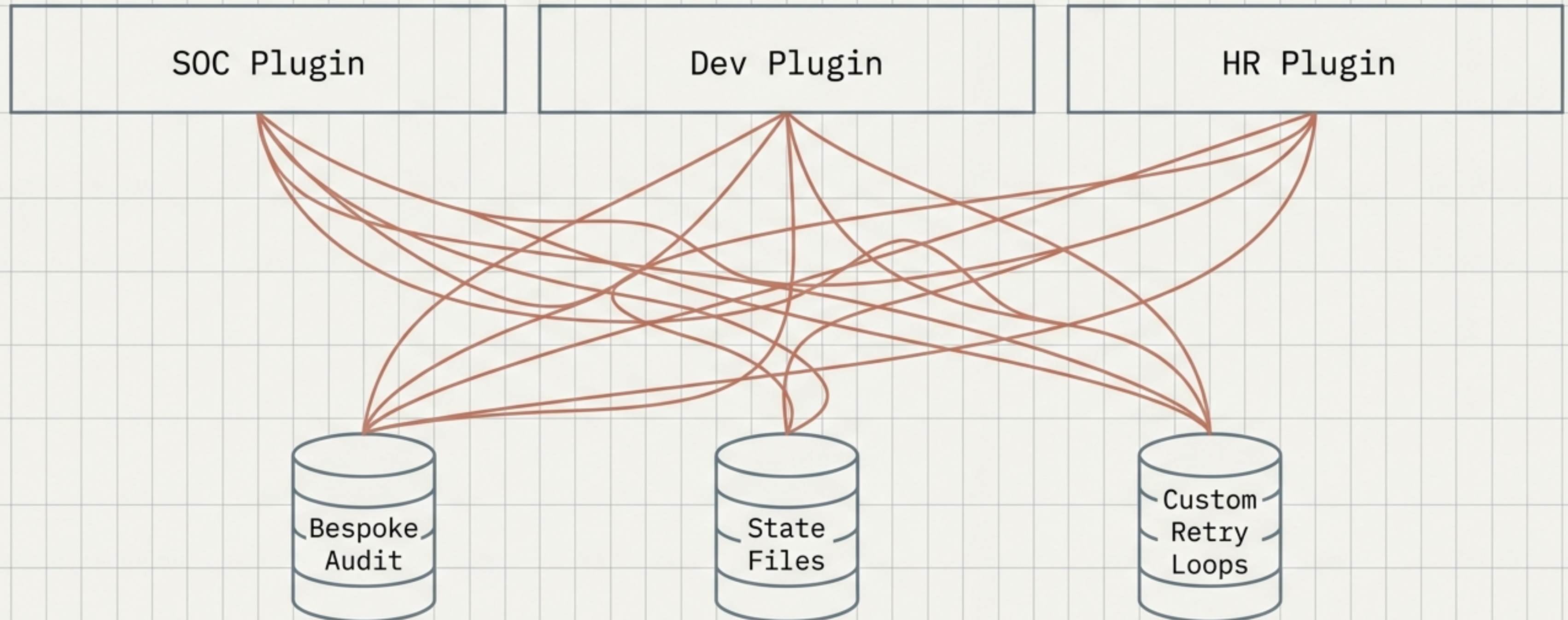


bulletproof-conductor-kernel

Domain-Agnostic Orchestration for Claude Code Multi-Agent Workflows.

