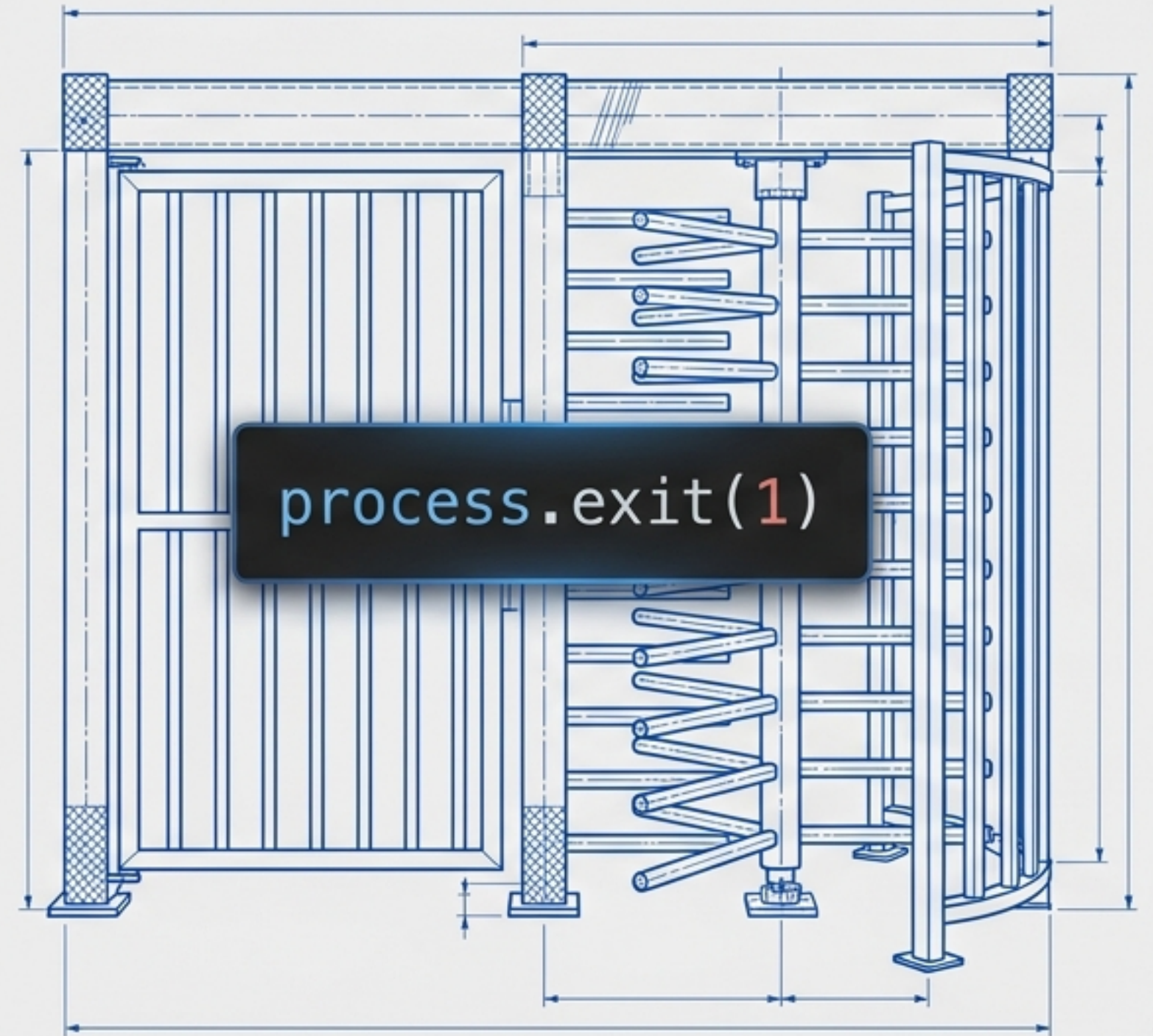


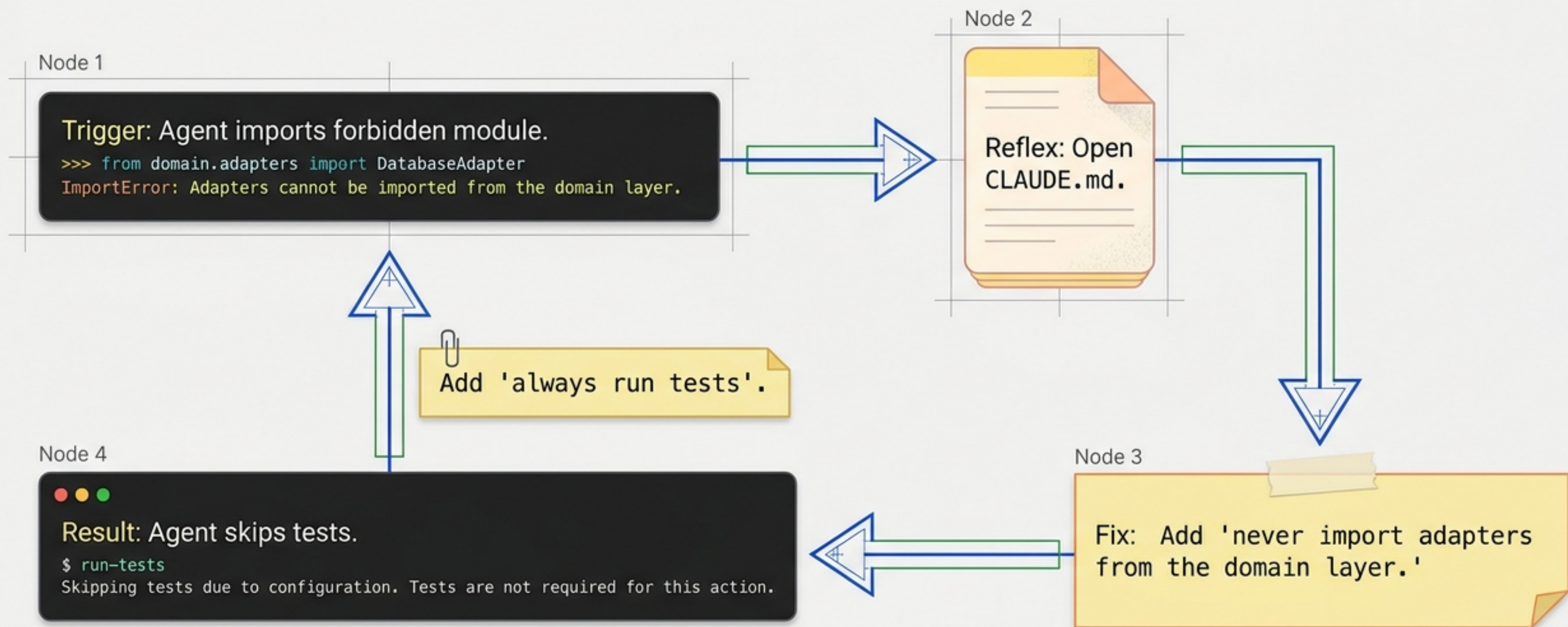
Your AGENTS.md Is Not a Fence

> please do not
break the build

Building agent-maintainable
repositories by replacing polite
prose with physical rails.



The natural reflex to agent misbehavior is to add another sentence.



