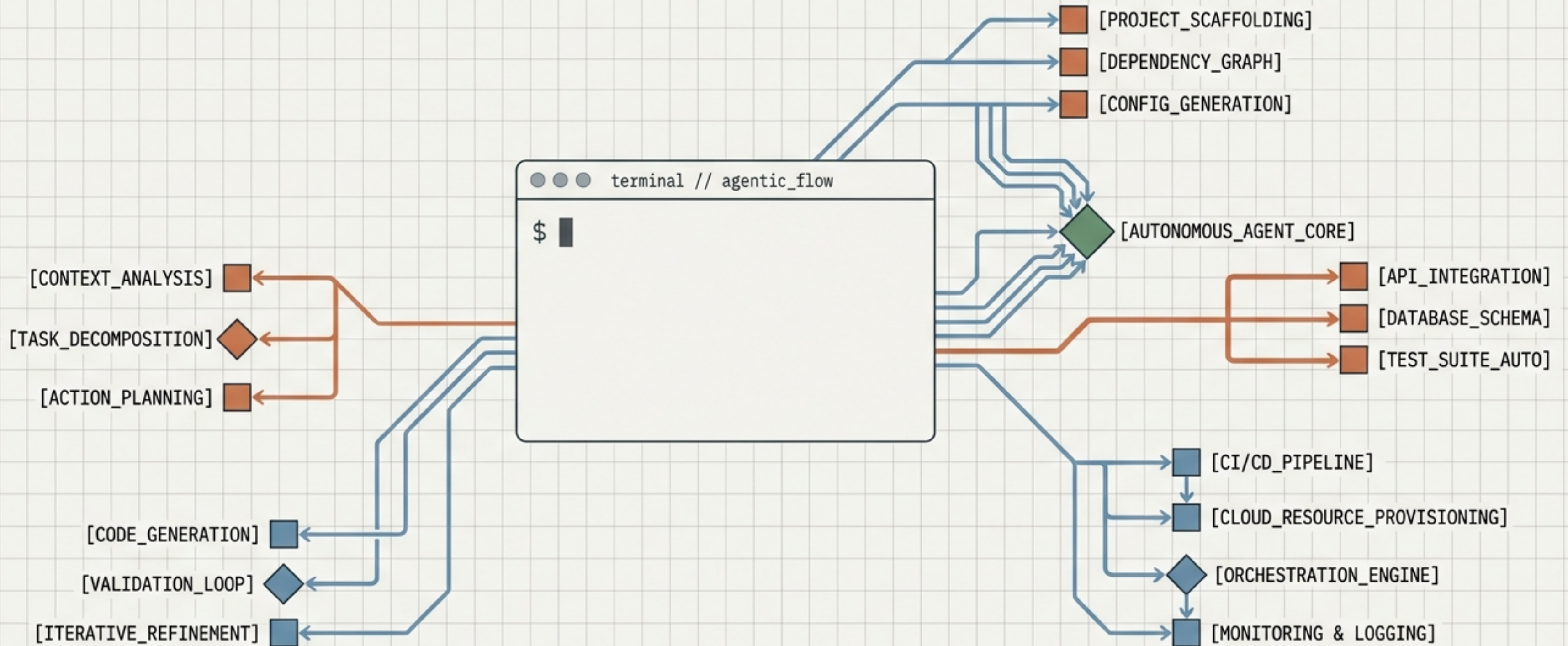


From Chat to Command: The Agentic AI Paradigm

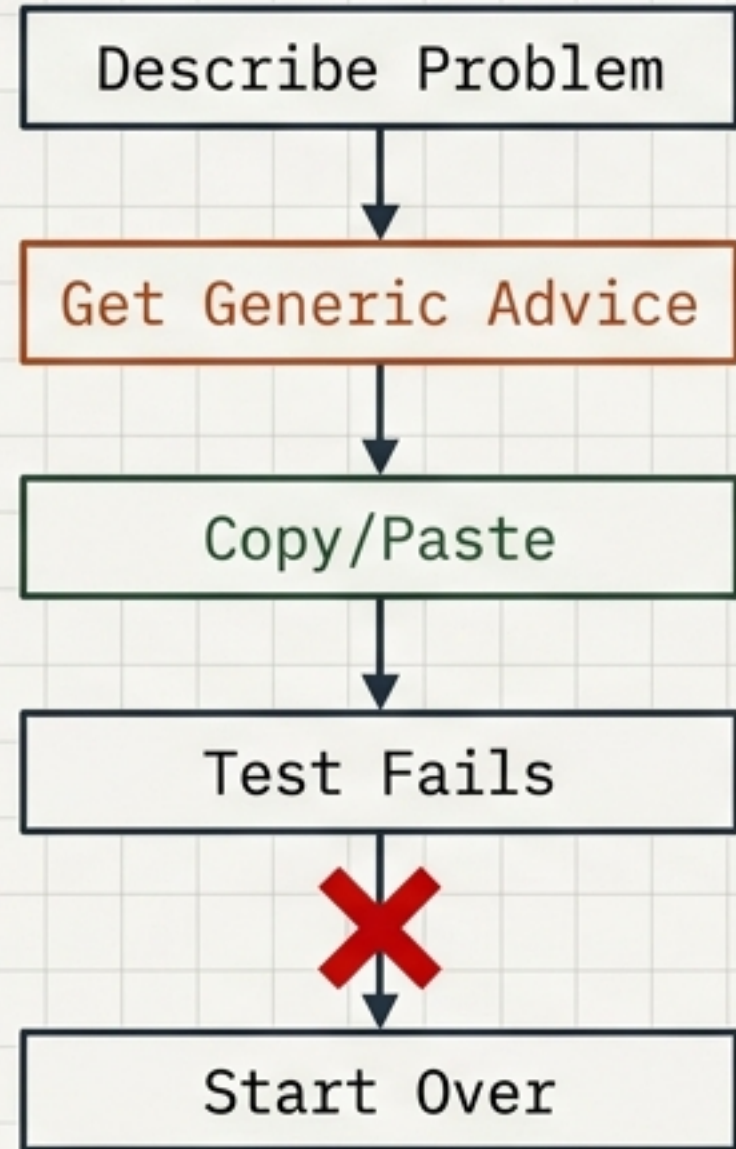
VERSION: CLAUDE_CODE // OPENCODE

Mastering the primitives of autonomous development workflows.