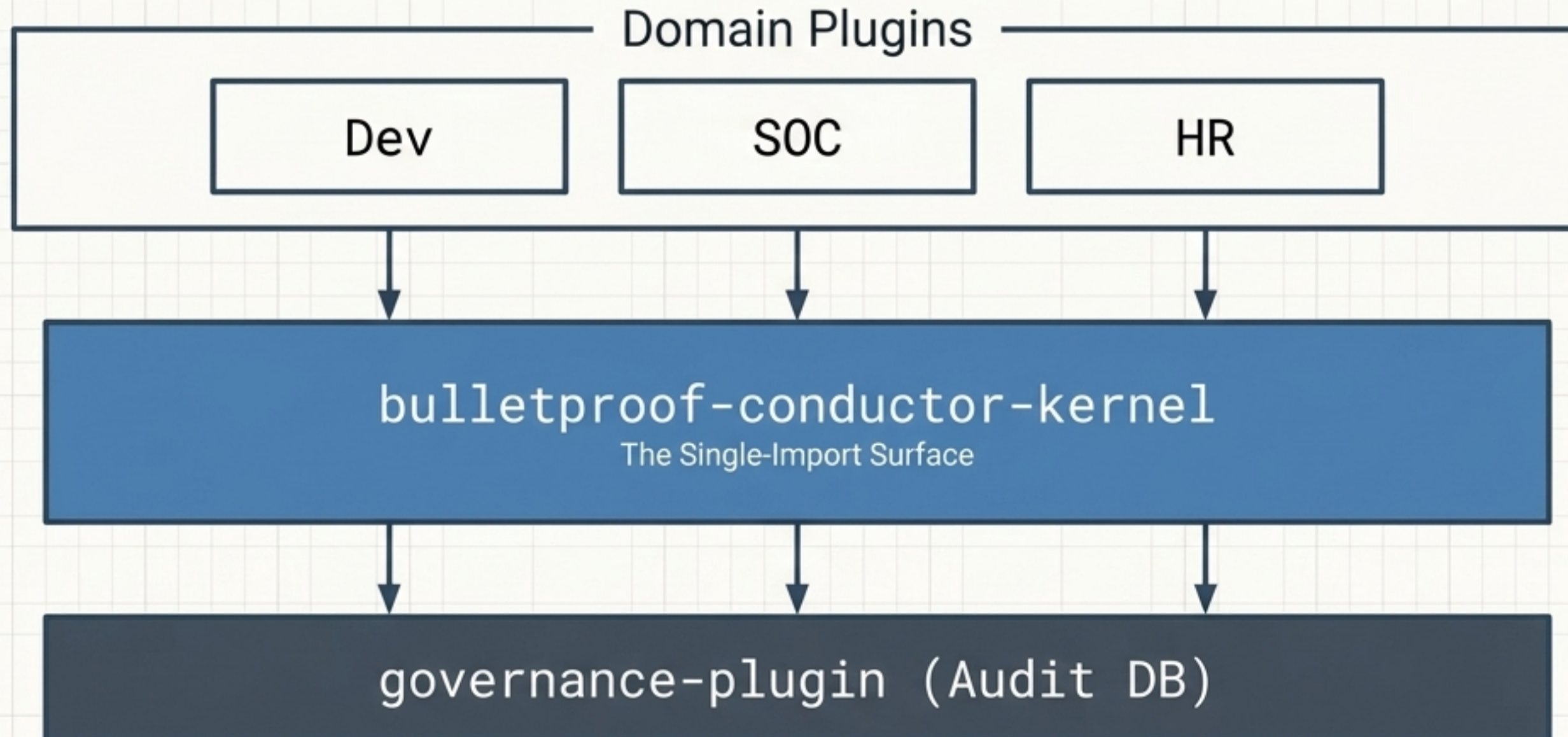
The multi-agent tax limits domain focus

Without a unified orchestration layer, every domain plugin ends up reinventing the same five foundational systems. Developers spend **80%** of their time building state machines and audit logs instead of investigating SOC alerts or writing code.

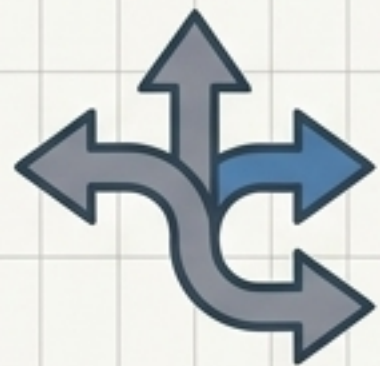


Beneath the domain, above the governance

bulletproof-conductor-kernel is a Claude Code plugin with zero runtime dependencies and no slash command of its own. It is an engine consumed entirely by downstream domain plugins.



