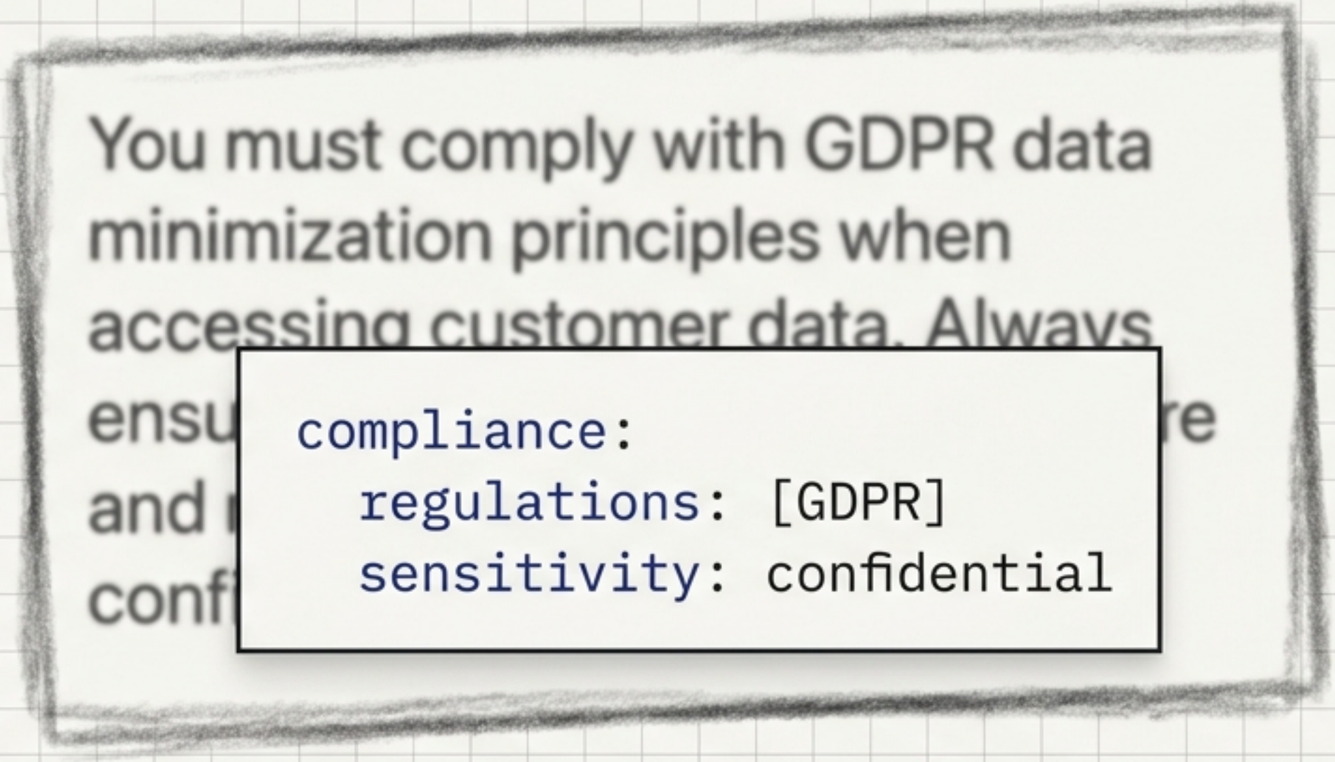


From Policy to Practice

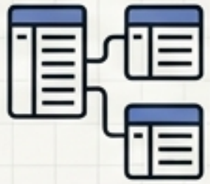
Making Regulations Machine-Readable for AI Agents.

The Infrastructure Mindset

Prose is for humans. Agents need data.

You do not put your database schema in comments or your API contract in a wiki page.

Why would you put your compliance policy in a system prompt?