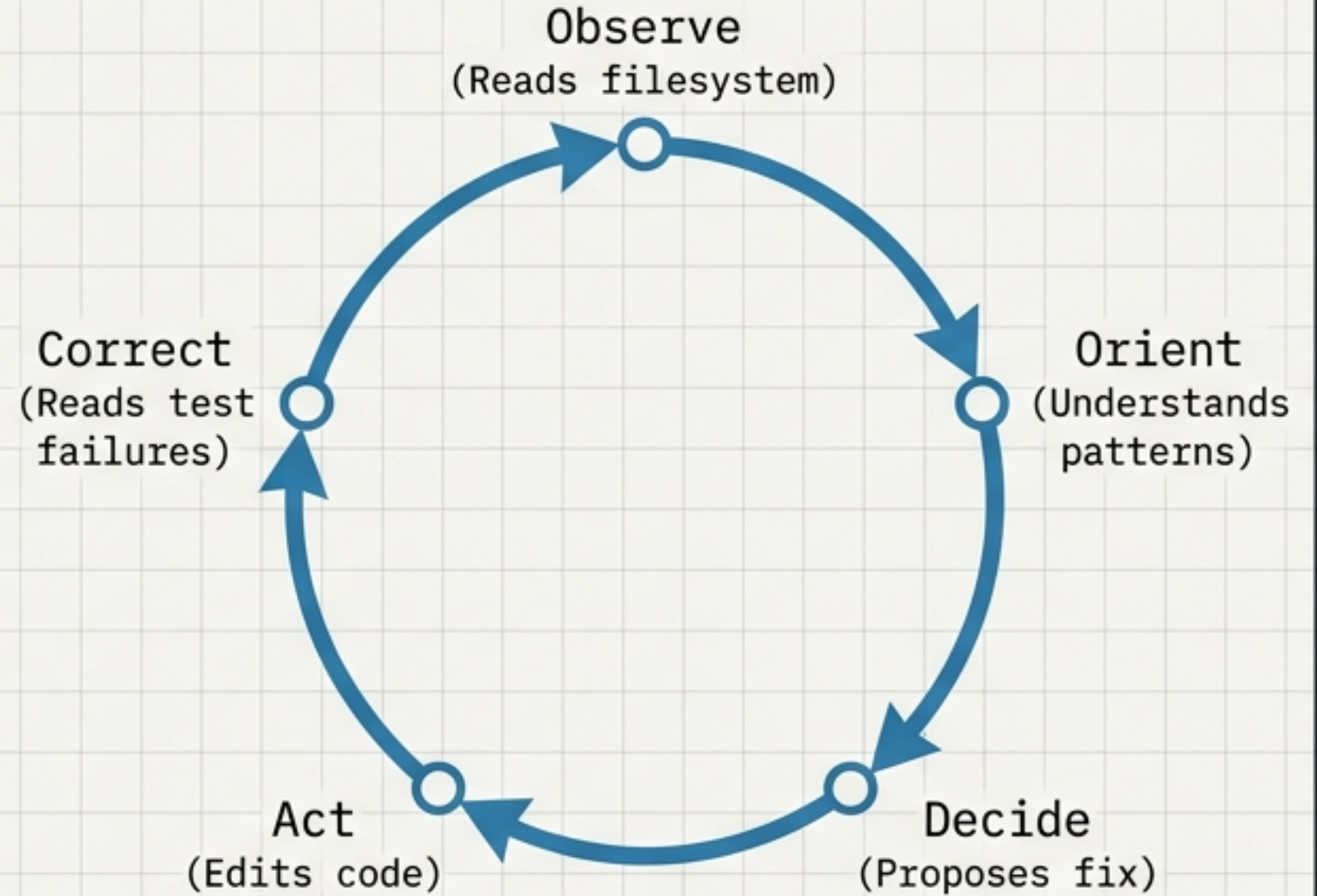


The Capability Was Always There, It Just Needed Eyes

Passive AI: The Consultant on the Phone



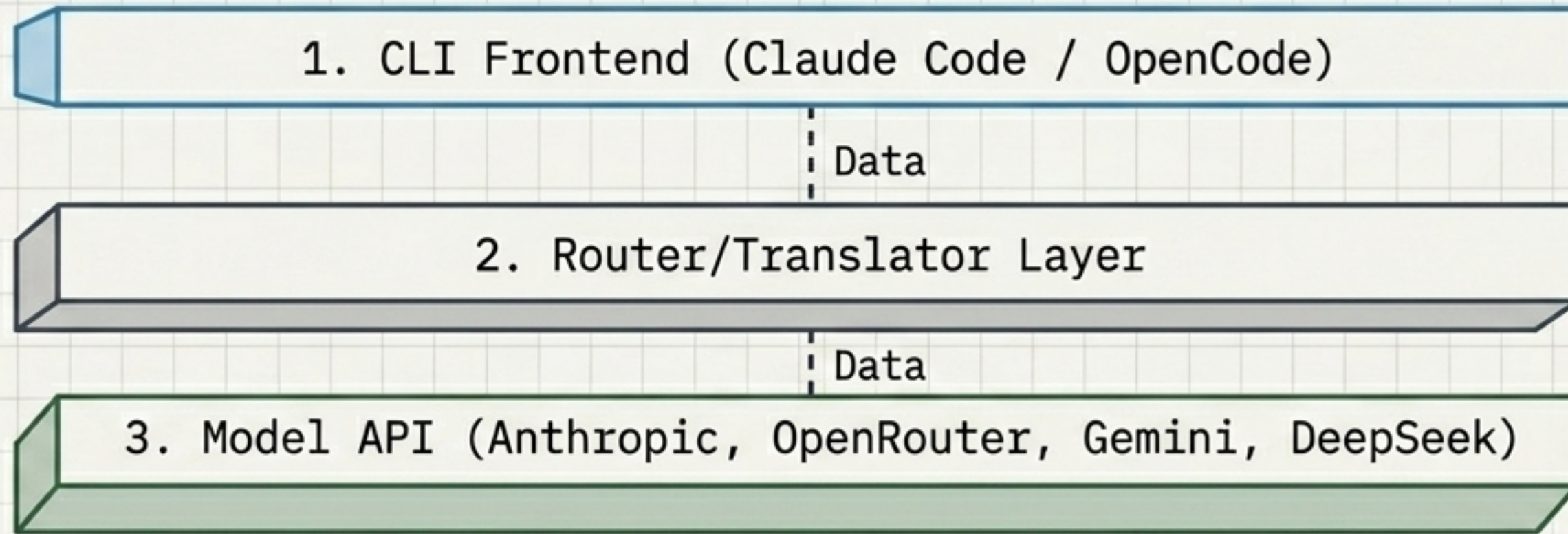
Agentic AI: The Pair Programmer



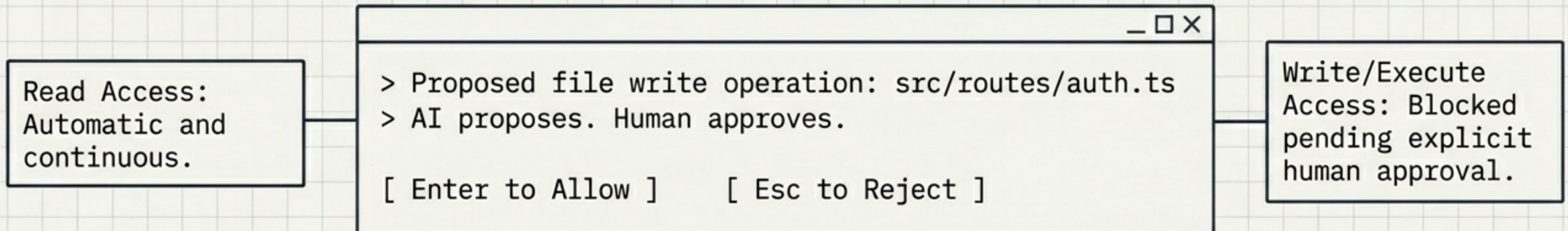
Product Overhang: The reasoning engine existed; it just needed direct filesystem access to eliminate copy-paste friction.