Solving the five orchestration concerns permanently



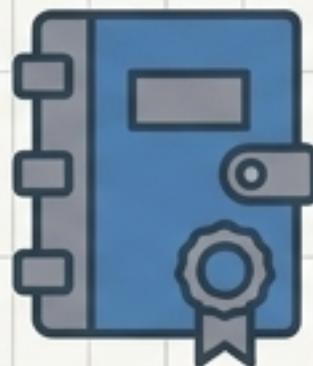
Dispatch
contract

Budgets,
retries, model
delegation



Validation
loop

Second-model
verification



Audit
trail

HMAC-signed,
append-only
records



State
machine

Cross-session
persistence



Gate
enforcement

Human
approval
barriers

19 domain-agnostic agents ready for dispatch

Validation

critic

gemini-validator

completeness-validator

checkpoint

event-router

Outcome & Reflection

outcome-collector

retrospective

prediction-engine

research

Code Investigation

bug-find

analyze-codebase

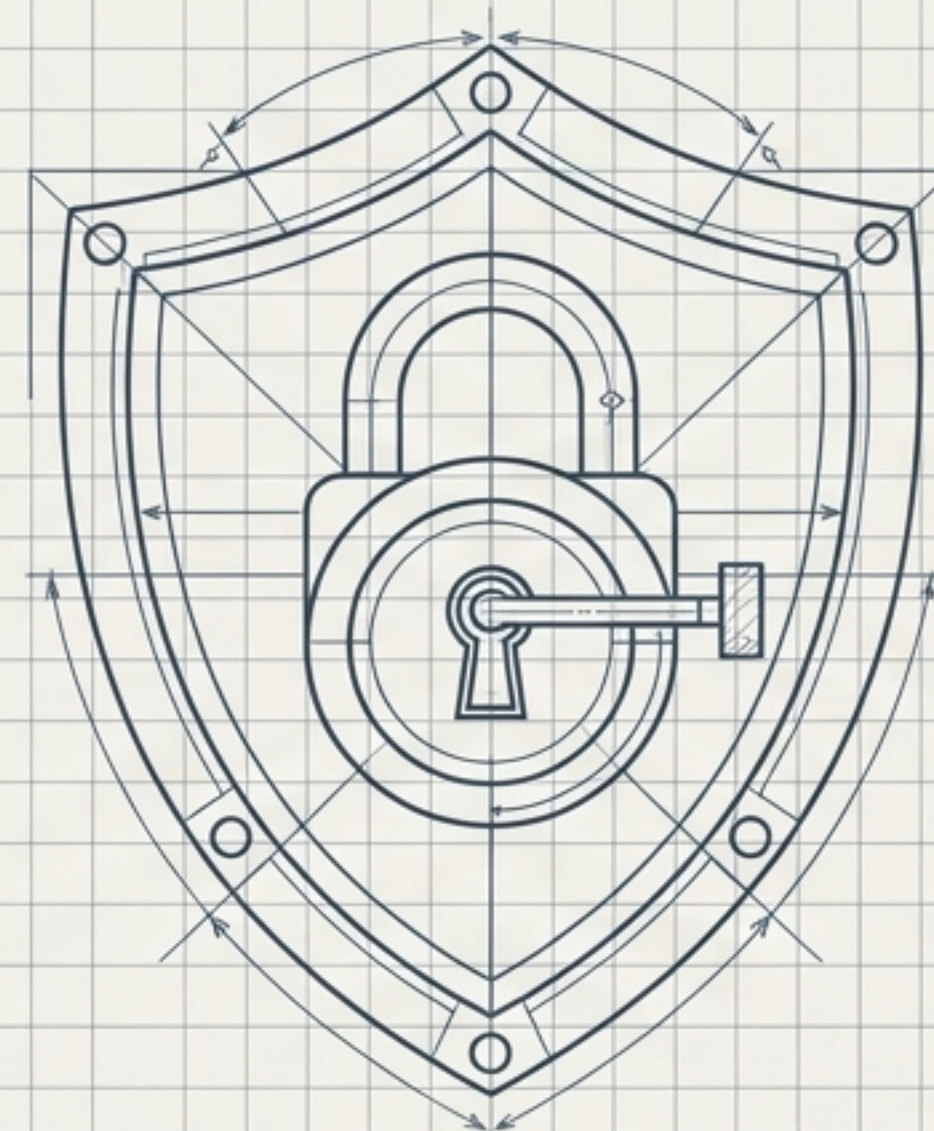
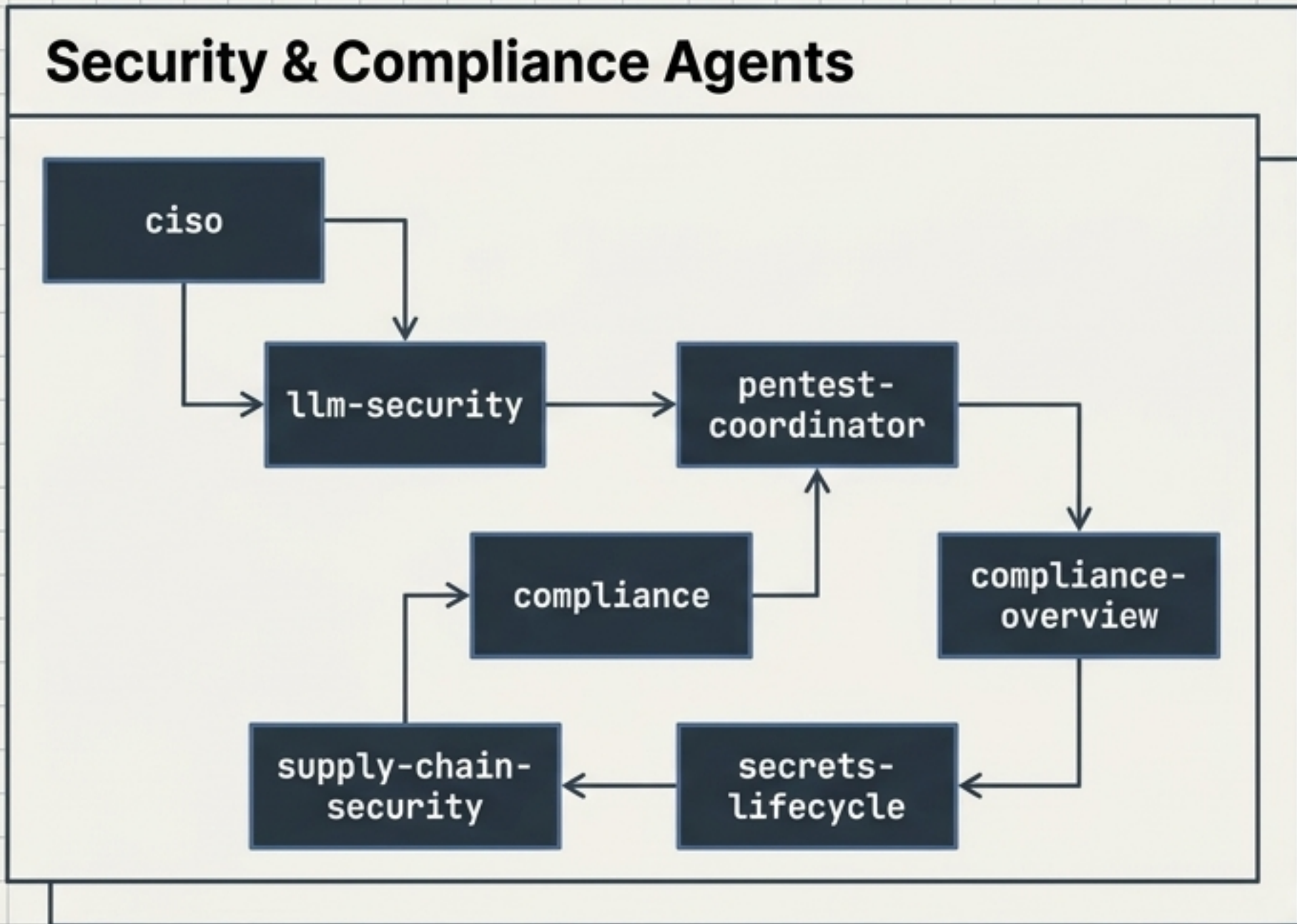
Recovery