Database Schema → Migrations ✓

Infrastructure → Terraform ✓

Compliance Policy → System Prompts? ✗

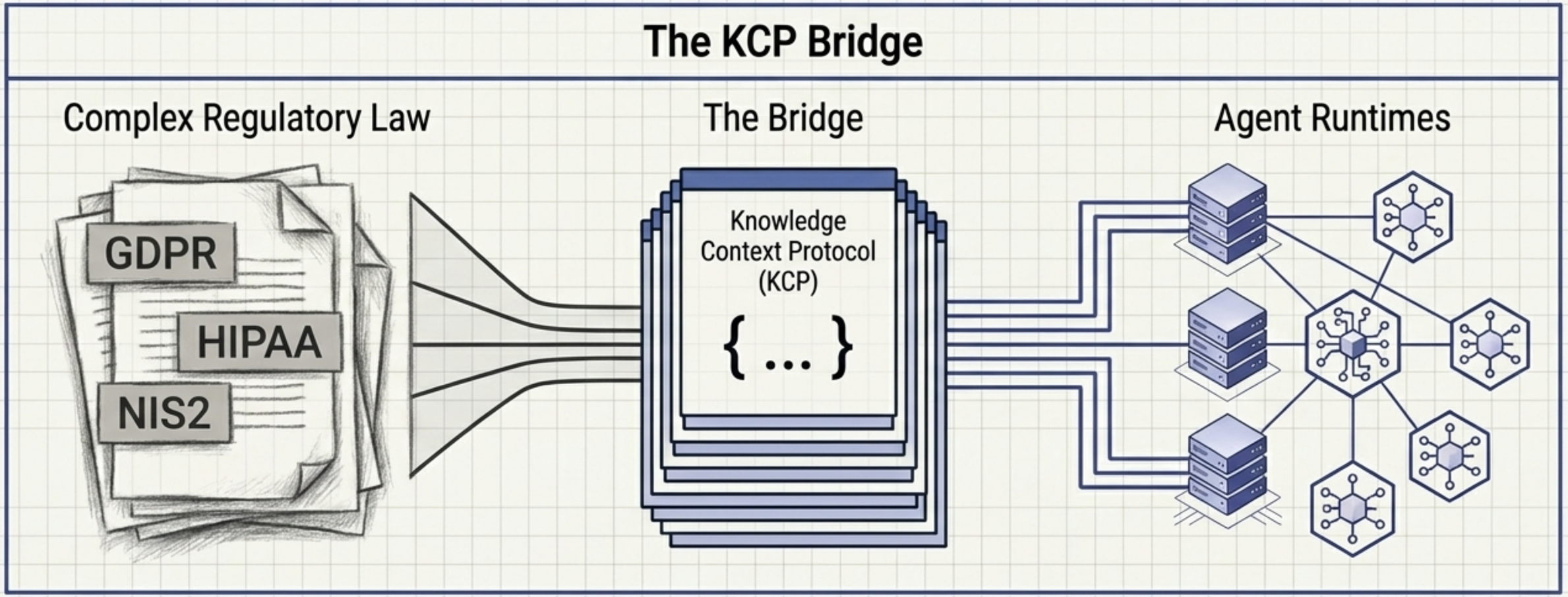
The Three Broken Patterns of AI Compliance

Pattern	Mechanism	Fatal Flaw
System Prompt Stuffing	Natural language instructions ("Always follow GDPR").	Competes for context window. Invisible to auditors. Cannot be evaluated programmatically.
Post-Hoc Review	Human or LLM reviews output after execution.	The damage is already done. Misses invisible API calls in tool invocations three layers deep.
Periodic Audit	Scrambling for logs quarterly to explain agent decisions.	Unscalable. Works for 10 requests/day, fails completely at 10,000.

The root cause: all three encode policy as prose.

Declarations as Data

KCP is a versionable, machine-readable declaration of what knowledge exists and what constraints apply. Auditors read the manifest, not your agent’s source code.



Anatomy of a Compliance Block

```
compliance:
  regulations: [GDPR]
  regulations: [EEAPIA]
  data_residency: [EEA]
  sensitivity: confidential
  rensitivits: [no-external-llm,
  restrictions: [no-external-llm, audit-required]
```

Defined Vocabulary

Maps directly to real law (GDPR, NIS2, HIPAA). Unknowns silently ignored.

Geographic Boundaries

Constrains where processing may occur.

ISO 27001 Aligned

4-level classification (public, internal, confidential, restricted).

