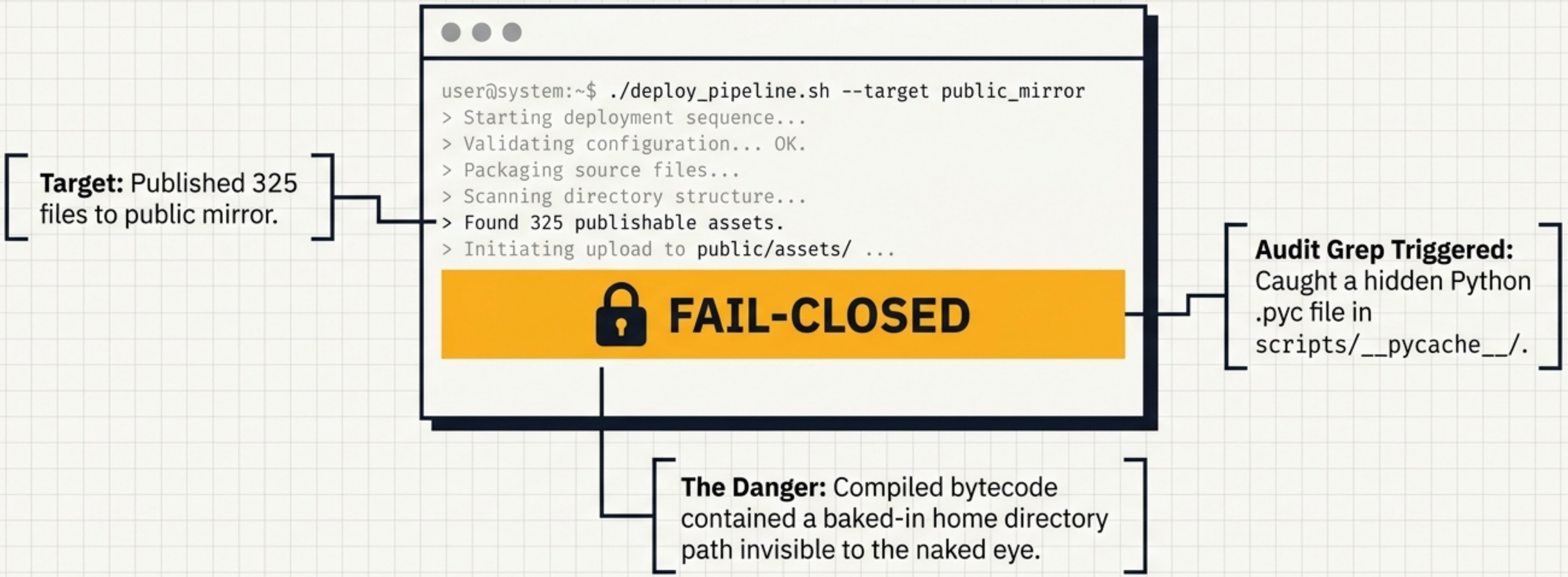
The AI Era Dotfiles

Building a Digital Workshop with ~/.claude

The system proved itself the moment it failed.



The configuration is interesting, but the automated audit-and-refuse mechanism around it is the real breakthrough. It protects the user from their own carelessness.

Six months of compounding configuration.