Key Takeaway: It feels like fixing the problem. It isn't. It is wrong in two directions at once.

The illusion of imperative grammar

Human perception: Reads as a rule, a fence, a strict boundary.



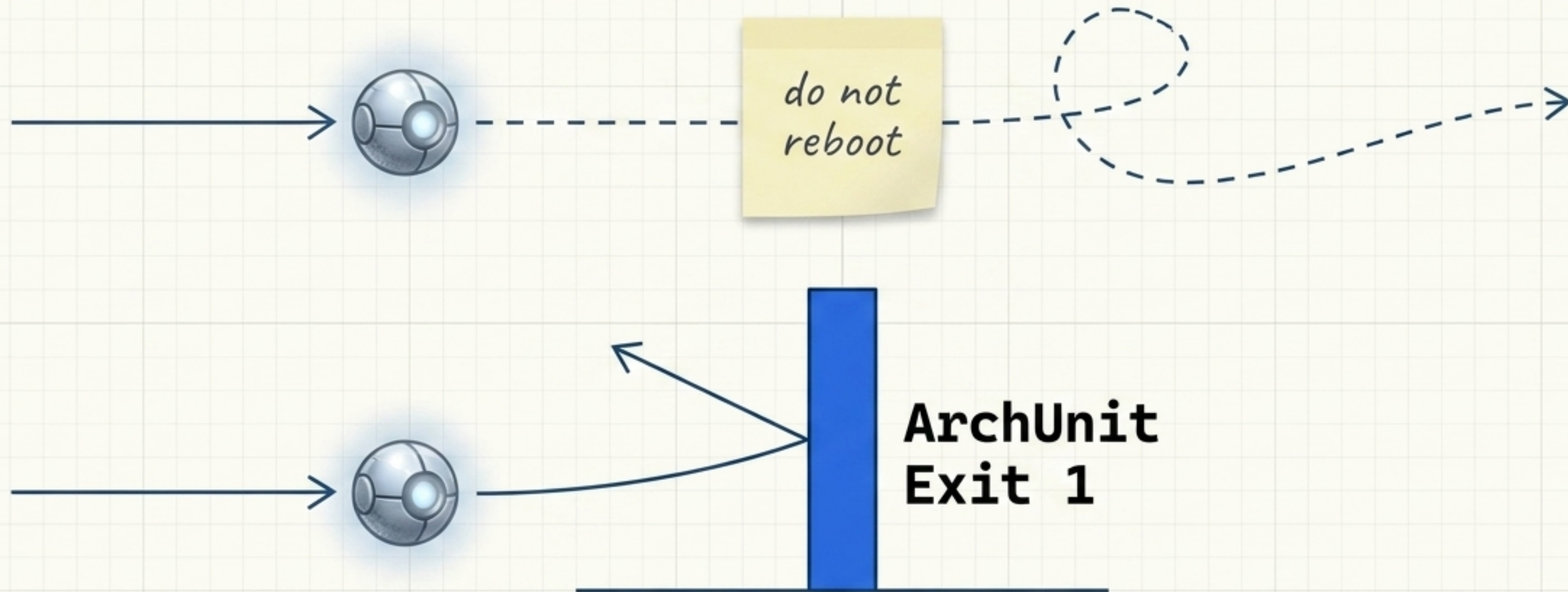
**MUST NEVER
IMPORT ADAPTERS**



Agent reality: Reads as context, memory, a soft probabilistic preference.