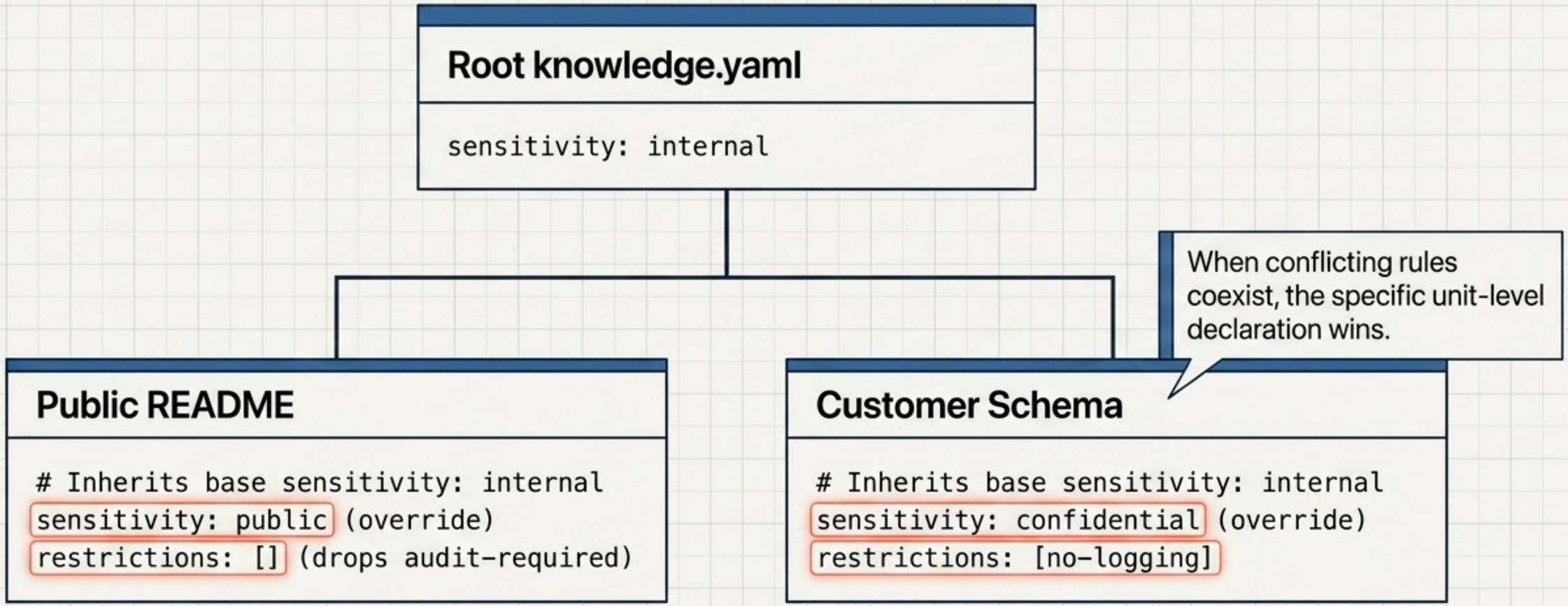
Processing Constraints

Forces local models; requires rigorous logging.

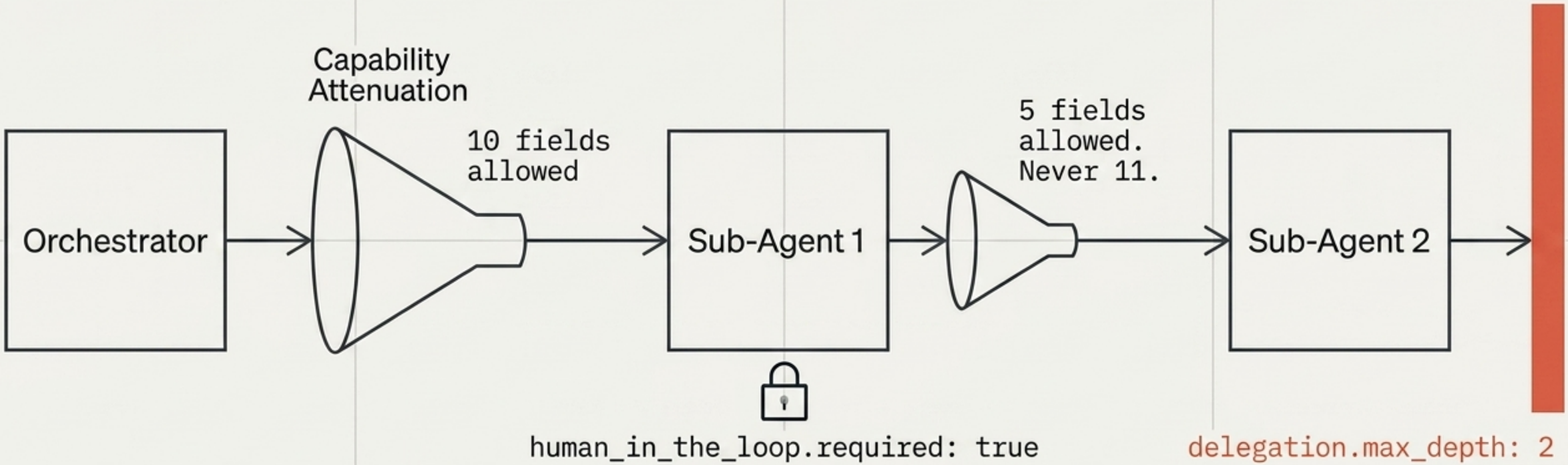
This is not a system prompt. It is structured metadata.