The Architecture of Human-in-the-Loop Autonomy

The 3-Layer Stack



The Security Gate

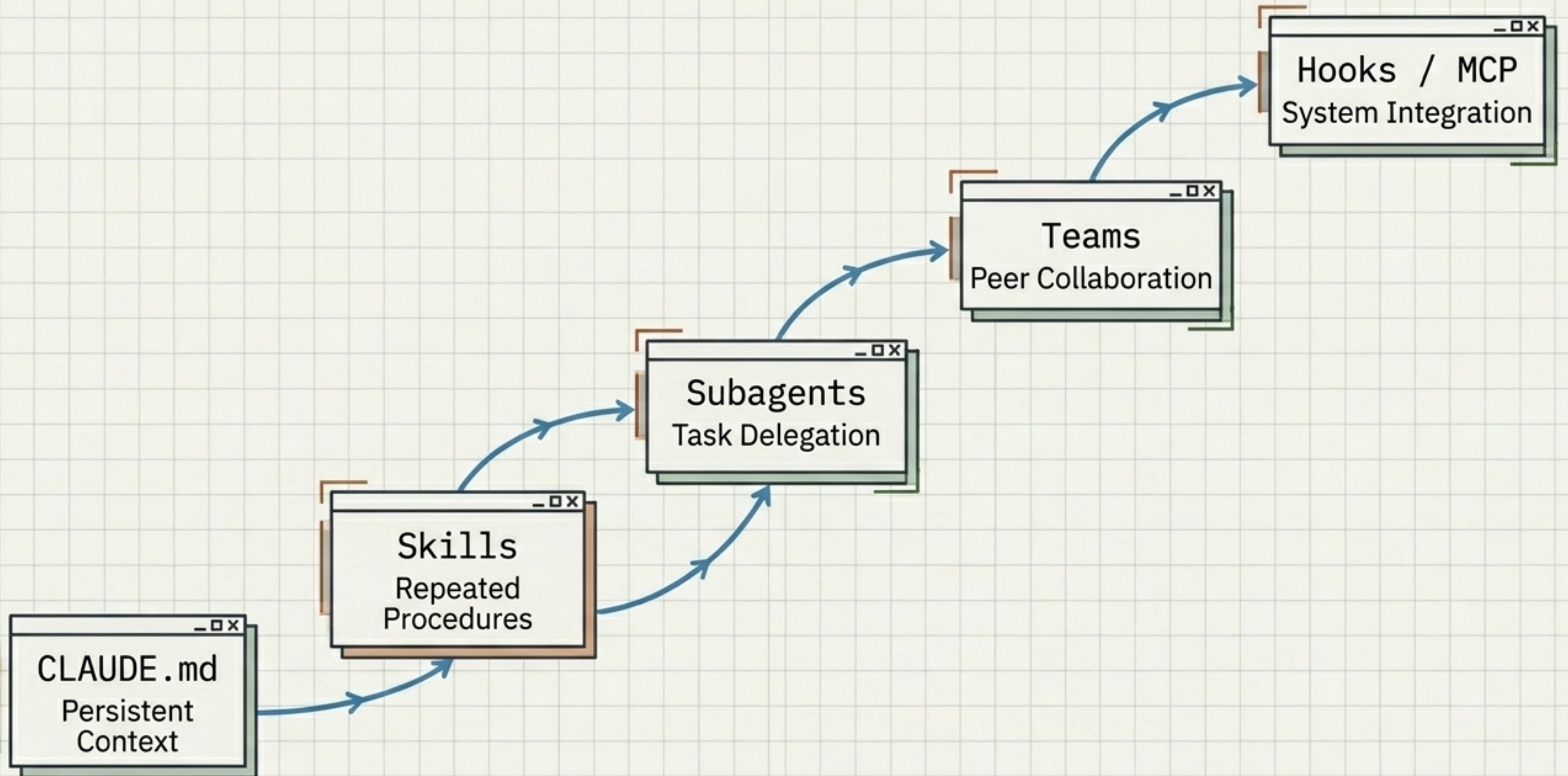


General Agents Build the Custom Agents

General Agent (Claude Code)		Custom Agent (SDK-Built)	
Analogy	Senior Consultant	Analogy	Assembly Line
Flexibility	Handles novel problems across any domain.	Flexibility	Highly optimized for one specific, repetitive workflow.
Setup	Instant	Setup	Weeks to build

The Agent Factory Model: You use the General Agent to build and orchestrate Custom Agents.

The Five Primitives of Agent Extensibility



Encoding Expertise into Executable Skills

YAML FRONTMATTER

```
name: code-reviewer
```

```