recovery-engine



Constraint Note: Kernel agents are read-mostly and never dispatch other agents. The Task tool is explicitly excluded from their allowlists. Only domain plugins drive orchestration.

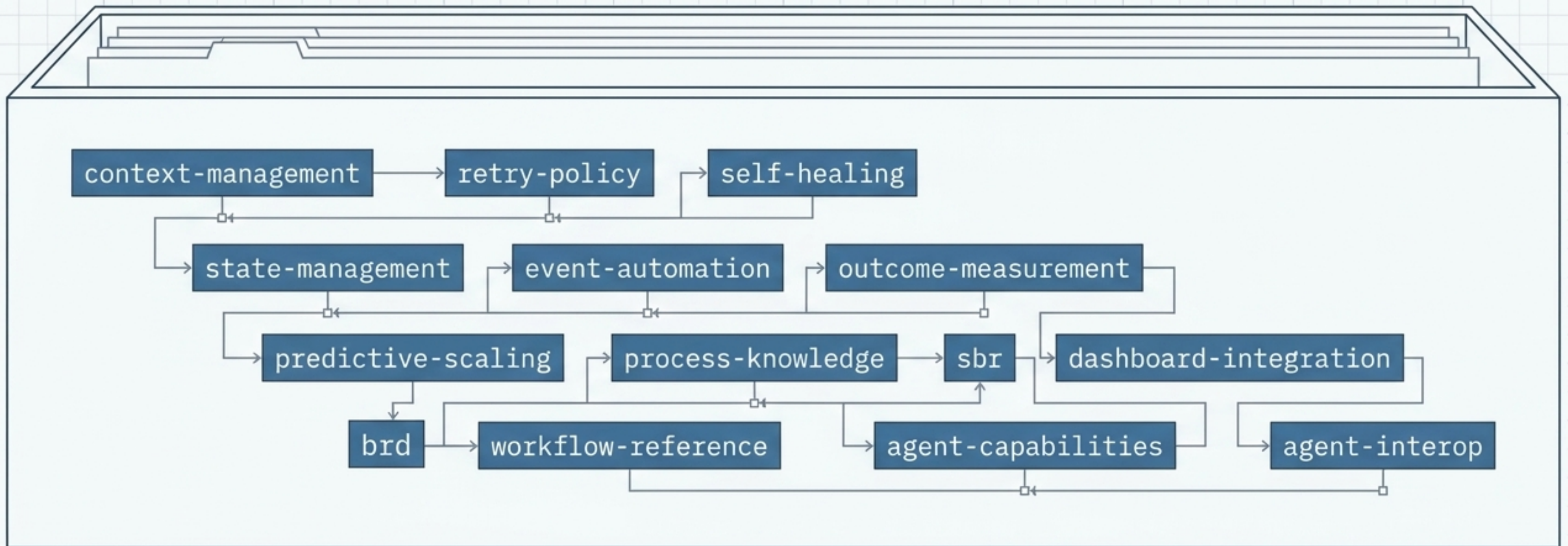
Built-in security and compliance muscle



7 of the 19 kernel agents are dedicated entirely to security analysis and compliance mapping. Security is embedded directly into the orchestration layer, not bolted onto the domain logic.

14 prose skills encoding tier-appropriate patterns

Skills in the kernel are not executable code. They are prose guidance surfaces loaded on demand by Claude Code to instruct agents on how to behave according to the workflow's classified tier.

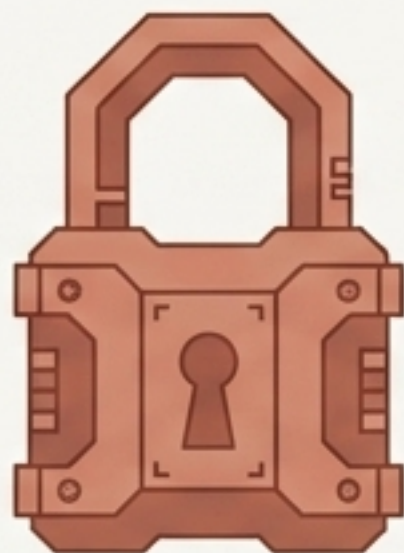


Two distinct modes of execution

	Workflow Mode	Stream Mode (Experimental)
Dimension 1: Lifecycle	Bounded task, start to finish (e.g., build a feature).	Continuous, event-driven subscription (e.g., webhooks, crons).
Dimension 2: State Persistence	Working-directory JSON file .	Qdrant collection keyed by stream_id.
Dimension 3: Schema Contract	v3.0, flexible phase-dependent fields.	v3.0, strict additionalProperties: false rejection at validation.

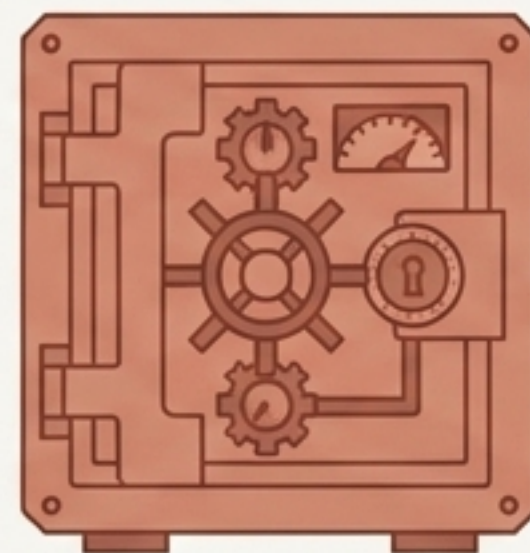
Amplified risk requires amplified guardrails

Because **Stream** mode operates continuously, the kernel contract strictly enforces two preconditions before any event is processed:



Requirement 1

Authentication is mandatory.
``authentication.kind != "none"`` is enforced prior to schema validation.



Requirement 2

Budget initialization is mandatory.
Prevents denial-of-wallet attacks on long-running subscriptions.

