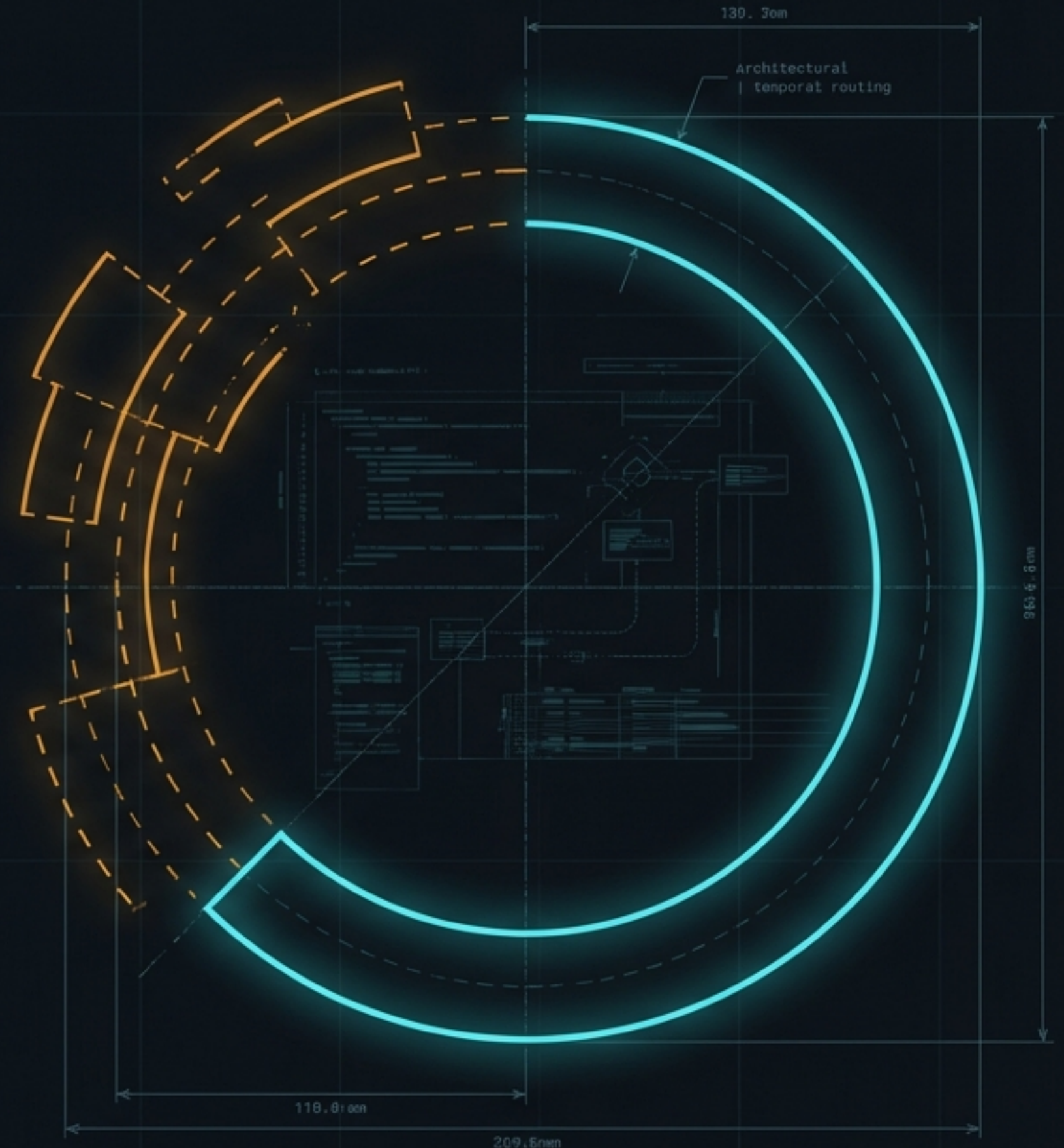


Foundry: Orchestrating the Autonomous Dev Team

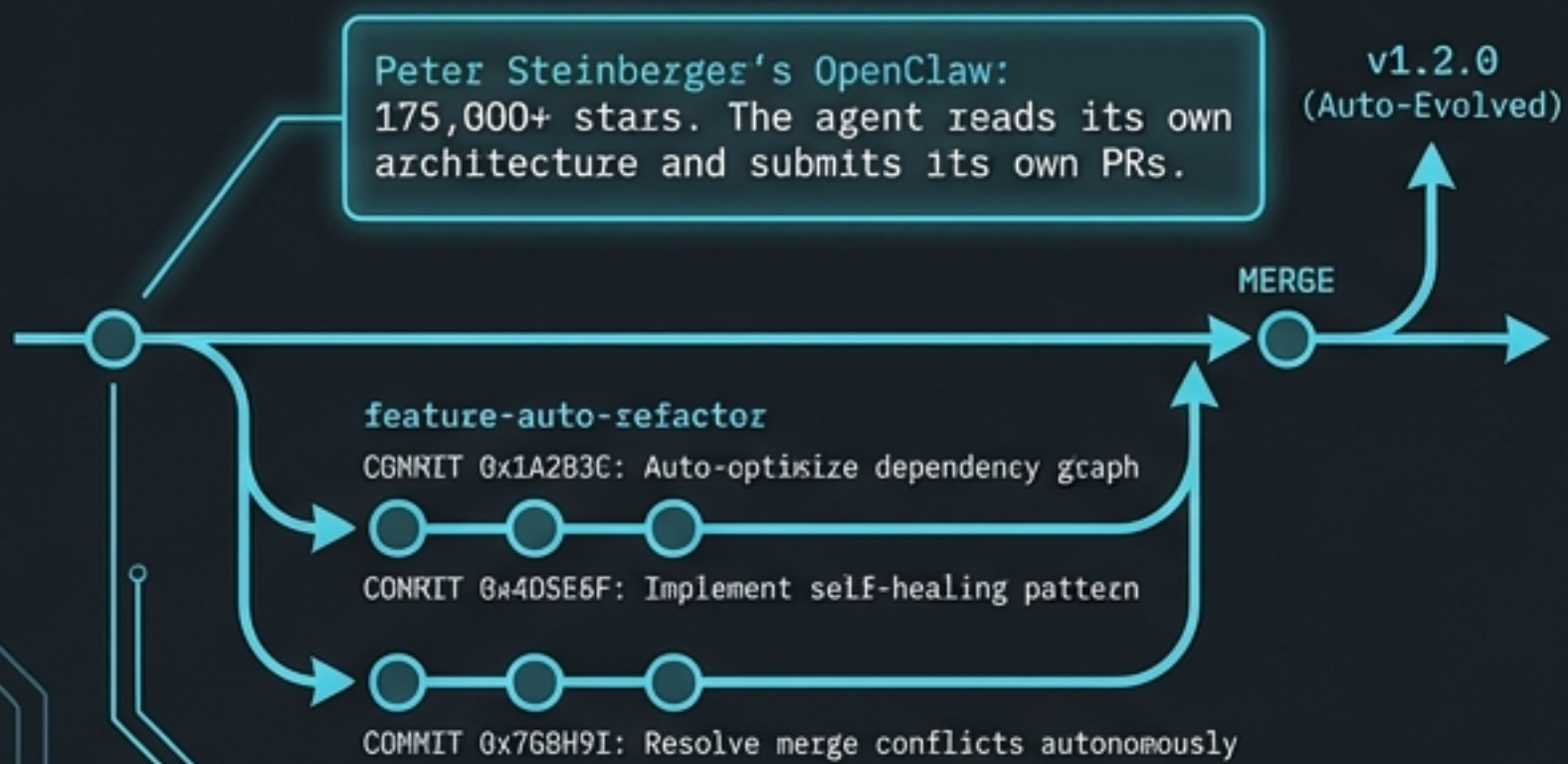
Closing the loop on self-modifying code with Temporal.

arch: temporal_backbone | status: deployed



Code has learned to evolve itself. But the loop isn't closed.

Models are brilliant. Harnesses are powerful.
But right now, the system cannot manage itself.