Inheritance & Overrides

Borrowing the CSS mental model: root blocks provide defaults, unit blocks provide specifics. Mixed-sensitivity data coexists seamlessly in a single knowledge base.



Trust & Delegation: Pre-Negotiated Constraints

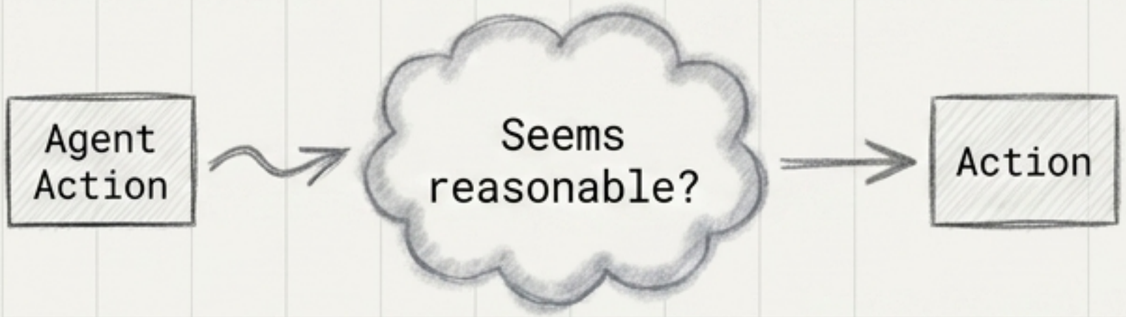


Each hop must mathematically narrow permissions.
Static limits prevent invisible scope creep.

The Missing Layer: Programmatic Evaluators

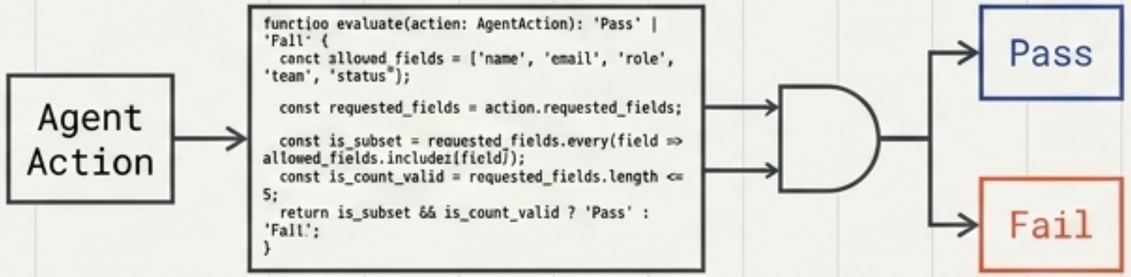
The evaluator is code, not LLM judgment. It does not ask a model if access is “reasonable.” It compares two lists. Pass or fail.

Prompt-Based “Compliance”



Fuzzy, invisible, unversioned.