The dispatch primitive and prompt isolation

Domain Agent calls `dispatch_agent`

Path A: Trusted Prompts

Uses standard string form for developer-authored instructions.

Path B: Untrusted Content

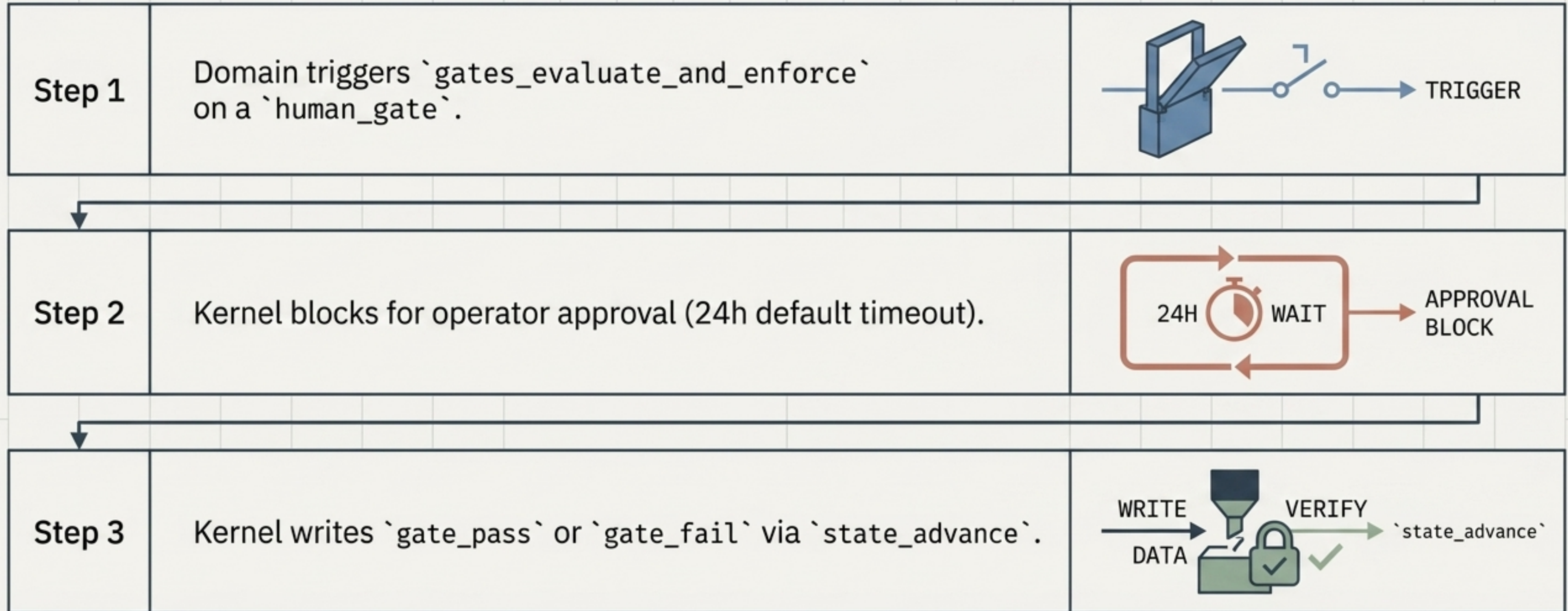
Logs, user inputs, and external alerts **MUST** use the envelope form.