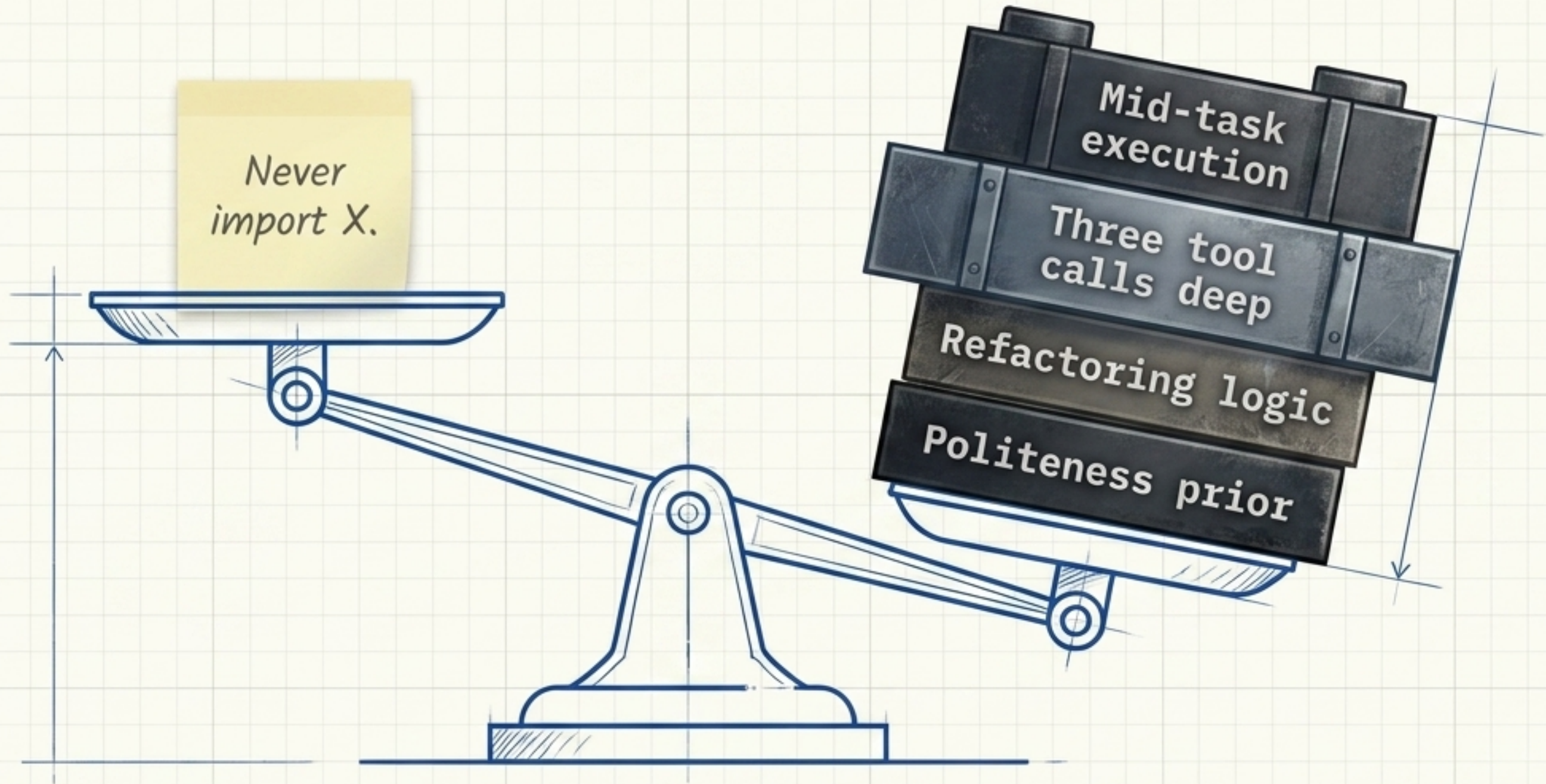
Anthropic frames instruction files as memory loaded into context. There is no part of the system that physically stops the agent from executing the forbidden action.

Instructions are context, not enforcement.



A README that says “core must not import adapters” while nothing in CI fails is not a rule. It’s a wish with good grammar.

Context pressure crushes soft preferences.



The model is a probabilistic system juggling many soft preferences. When under context pressure, the softest preferences are reliably dropped.

More instructions make the agent worse at everything else.



Context is a fixed budget. A 5,000-line root instruction file doesn't make the agent more careful—it makes it lose the three rules that actually matter for this specific change.

The Double Failure of the Markdown Reflex

Fails to Bind

You rely on prose.