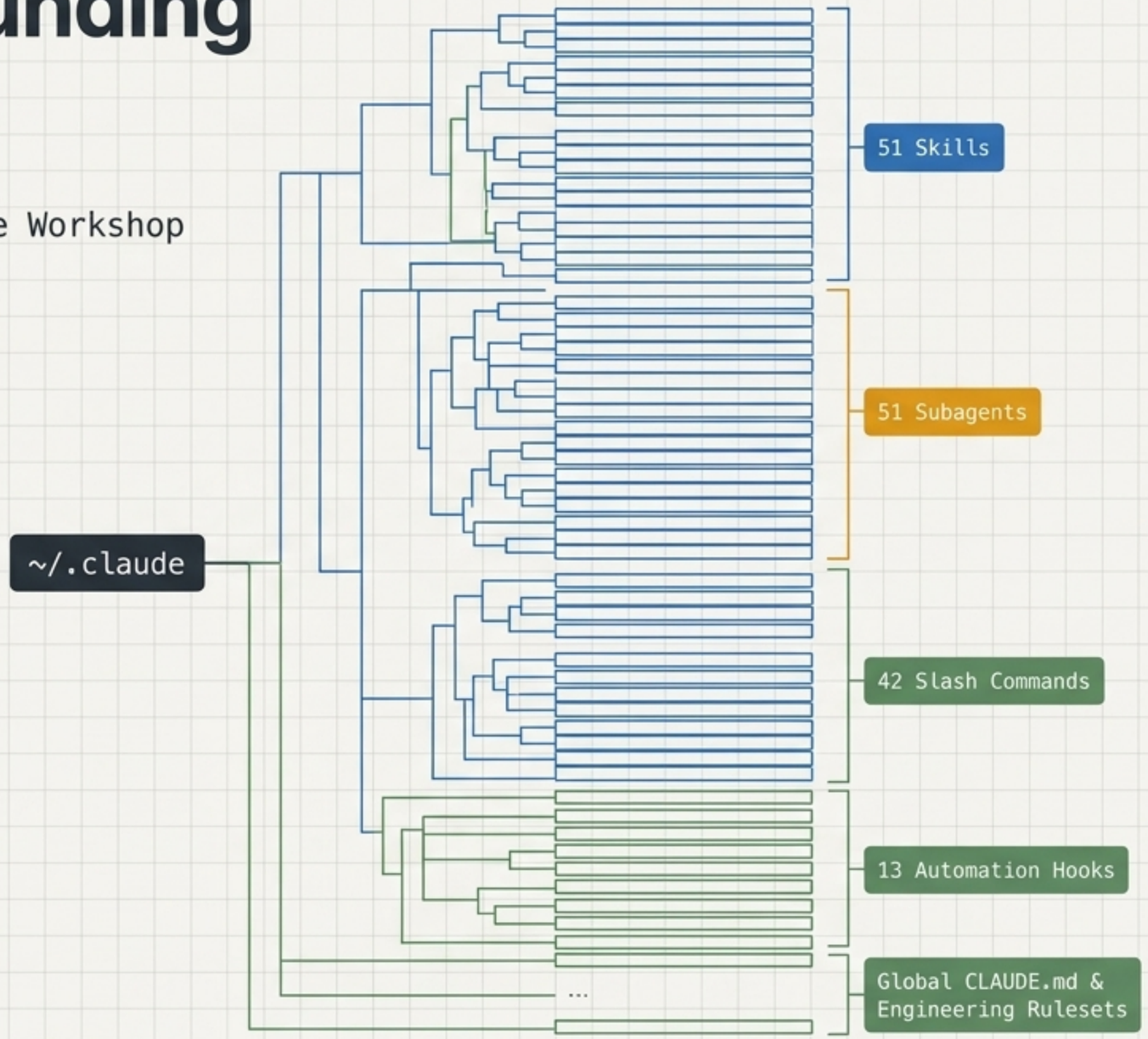
The Default

~/.claude

plugin

settings.json

The Workshop



The anatomy of a digital workshop.

Category: Writing

khazix-writer: 4-layer self-check to drag AI prose back to sounding alive.

narrative-research: 10k-word PDF report generator.

Category: Design

huashu-design: HTML hi-fi prototyping. Features baked-in philosophies (Sagmeister boldness) and anti-AI-slop checklists banning purple-pink gradients.

Category: Engineering

big-task: Workflow risk router.

audit-fix-loop & **pr-fix-loop**: Autonomous CI response.