description: Reviews typescript files for anti-patterns. Triggered when asked to review code or check for bugs.
```

MARKDOWN BODY

```
# Instructions
```

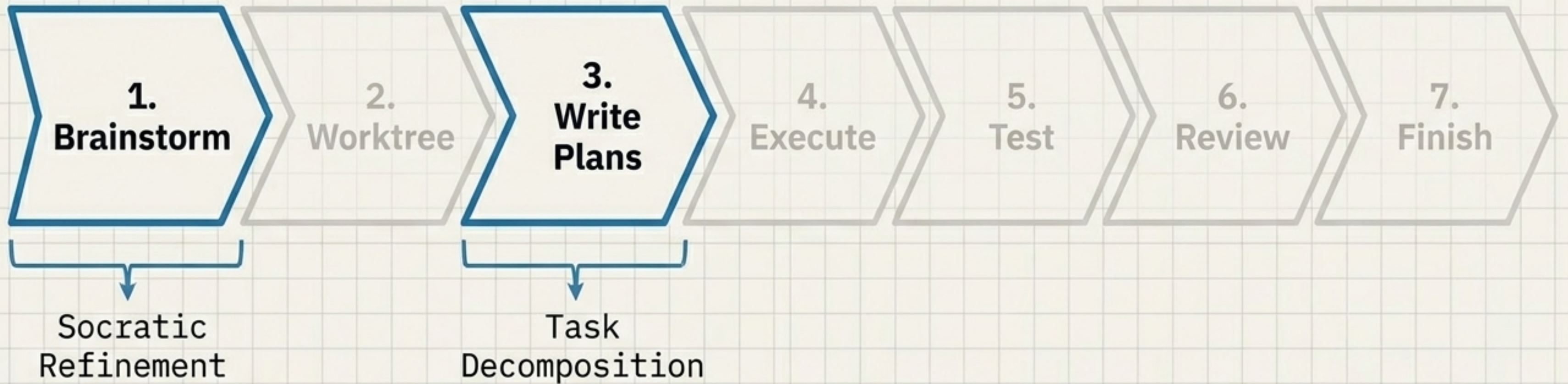
1. Check for memory leaks in React effects.
2. Enforce strict type safety.

The Description Formula:

Action Verb + Input Type + Output Type + Trigger Conditions
Action Verb Input Type Output Type Trigger Conditions

The **description** is the ID card. It dictates exactly when the agent auto-activates the skill without manual prompting.

The Superpowers Workflow Front-Loads the Architecture



The Philosophy

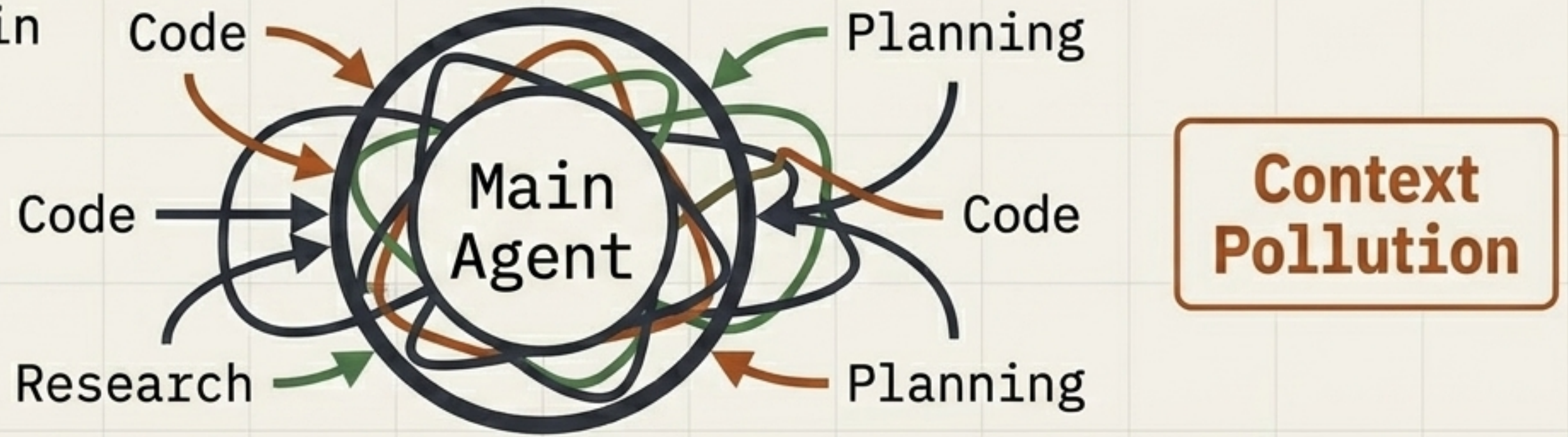
Bugs are **cheap to fix**; building **the wrong thing** is expensive.