The rule is ignored
under pressure.

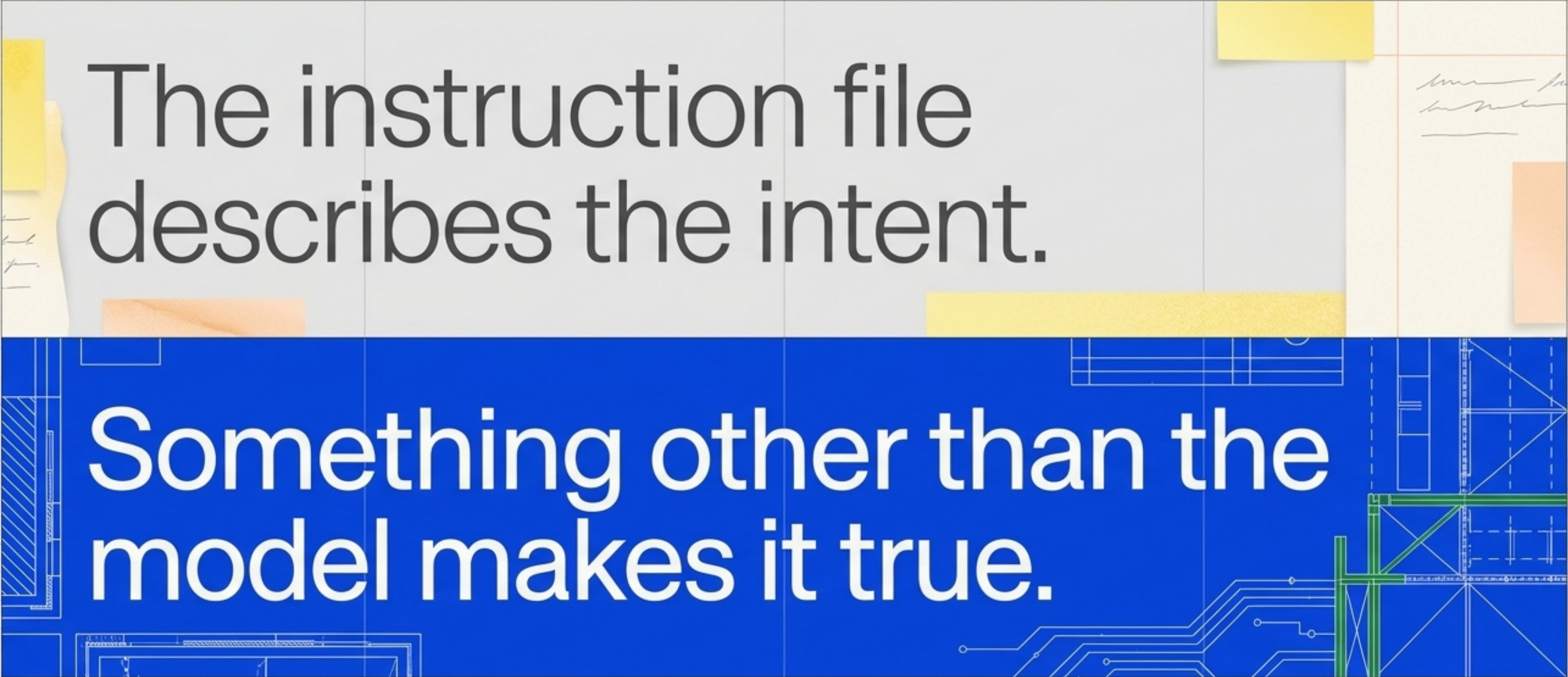
Dilutes the Rest

You added more prose.

Rules contradict, and
the agent guesses.

You have spent your context budget to buy enforcement you didn't get.

Prose proposes, hooks dispose.



The instruction file
describes the intent.

Something other than the
model makes it true.

Make your load-bearing rules executable.

Agent, please don't
hand-edit generated code.



```
regenerate  
git diff --quiet  
fail if moved
```

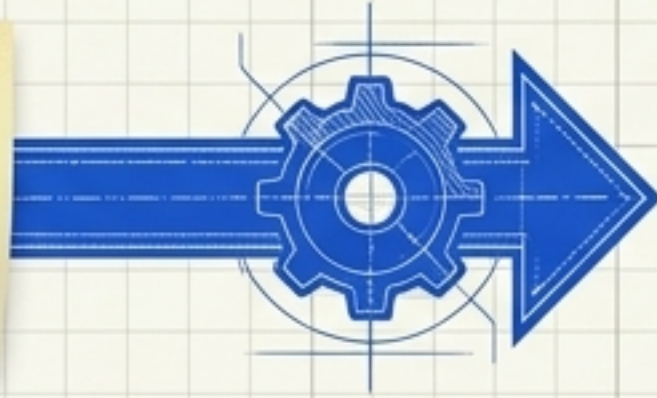
Do not trust the model. A check is indifferent to how persuasive the prose was or how confident the agent feels. It is a wall.