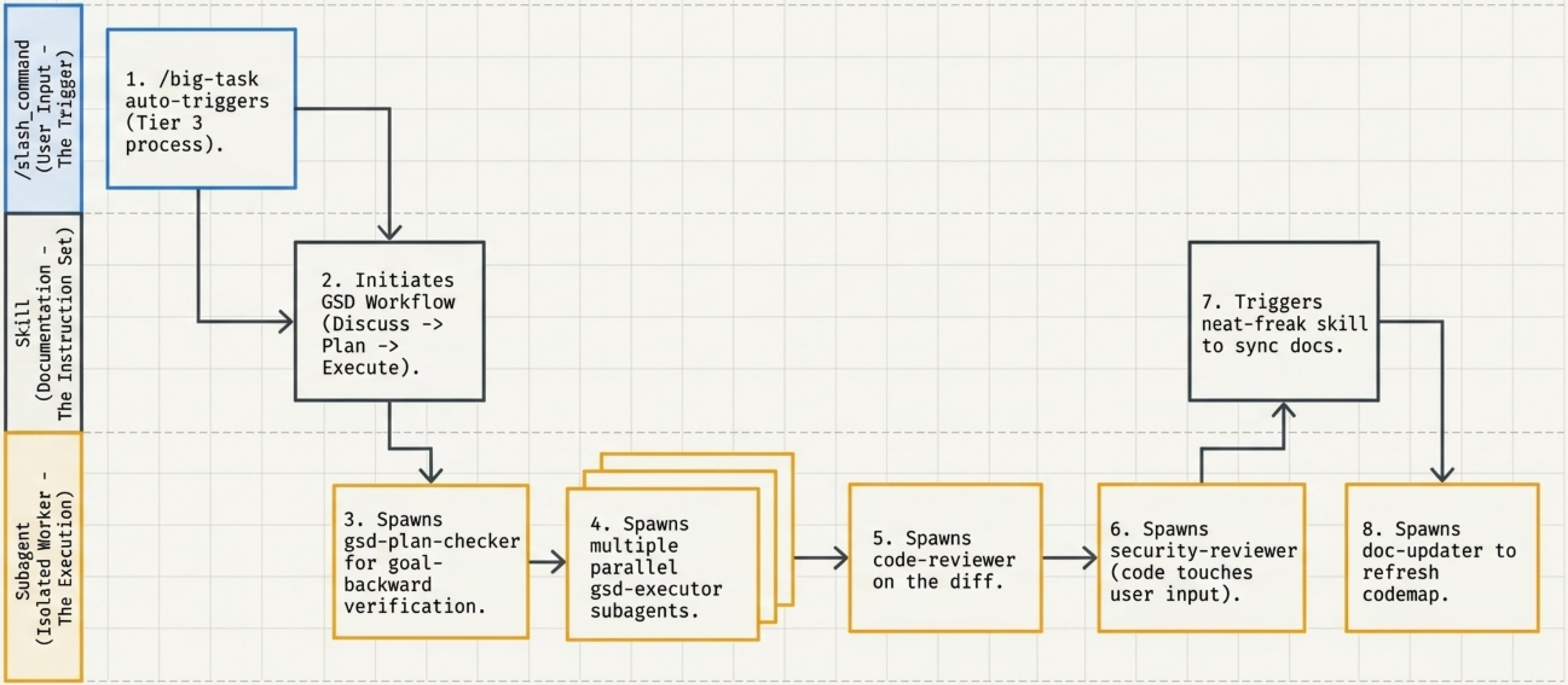
Category: Weird & Personal

banxian-skill: 3-system Chinese divination engine.