Envelope Boundary

`<UNTRUSTED:label>...</UNTRUSTED:label>`

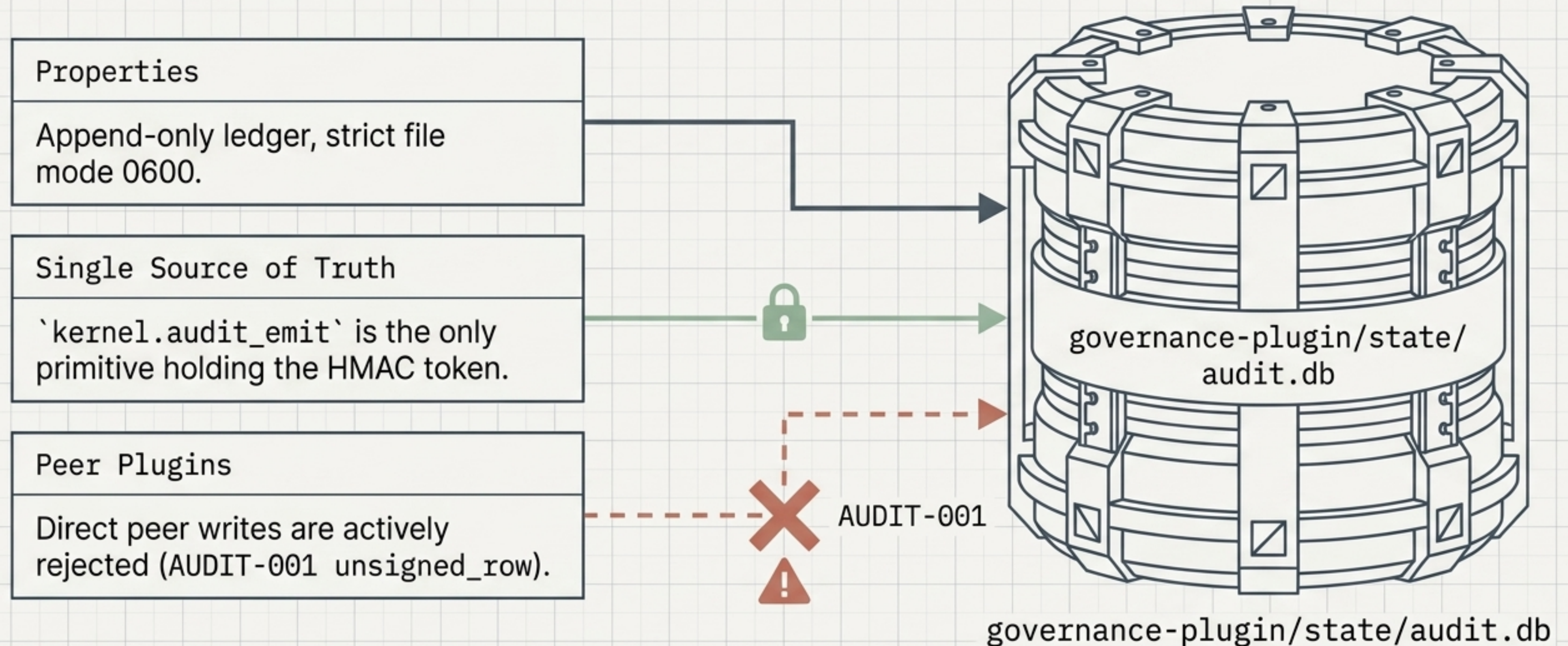
The Envelope Boundary: The kernel wraps untrusted regions with a fixed preamble, isolating instructions from injected data. Exceeding budgets yields KER-DA-005.

Gate enforcement and immutable state mutation



Enforcement: All state mutations must use ``state_advance``. Direct JSON edits are rejected by the ``post-state-write.sh`` hook. Domain code cannot bypass human governance.

Absolute auditability via cryptographic isolation



Default-deny egress boundaries

Local Operator Environment

Memory
(local Qdrant via Docker)

Audit DB
(0600 file)

gemini_validate

Google
Gemini API

Local by Default: Memory and Audit never leave the machine.

The Single Egress Exception: option: gemini_validate (for second-opinion validation).
Operator Control: By default, gemini_validate refuses regulated data unless the operator explicitly sets an operator_override.