The Anatomy of a Rule: Prose vs. Hook

Dimension	Prose (CLAUDE.md)	Executable Hook (CI/Lint)
Medium	Markdown	Executable Code
System Type	Probabilistic	Deterministic
Agent's Perception	Soft Preference	Hard Wall
Under Pressure	Dropped	Indifferent
Failure Consequence	Bad code reaches main	Blocked before commit

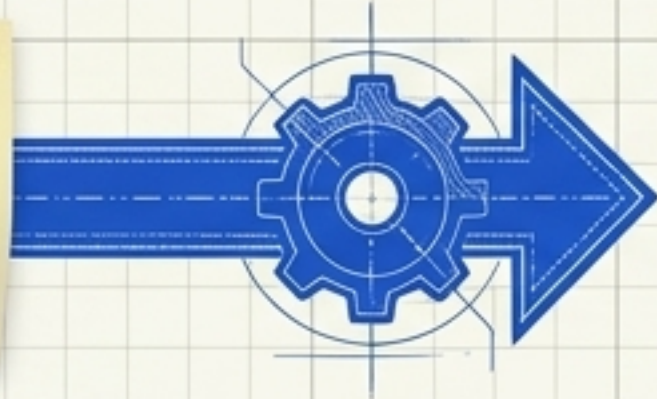
Turning wishes with good grammar into structural walls.

Domain stays pure



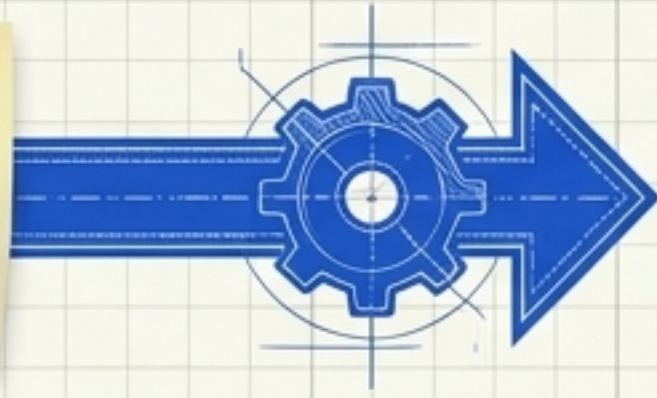
Lint Rule:
no-restricted-imports /
dependency-cruiser

No files over
300 lines



CI Pipeline:
giant-file-check.sh

Stop asking to
review work



Agent Harness: Stop Hook
(blocks review patterns)