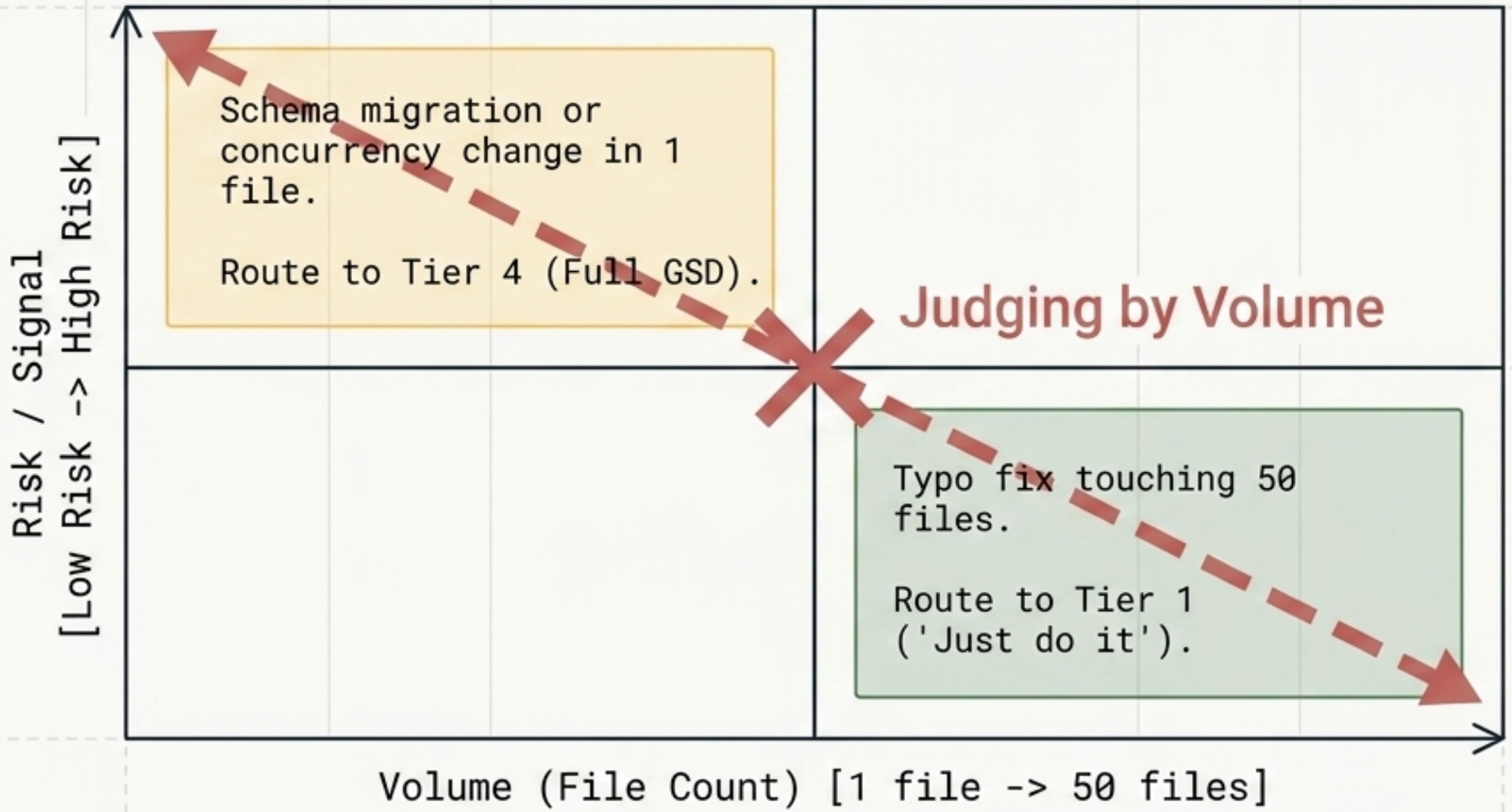
jewelry-marketing: Generating Xiaohongshu copy from a single photo.

One command spawns an entire autonomous pipeline.

Prompt: implement a new API endpoint



Routing tasks by signal, not by size.