The Output

AI decomposes a high-level design into 2-5 minute granular tasks with **exact file paths** and **verification steps** before writing any code.

Subagents Protect Against Context Pollution

The Cluttered Brain

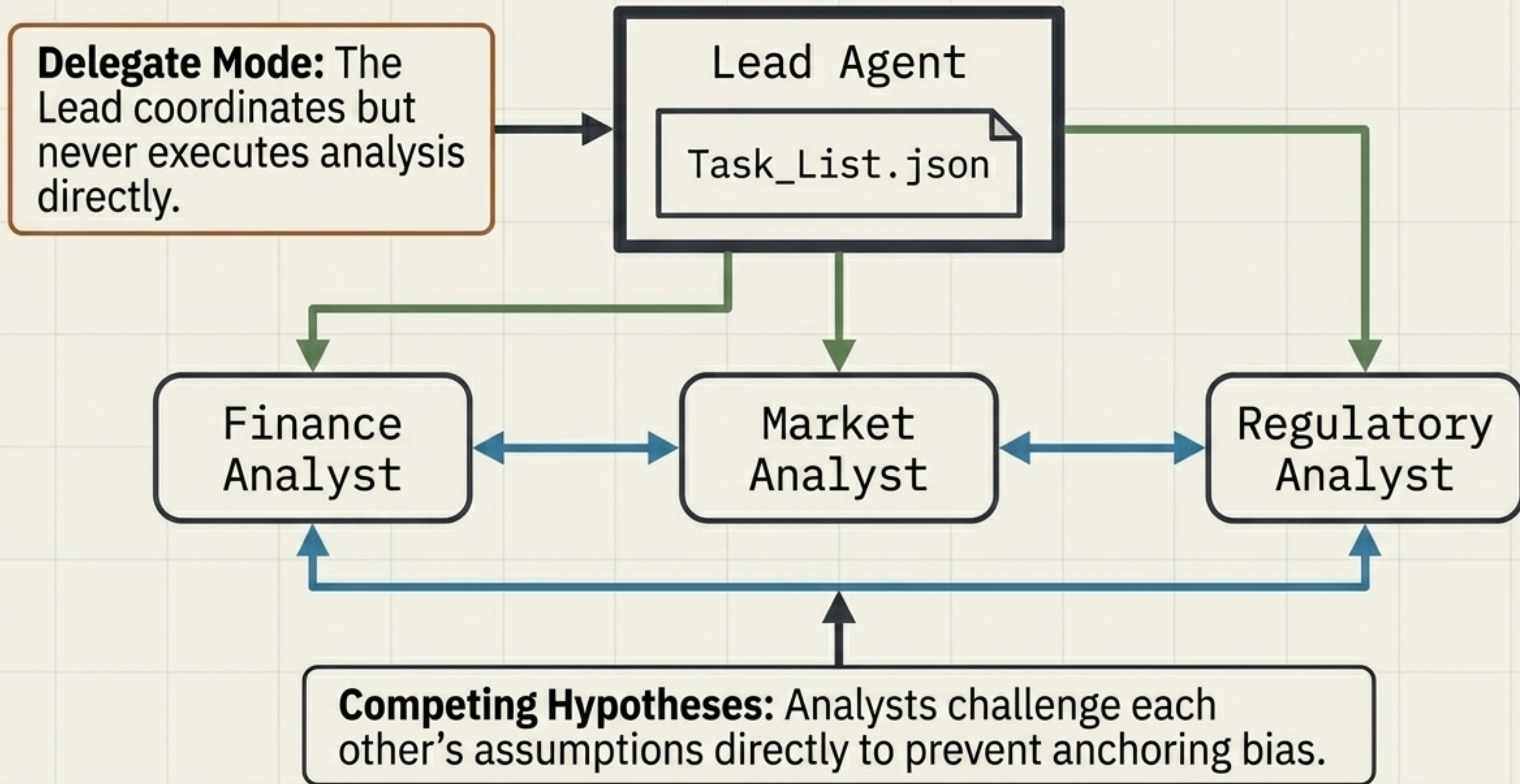


The Hub-and-Spoke Model



Subagents are **fire-and-forget specialists**. They execute one specific goal in a completely isolated context window, return the result to the main session, and terminate.