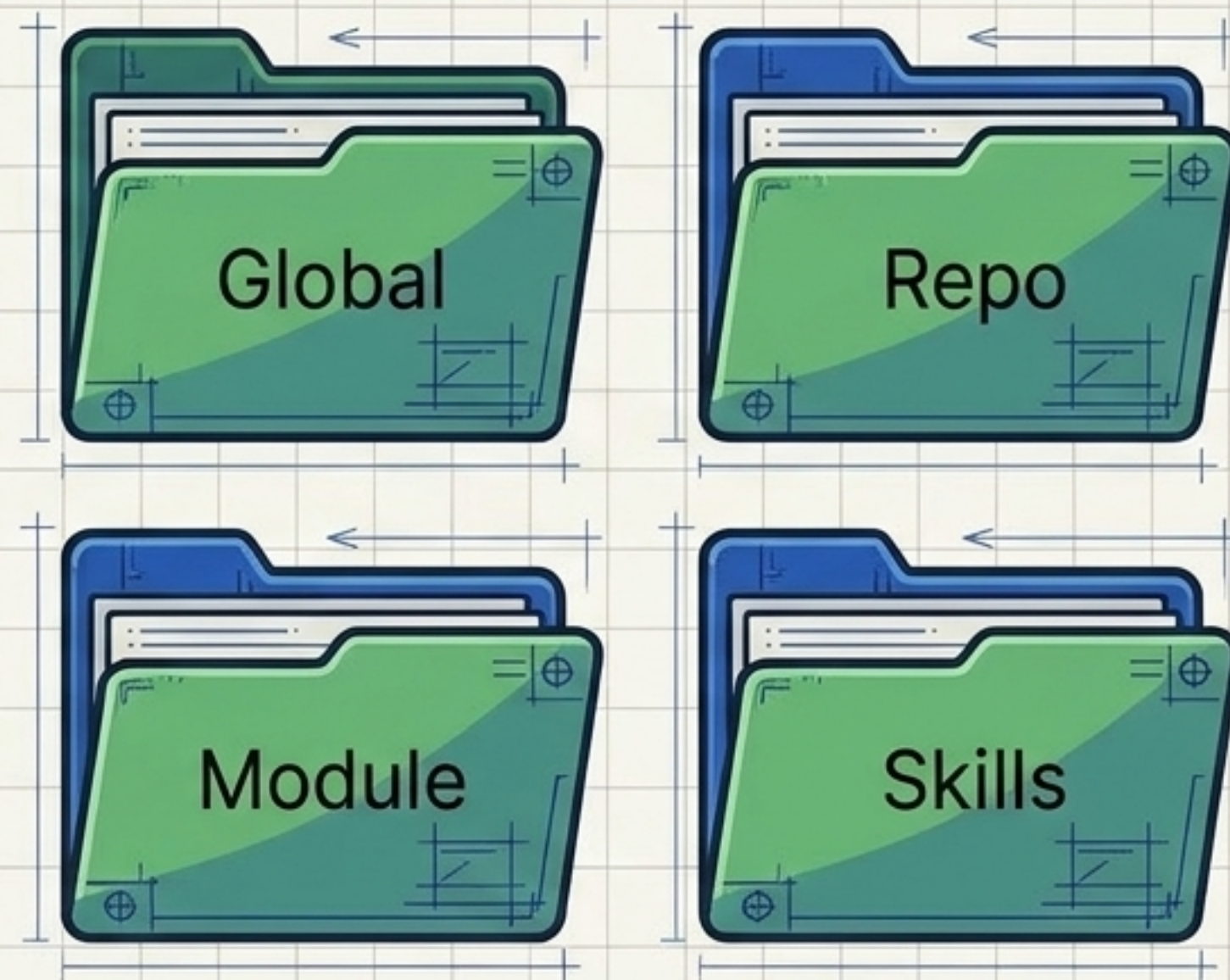
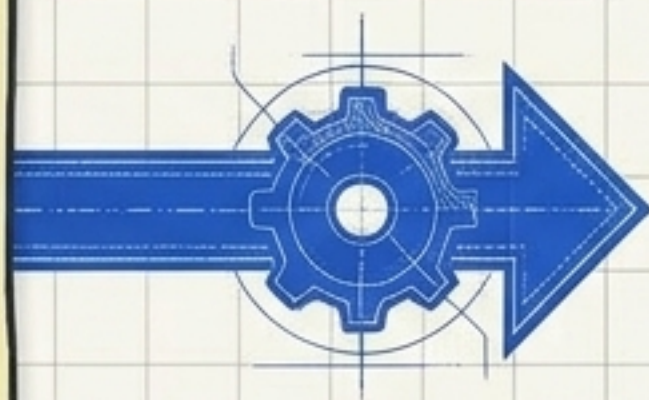
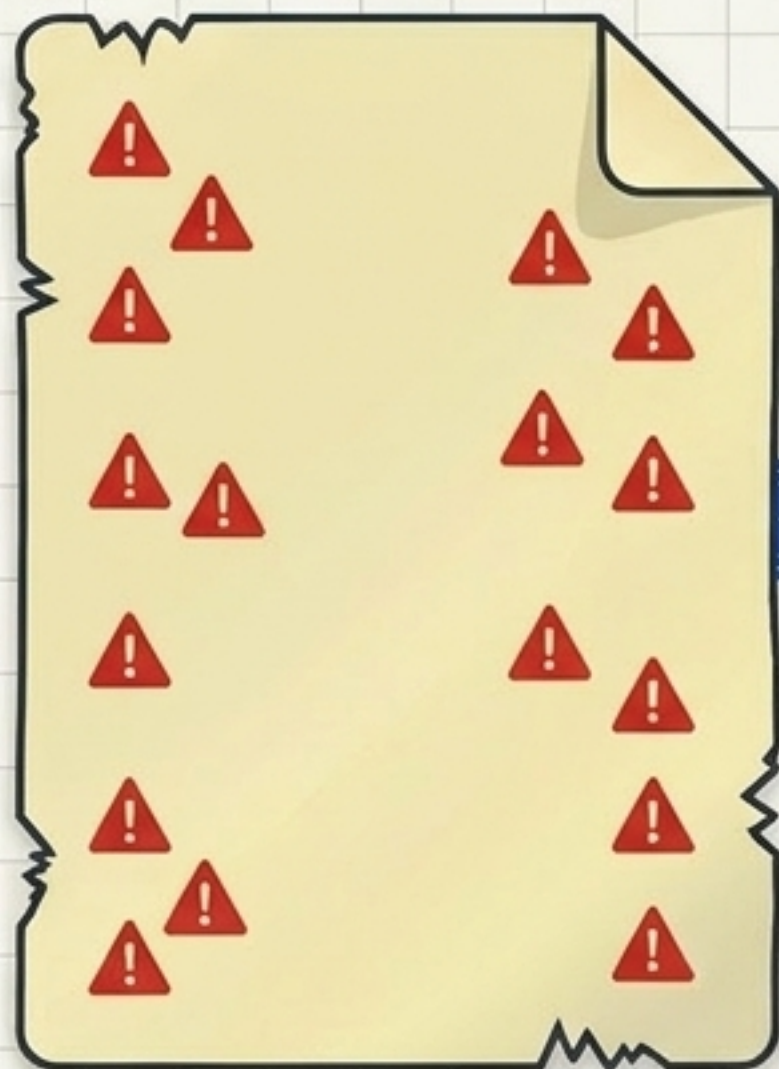
Done is an exit code, not a vibe.



The agent can run into this wall as many times as it likes.
Without a zero exit code, the task cannot proceed.