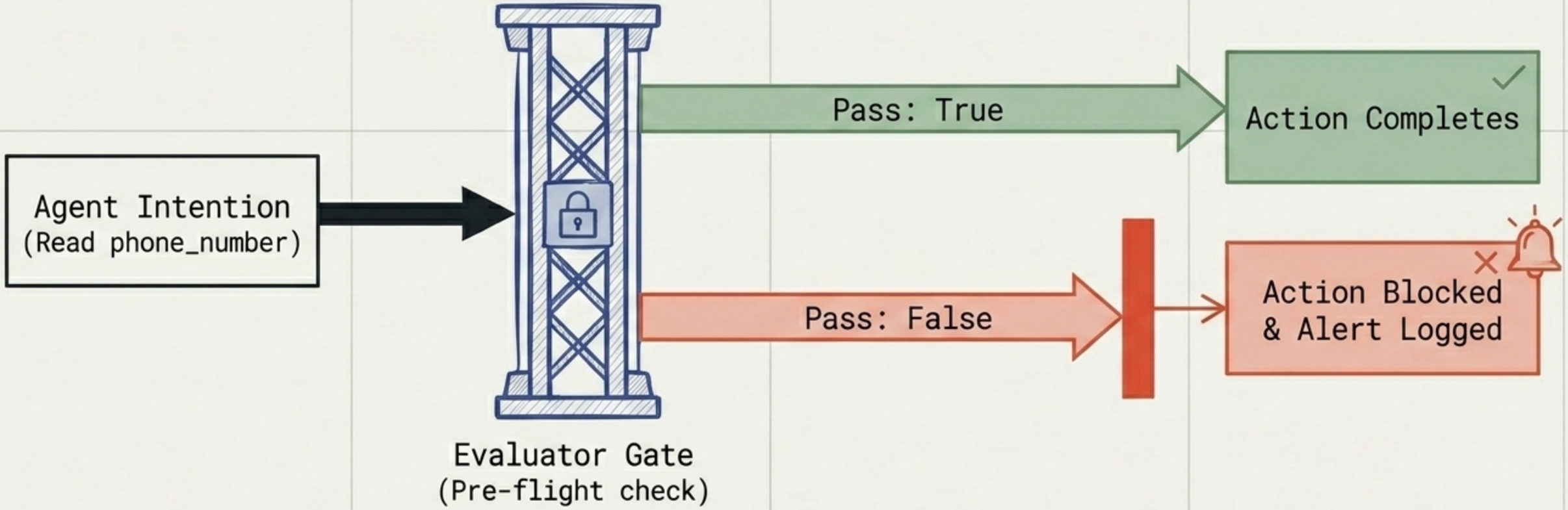
KCP Evaluator Pattern



Deterministic, binary, CI/CD testable.

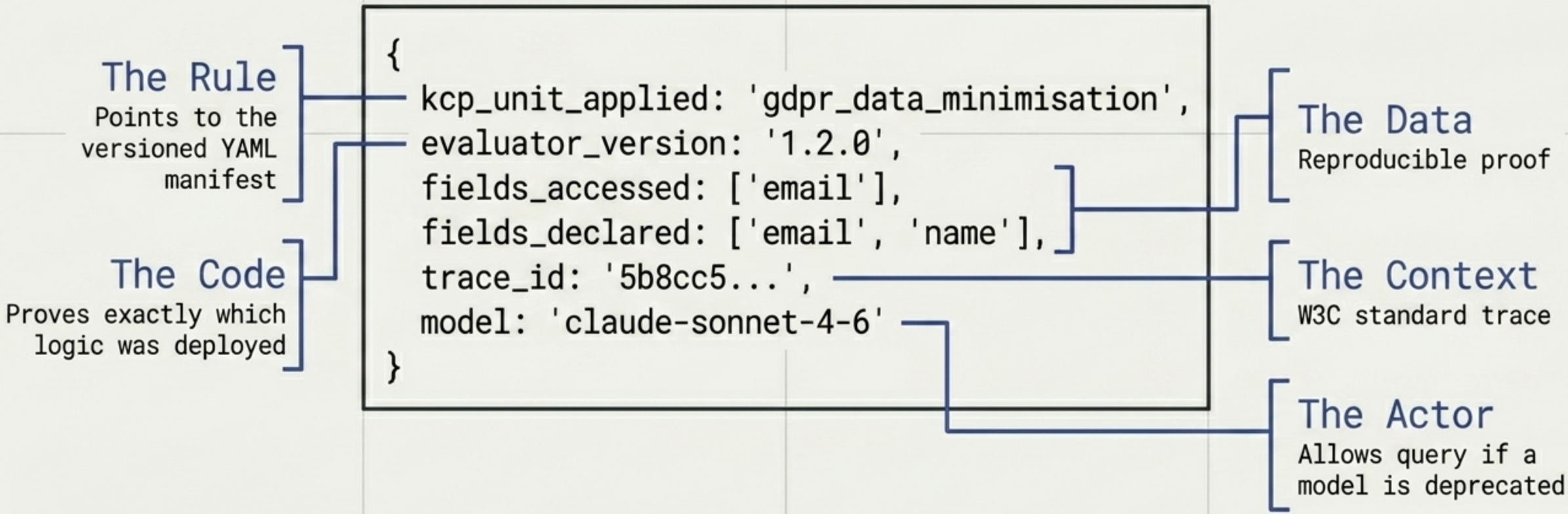
The Evaluator as a Deterministic Gate

Not a review. A deterministic gate. It blocks the action before the damage is done. Versioned independently of the agent.