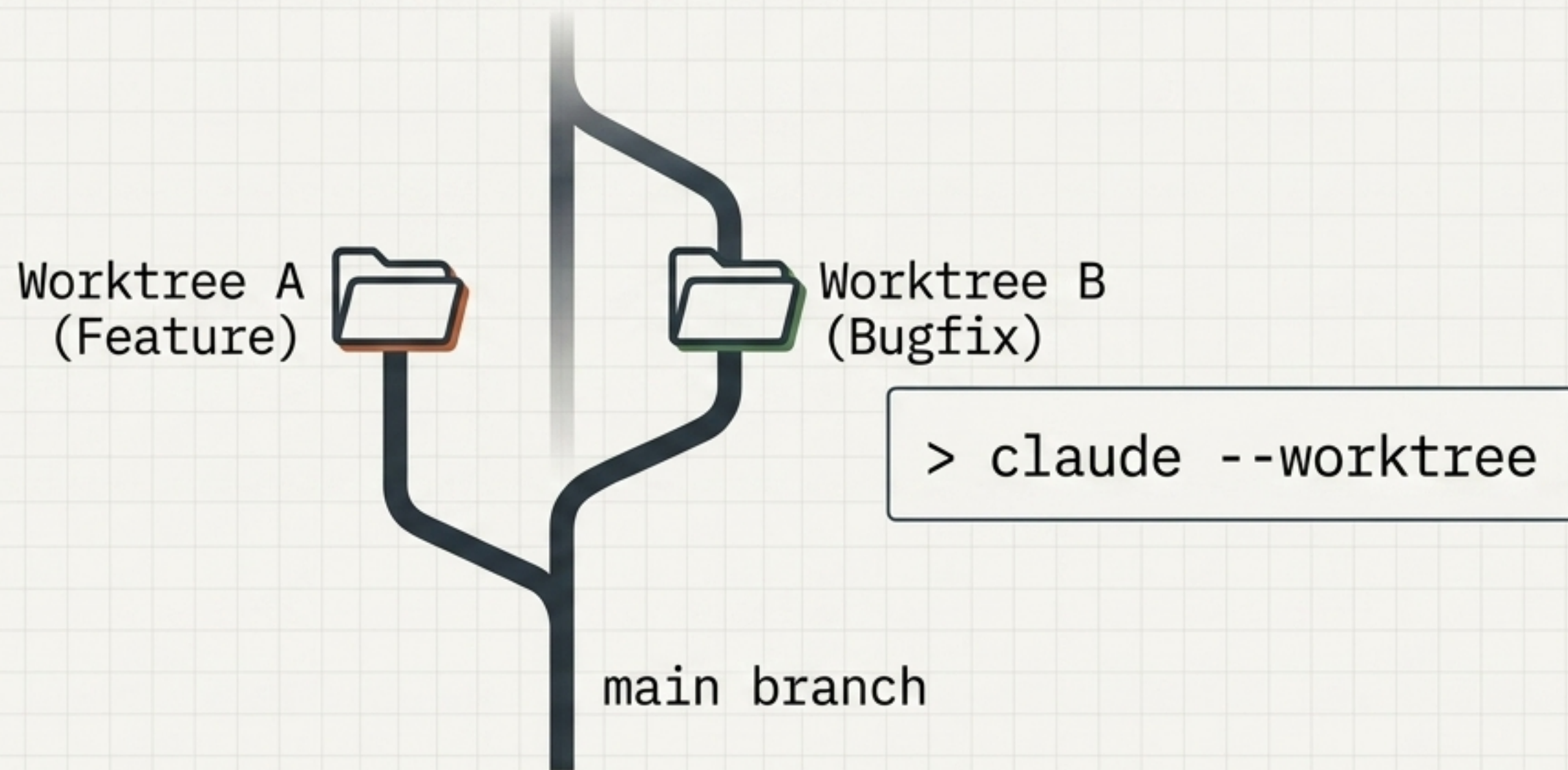
Agent Teams Enable Peer-to-Peer Orchestration



The Delegation Diagnostic

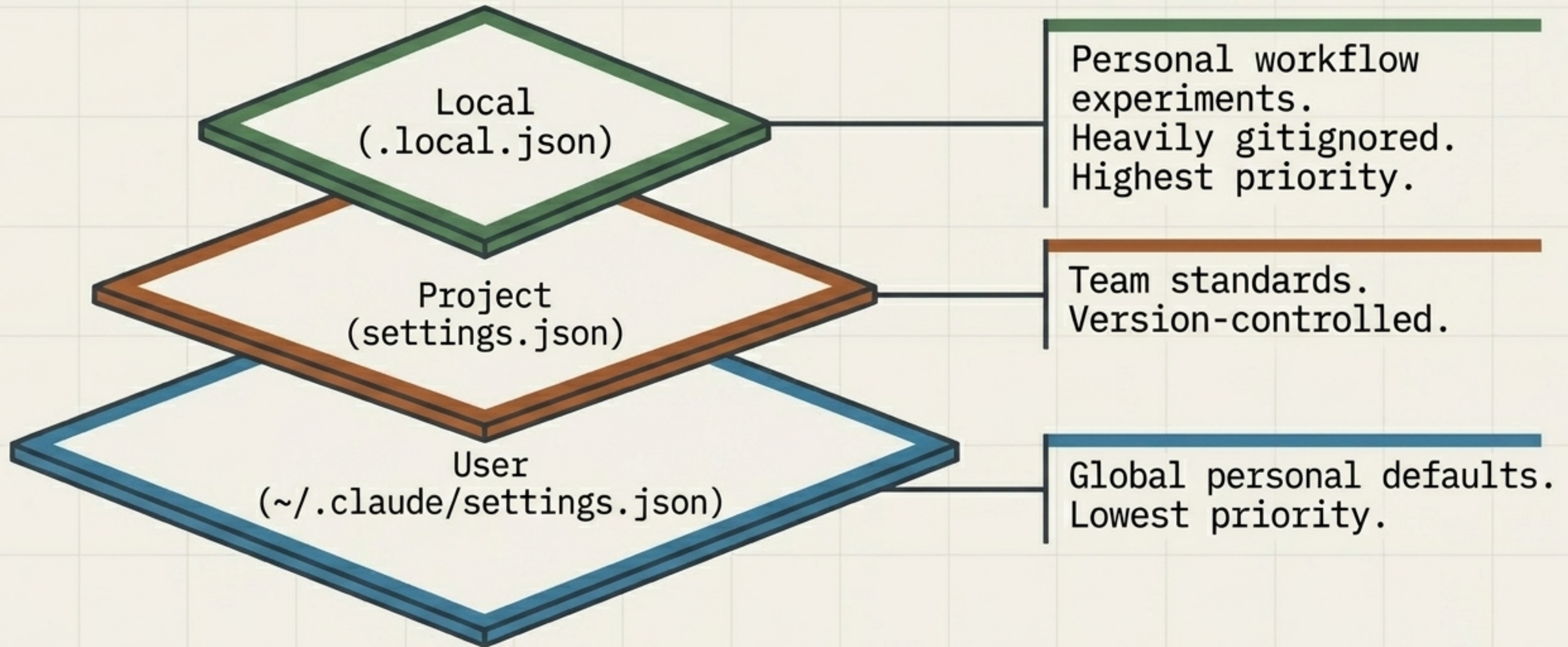
	Skills	Subagents	Agent Teams
Invocation	Auto-activation	Explicit invocation	Lead-orchestrated
Context	Shared context	Isolated context	Isolated + peer messaging
Ideal Use Case	Lightweight formatting & procedures	Single-focus audits & deep research	Complex cross-functional debate

Worktrees Provide Safe Parallel Execution



Parallel agents require physical isolation. Worktrees create a full, isolated copy of the repository with a separate git index, allowing simultaneous rewrites without conflict.