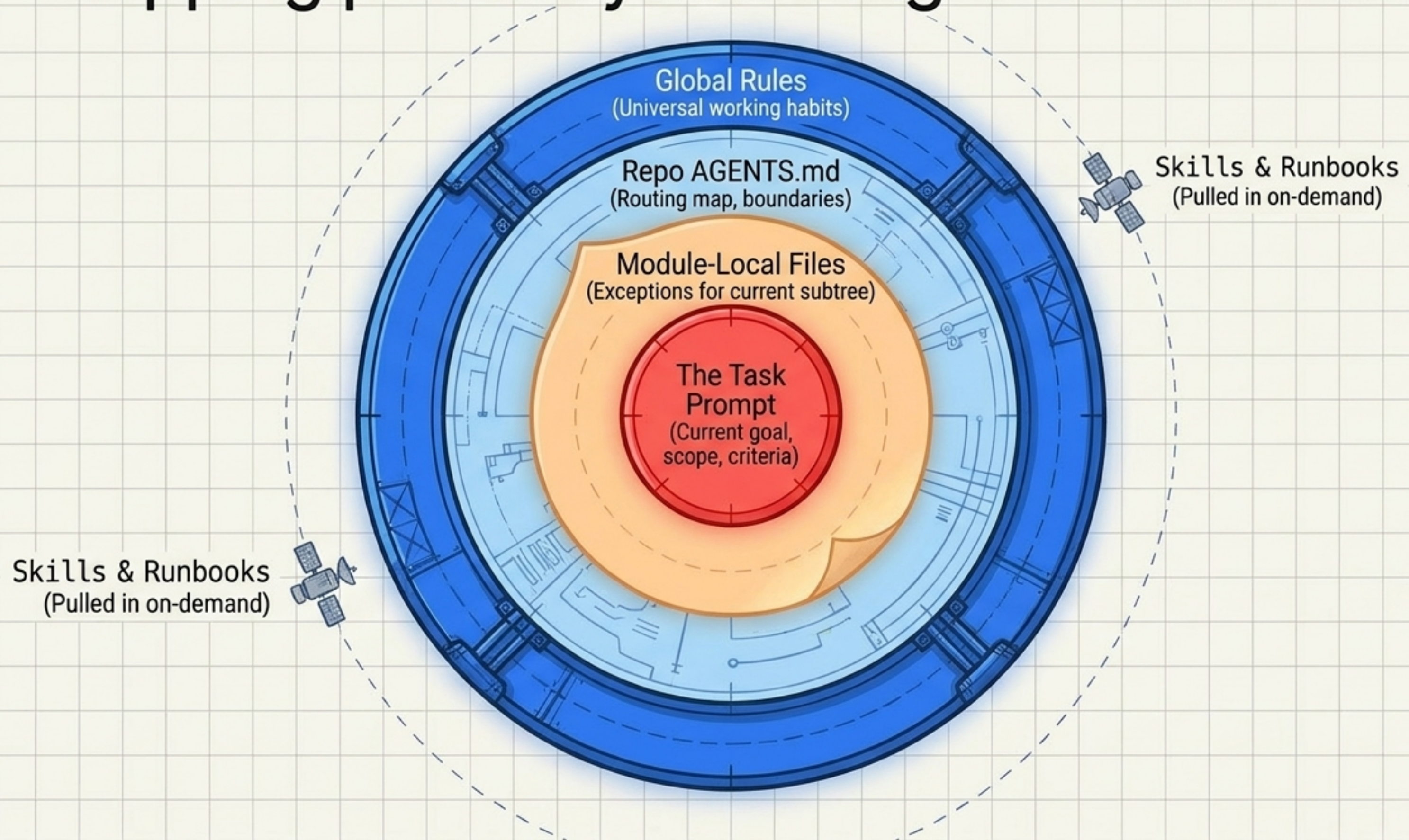
Restructuring Context: Progressive Disclosure

5,000-line AGENTS.md



Once the walls are doing the enforcement work, the prose can get shorter. **Use layers to keep the hot path lean.**

Mapping proximity to the agent's attention.



Stop writing knowledge dumps. Start writing routing maps.

~/.claude.md (Global Level)

What the agent needs every single time, for every task.

✓ Contains: Verification discipline, commit preferences, working habits.