The Audit Ledger

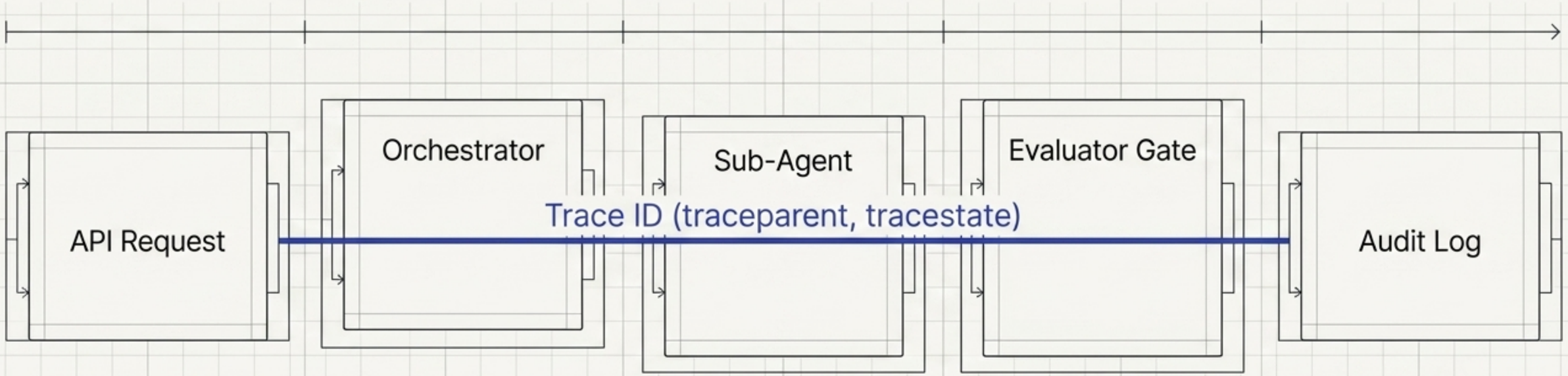
Not a dashboard. A ledger. Any auditor can read this log and independently confirm that passed: true was mathematically correct.



Traceability Across Hops

```
trust.audit.require_trace_context: true
```

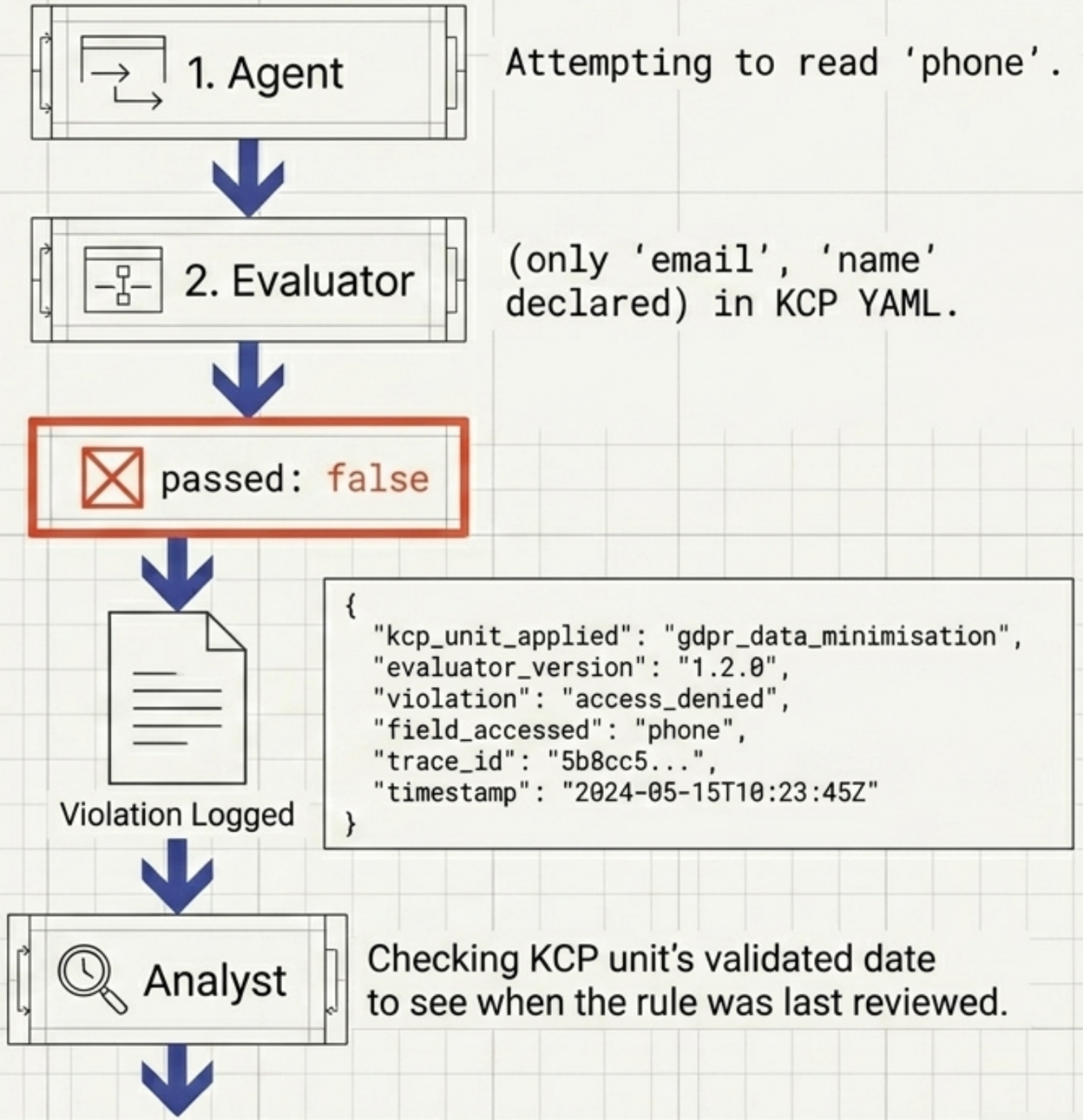
No guesswork required. Agents must include W3C Trace Context headers. The agent decision, delegation hop, and compliance log share a single immutable trace.