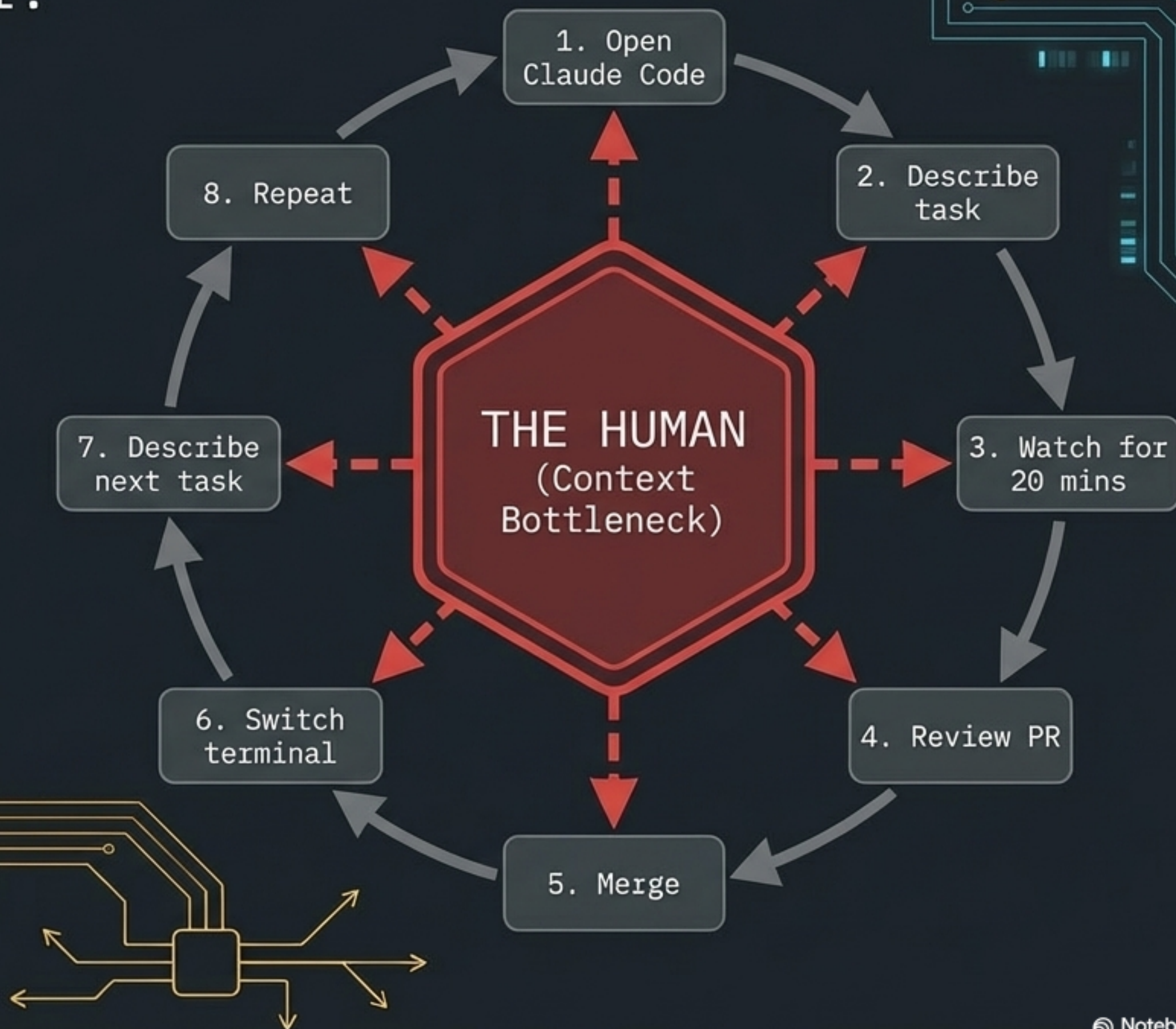


The model can code.
But nobody has built the foreman.

You are the Orchestrator.

Every time you copy-paste context, evaluate a dependency, or map a Linear issue to a terminal session, you are executing tasks a system should automate.

You are the loop.



Good Ideas, Fragile Bones.

	Tool	Pros	Cons
Row 1	Composio A0	Great plugin architecture (7 typed interfaces).	Single-machine polling. Flat files. No durable execution (crashes lose state).
Row 2	AgentsMesh	Perfect Kanban-to-pod binding abstraction.	BSL-licensed. Wrong deployment model. No webhook-driven flow.
Row 3	dmux	Beautifully simple local terminal multiplexer.	Local-only. No API surface. No webhook integration.

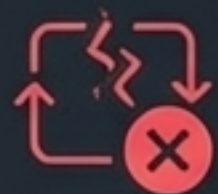
None of the existing tools natively support durable, long-running, multi-system loops.

The Goal: Instruction In, Merged Code Out.

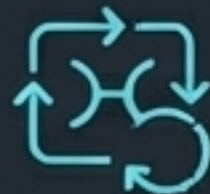
The human provides the intent and walks away. The system autonomously handles decomposition, parallelization, review feedback loops, and deployment.