The Three-Tiered Settings Precedence Hierarchy



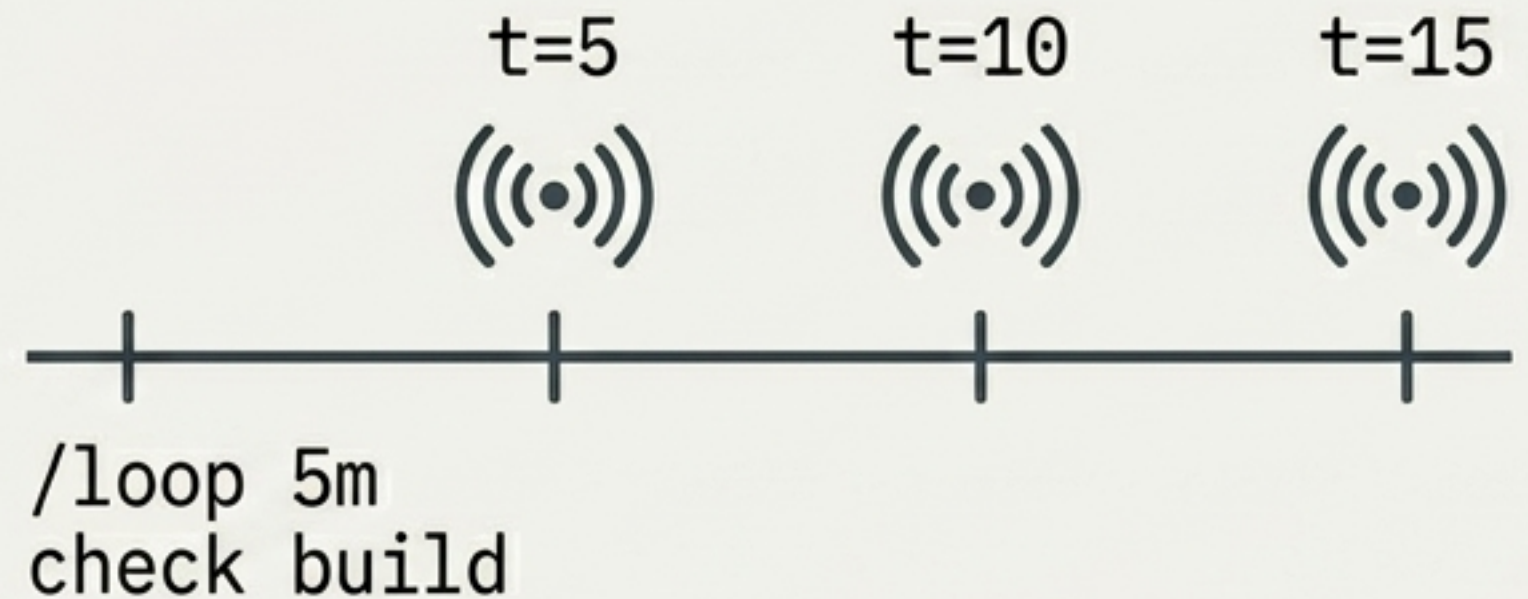
Precedence Rule: Local ALWAYS overrides Project. More specific scope wins.