Incident Investigation

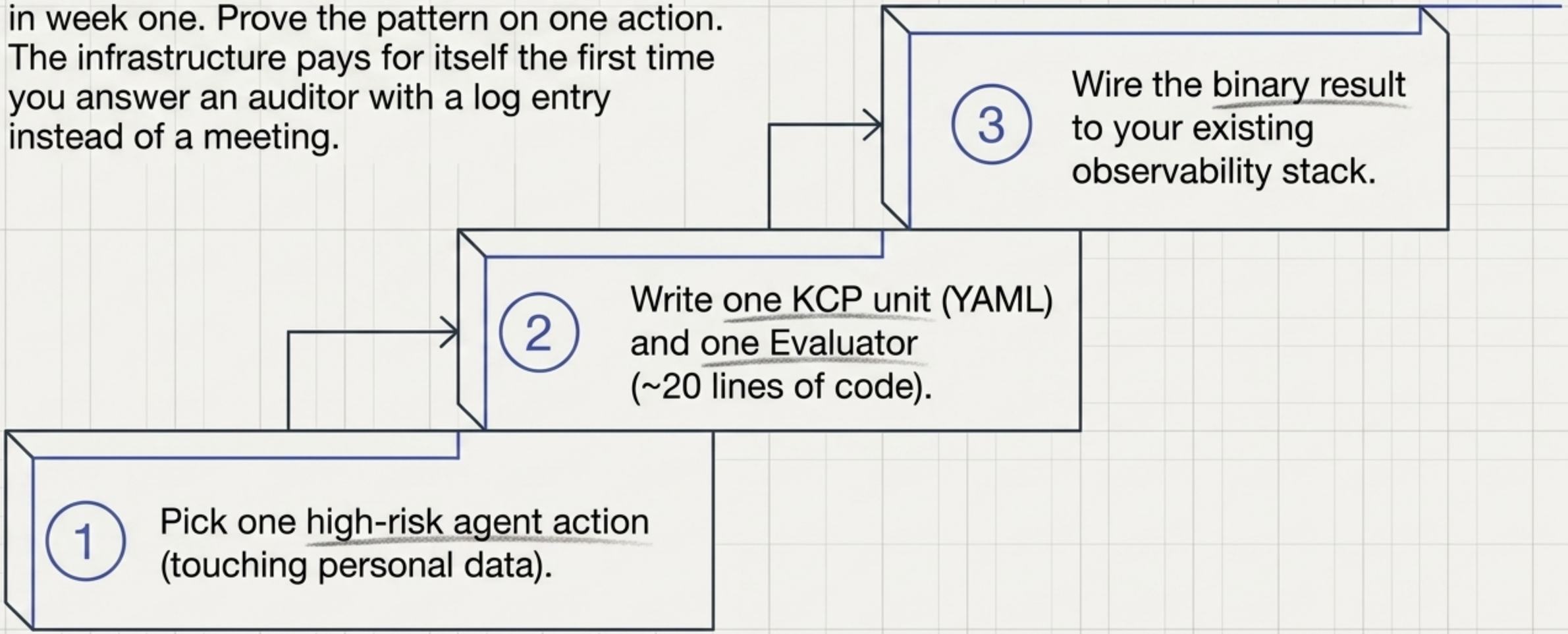
‘Logging doesn’t prevent wrong decisions. It makes wrong decisions impossible to hide.’

You know exactly where the failure originated.