The 5 Unsolved Primitives of AI Orchestration.



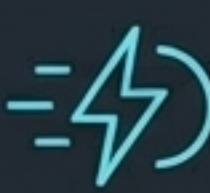
State: Crash at minute 29 of a 30-min agent task?



Durable execution.
Survive, replay, resume.



Compute: Wait 5 minutes for a PR review? Polling is wasteful.



Zero-cost wait.
Event-driven resumption.



Secrets: Prompt injection in full-auto bash exfiltrates keys?



Privilege separation. Code writers cannot call APIs.



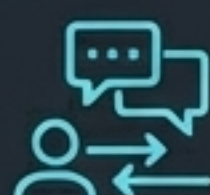
Identity: GitHub blocks an agent approving its own PRs?



Dual identities (coder vs. reviewer).



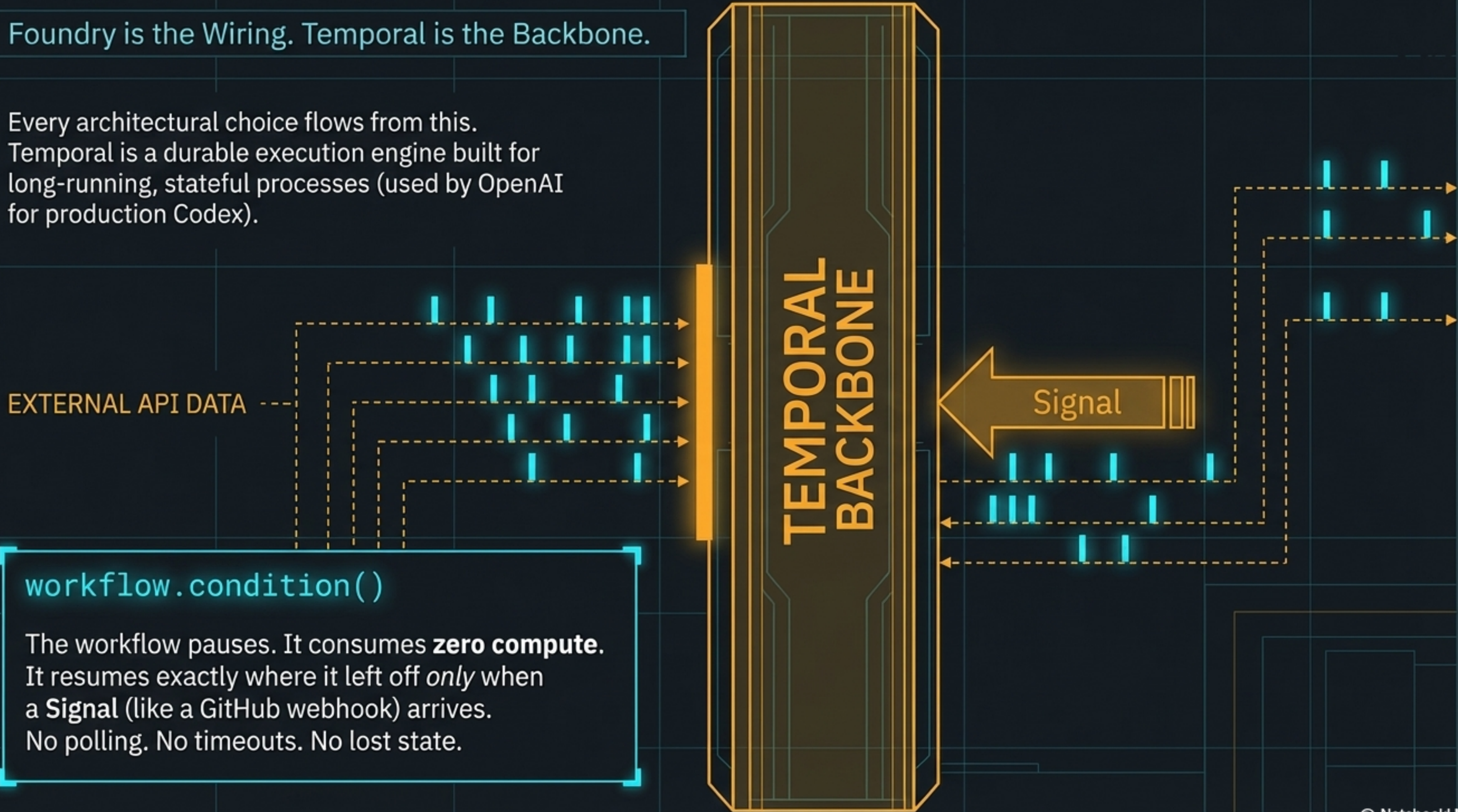
Feedback: Async PR comments land in a dead inbox?