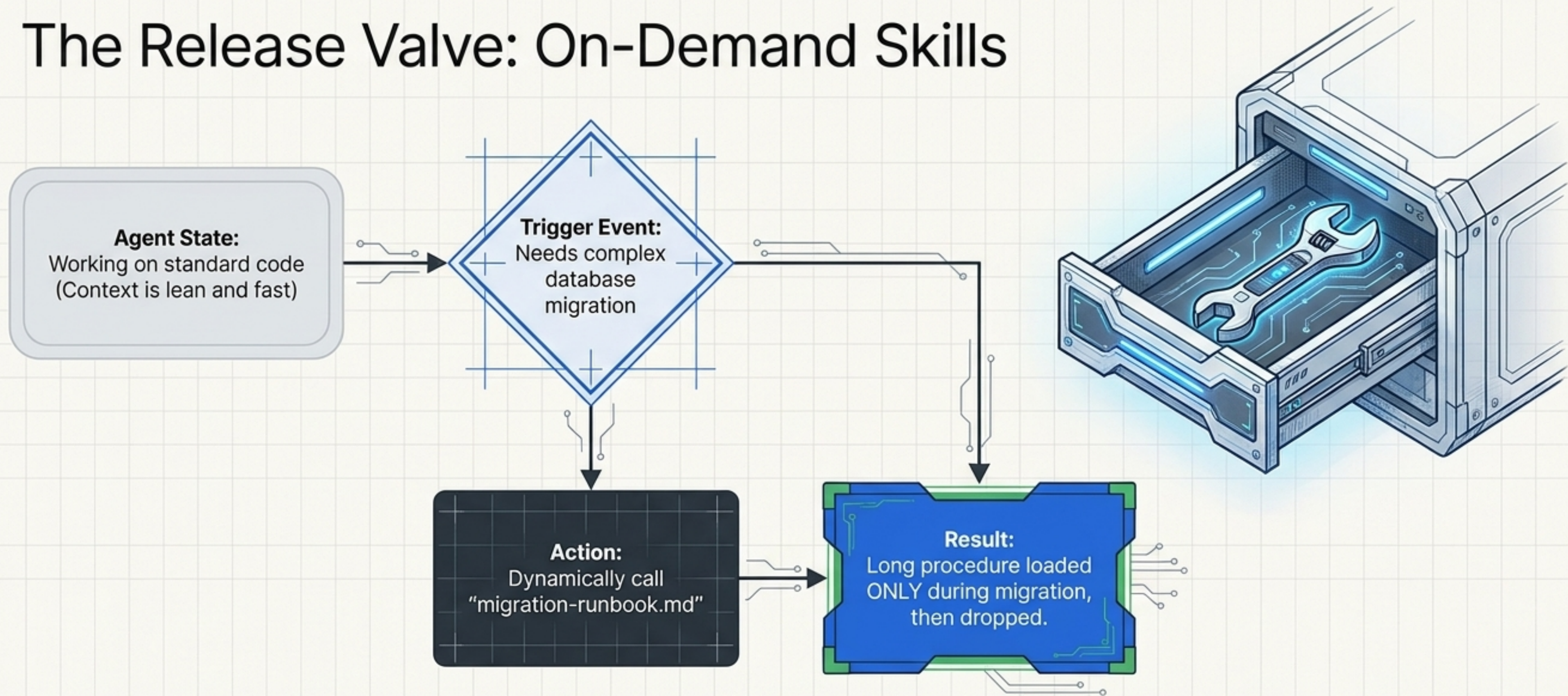
✗ Excludes: Repo architecture.

/repo/AGENTS.md (Repo Level)

The routing map.

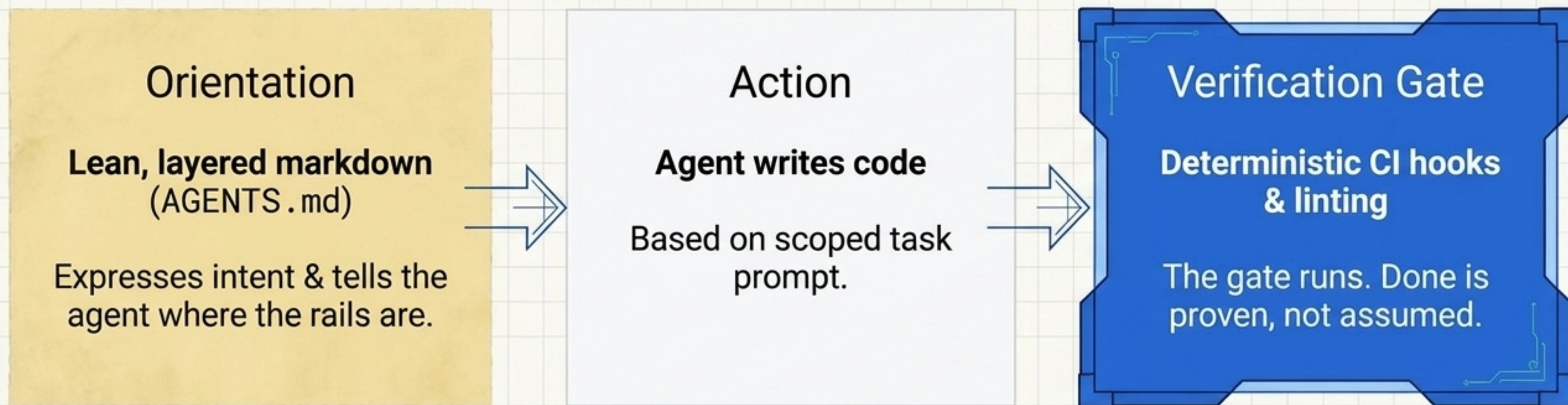
✓ Contains: Architecture map, available commands, boundaries, definition of done.

The Release Valve: On-Demand Skills



Multi-step runbooks don't belong in permanent context.
Pull them in on demand to keep the always-on text lean.

The Architecture of Honesty



The markdown's job was never to police the agent.
Its job is to tell the agent where the police are.

Let the gate keep the repo honest.



The model isn't the product. The harness is. Point it at your own repo.