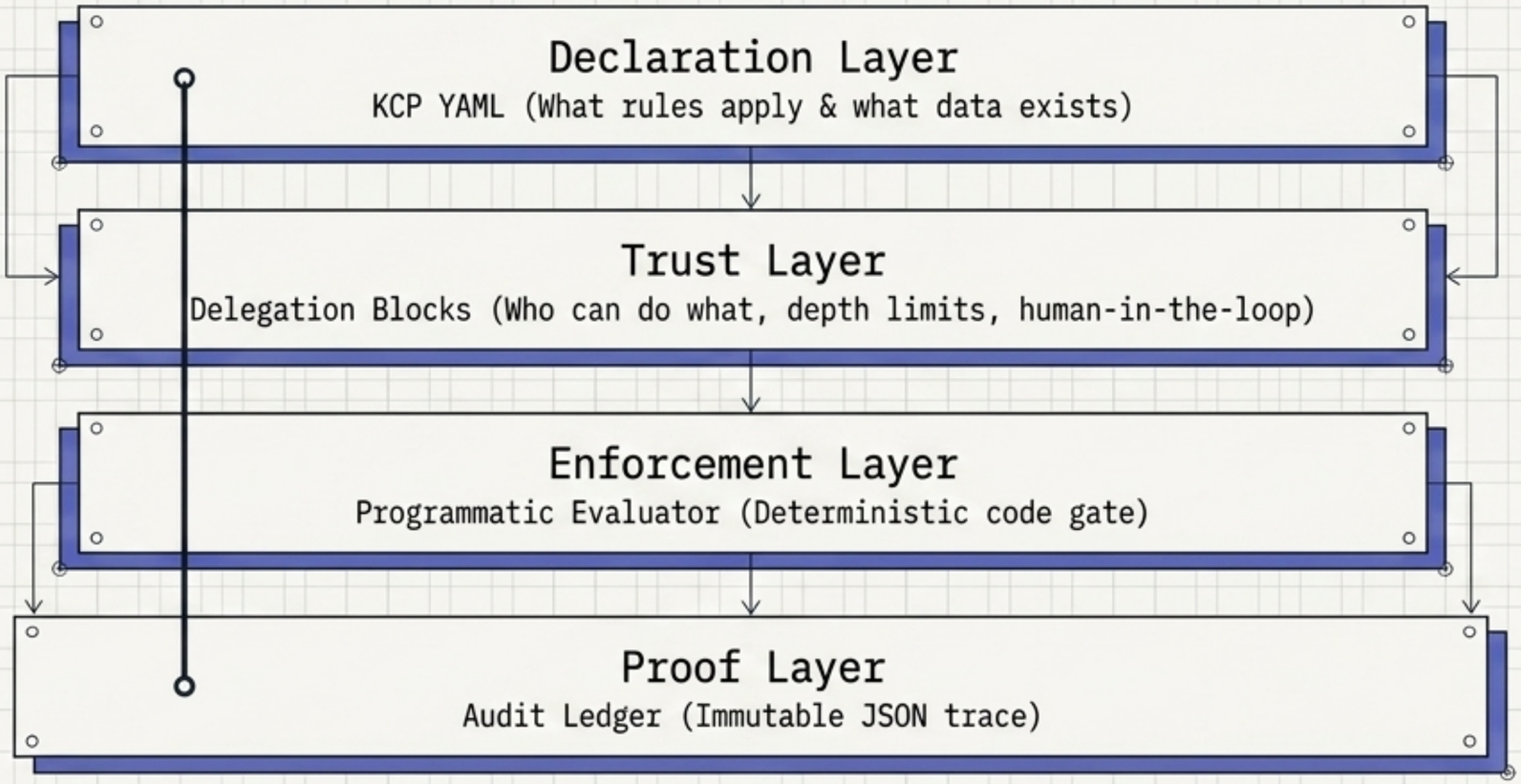
The Adoption Gradient

Do not model your entire regulatory framework in week one. Prove the pattern on one action. The infrastructure pays for itself the first time you answer an auditor with a log entry instead of a meeting.



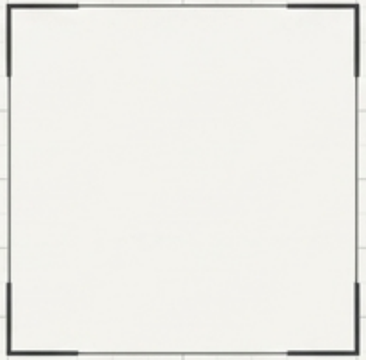
The KCP Infrastructure Stack

Synthesis:: The unified architecture for machine-readable regulation.
Independent components, versioned separately, working deterministically.



Boring passes audit.

Build the infrastructure.



github.com/Cantara/knowledge-context-protocol

(Apache 2.0 Spec & GDPR ROPA generator examples)