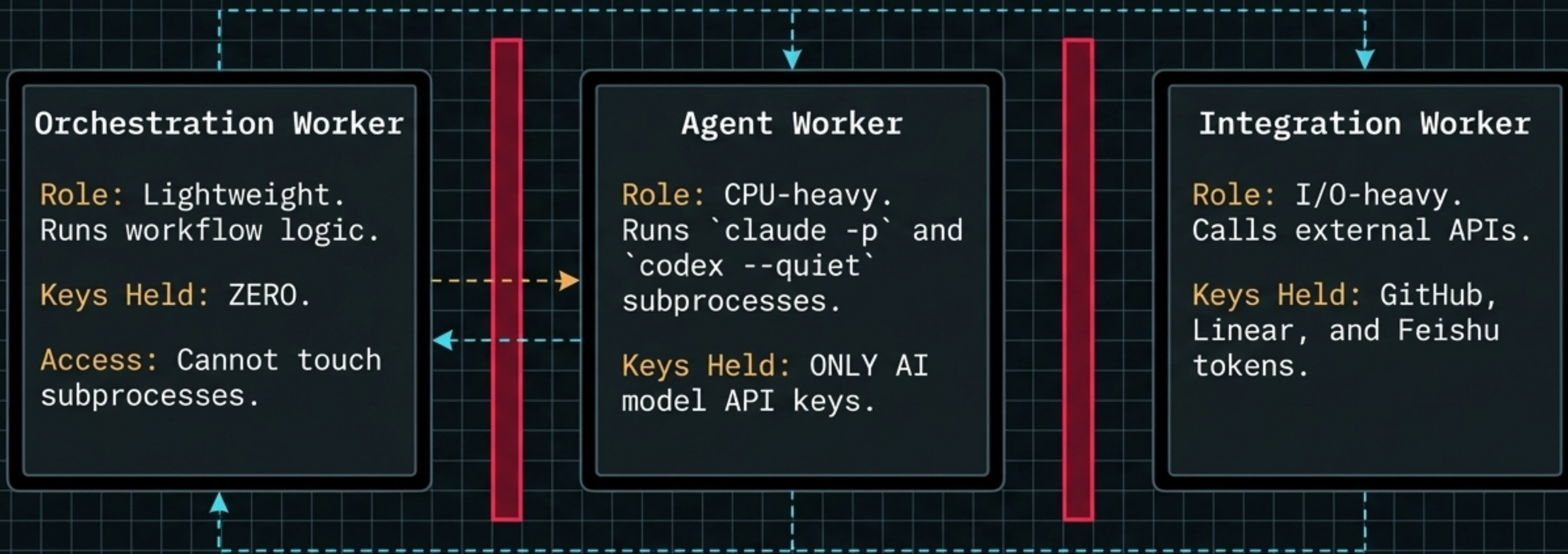
Stateful conversational bridging across systems.

Foundry is the Wiring. Temporal is the Backbone.

Every architectural choice flows from this. Temporal is a durable execution engine built for long-running, stateful processes (used by OpenAI for production Codex).



The 3-Worker Architecture: Minimal Attack Surfaces



If an agent is prompt-injected via a malicious PR comment, it physically cannot exfiltrate your Linear API key. It literally doesn't have it.

The Job Site Hierarchy.

Parent Workflow (The Foreman)

Receives instruction -> Calls Claude to decompose -> Creates Linear issues -> Fans out N parallel child workflows -> Aggregates results -> Sends Feishu summary.

[isolated git worktree]

Child Workflow (The Builder)

Code -> PR
-> Wait for Review
-> Iterate (up to 5 rounds)
-> Merge

[isolated git worktree]

Child Workflow (The Builder)

Code -> PR
-> Wait for Review
-> Iterate (up to 5 rounds)
-> Merge

[isolated git worktree]

Child Workflow (The Builder)

Code -> PR
-> Wait for Review
-> Iterate (up to 5 rounds)
-> Merge

Software Building Itself.

The model can code. The harness gives it a job site, a foreman, and a safety protocol.

The agent doesn't need a manager. It manages itself.
Orchestration of orchestration is the final unlock.