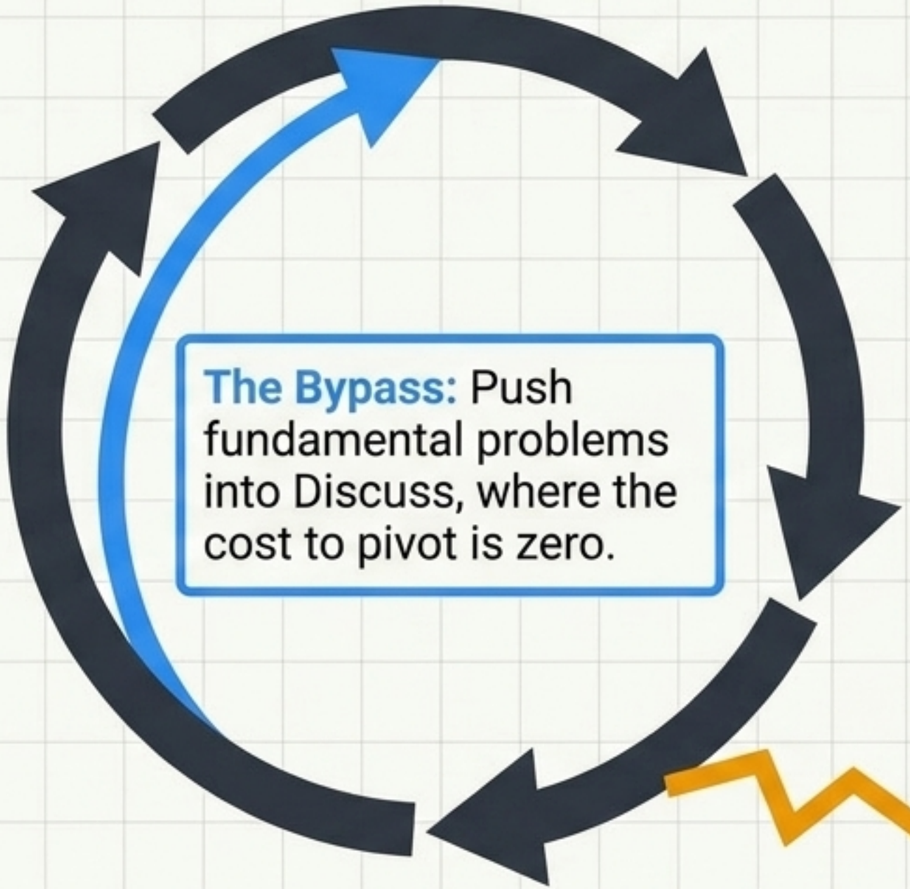


Behind every line in this routing table is a specific past mistake.

The GSD Loop: Forcing decisions when they cost nothing.

1. Discuss

- Surface gray areas (`GET vs POST, fail open vs closed`).



The Bypass: Push fundamental problems into Discuss, where the cost to pivot is zero.

2. Plan

- Write executable steps (`specific file, code snippet, command`). No placeholders.