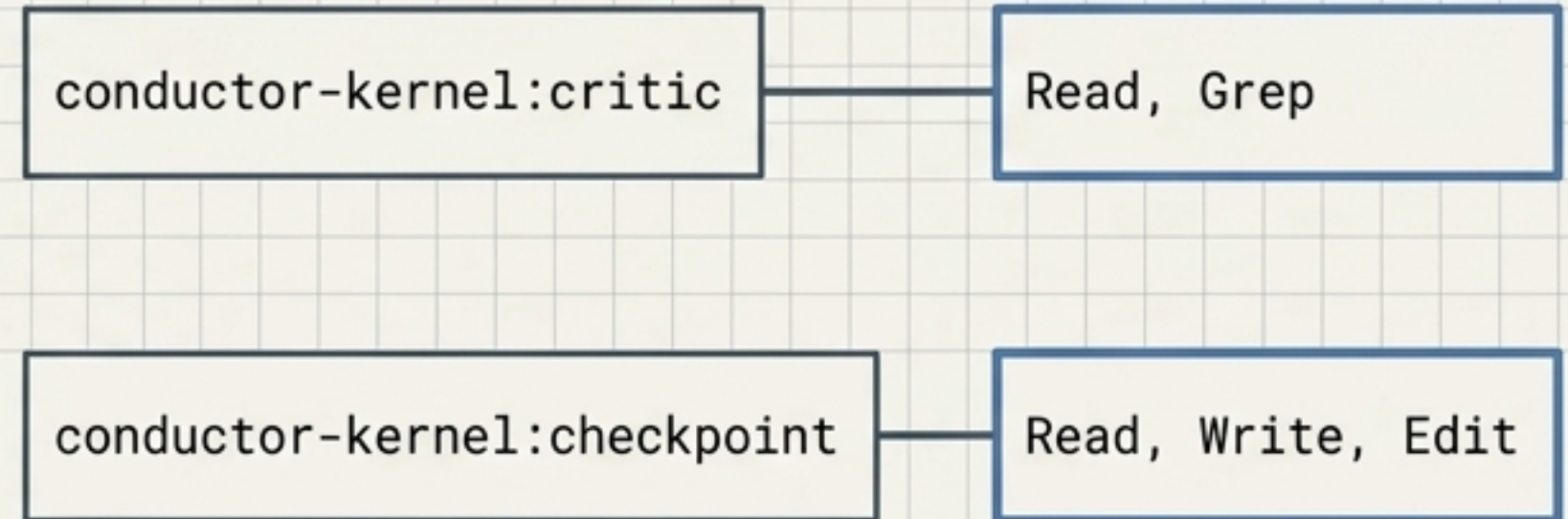
Hardened agent surfaces and minimal tools

Enforcement



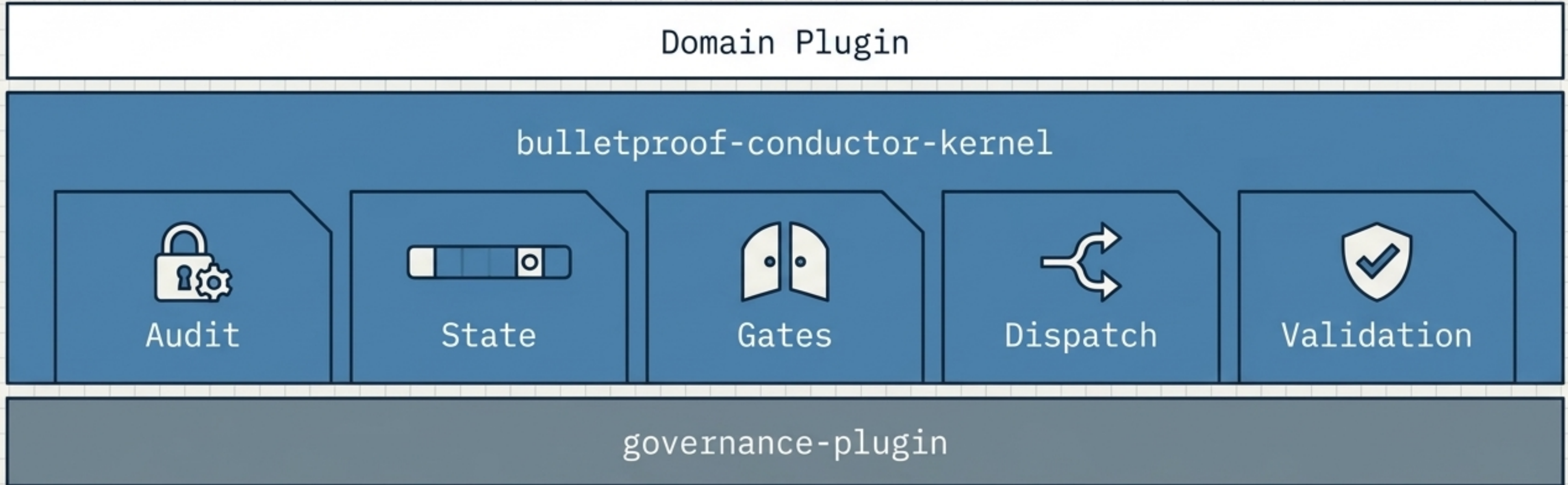
Enforcement: Every agent declares an explicit, minimal allowed-tools frontmatter. Widening this fails the CI build.

Example Allow-lists



Strict Boundary: No kernel agent has access to the Task tool. They cannot self-orchestrate or spawn unapproved sub-agents.

Focus on the domain logic. The kernel handles the rest.



By depending on `bulletproof-conductor-kernel`, domain developers no longer build bespoke state machines, validation loops, or audit logs. You declare the domain logic; the kernel guarantees it is executed securely, durably, and transparently.