Unbound Sessions: Remote Control & Scheduled Loops

Remote Control

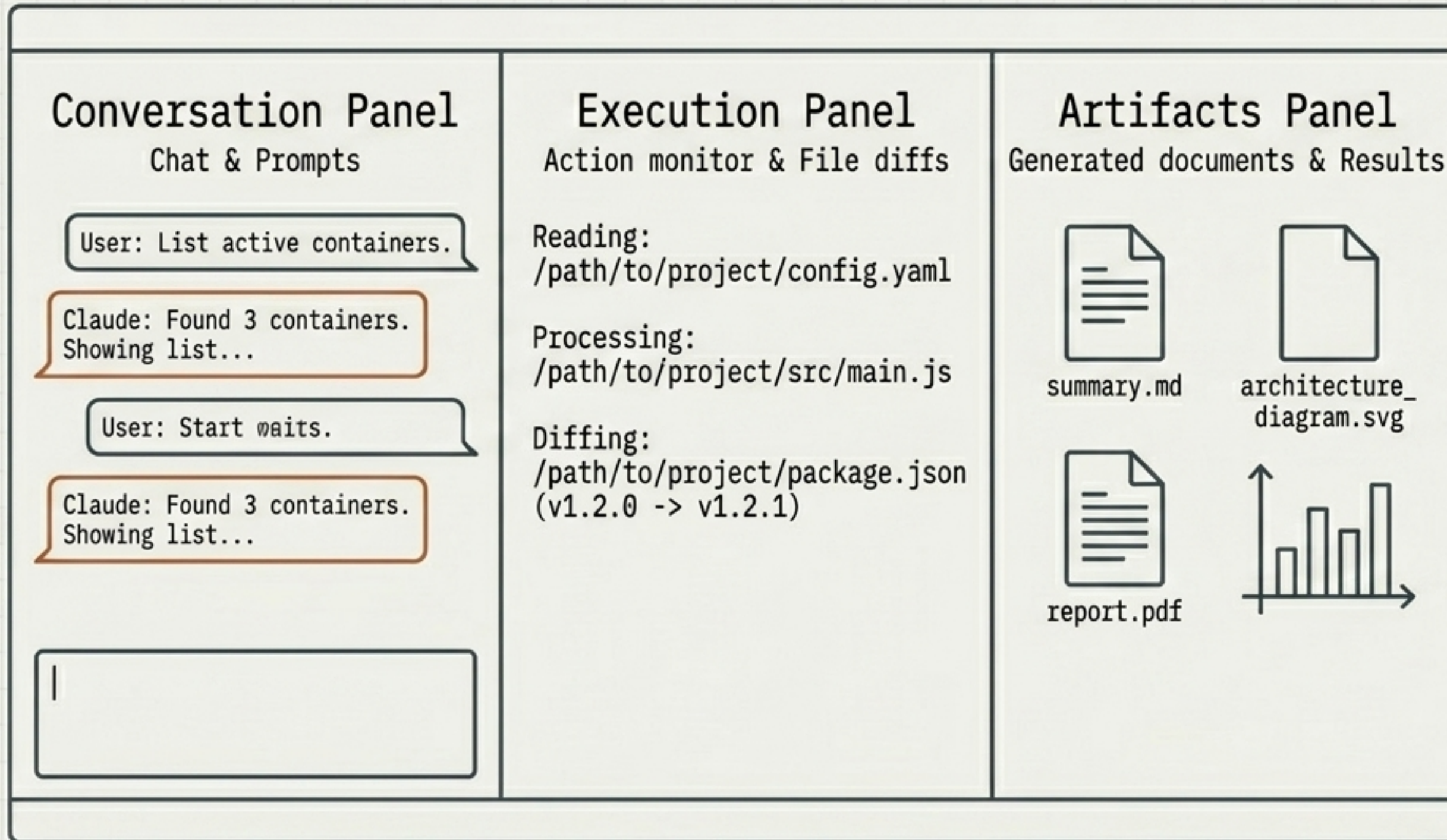


Scheduled Tasks (/loop)



Code executes strictly on local hardware; the remote device acts merely as an outbound-only message bridge.

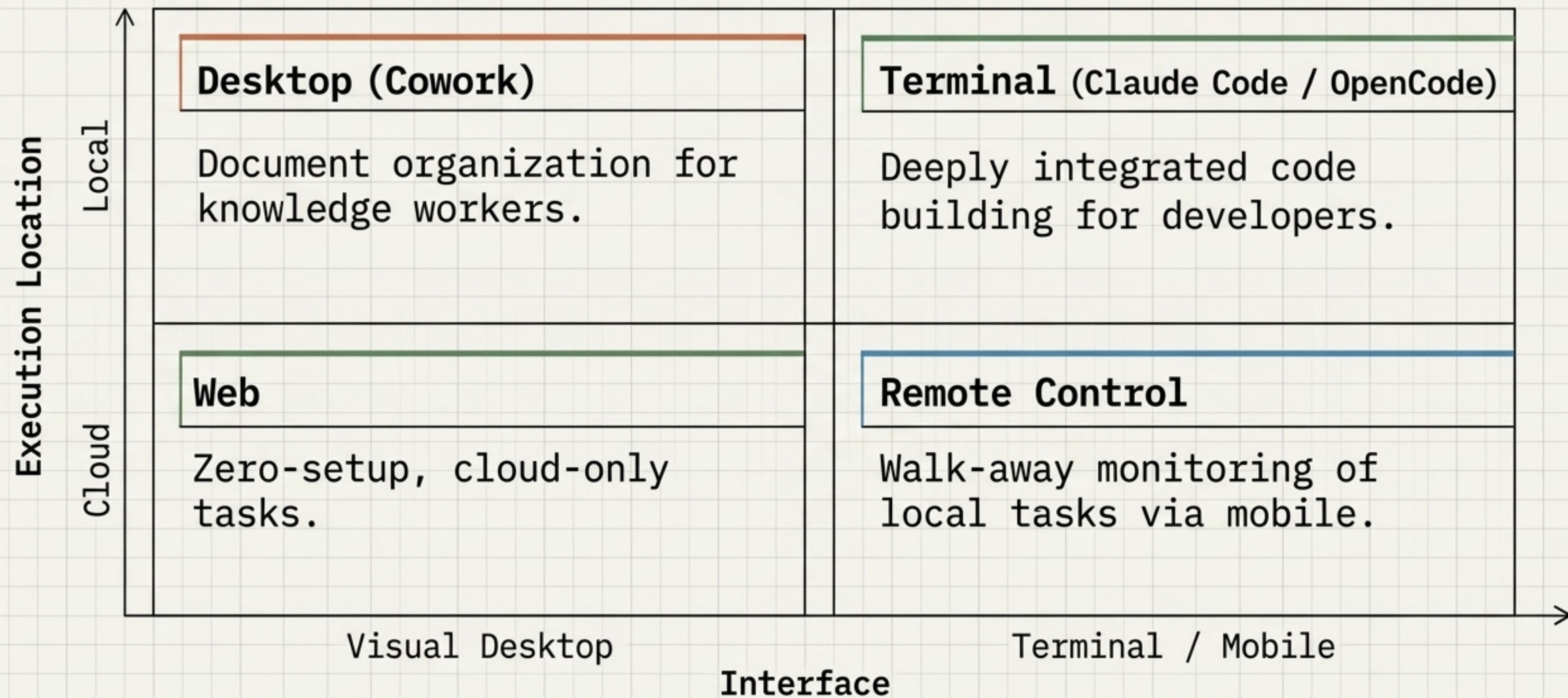
Claude Cowork: The Visual Desktop Agent



Security Boundary
Requires explicit, targeted folder access before any operation begins.

OODA-loop Foundation
The interface changes, but the OODA-loop foundation remains identical.

The Automation Capability Matrix



The interface adapts to the environment, but the underlying agentic engine remains the same.