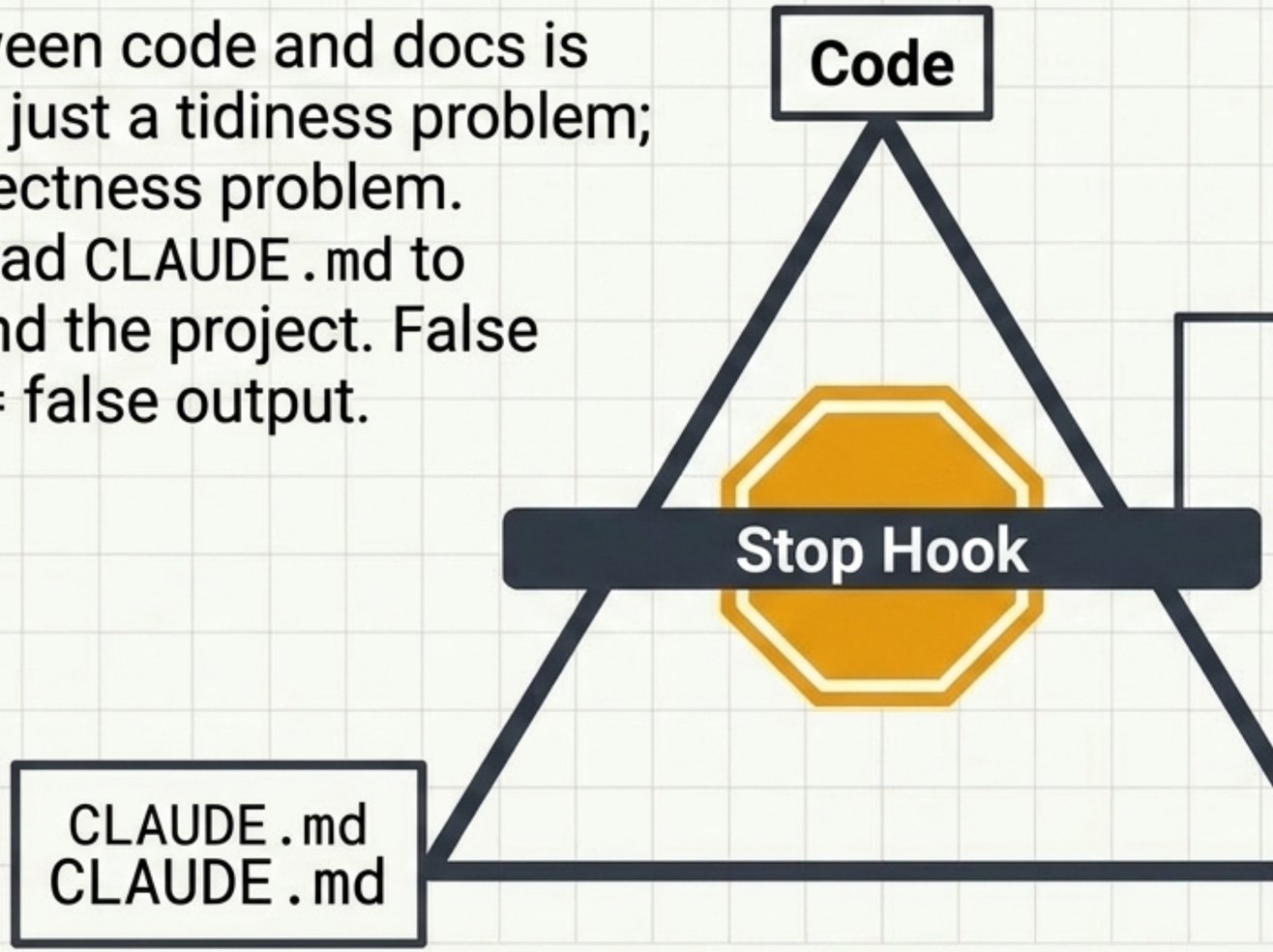
3. Execute

- Run steps, commit. If failure occurs, return to Discuss.

The Old Failure Mode:
`Start coding -> Hit fundamental problem at 80% -> Throw away work.`

The Neat-Freak Reconciler: Enforcing documentation hygiene.

Drift between code and docs is no longer just a tidiness problem; it's a correctness problem. Agents read `CLAUDE.md` to understand the project. False premise = false output.



Tollbooth Logic: After every meaningful change, the Stop Hook halts the session.

If code added a new env var, it checks the README.

No changes pass until all three points of the triangle are in alignment.

Why autonomous critique outperforms human review at scale.

Human Review

Glaze over at 1,000 lines of code.
Rubber-stamp the plan.
Bugs ensue.

1. Architecture / Correctness: Does it work? Are claims verified empirically?

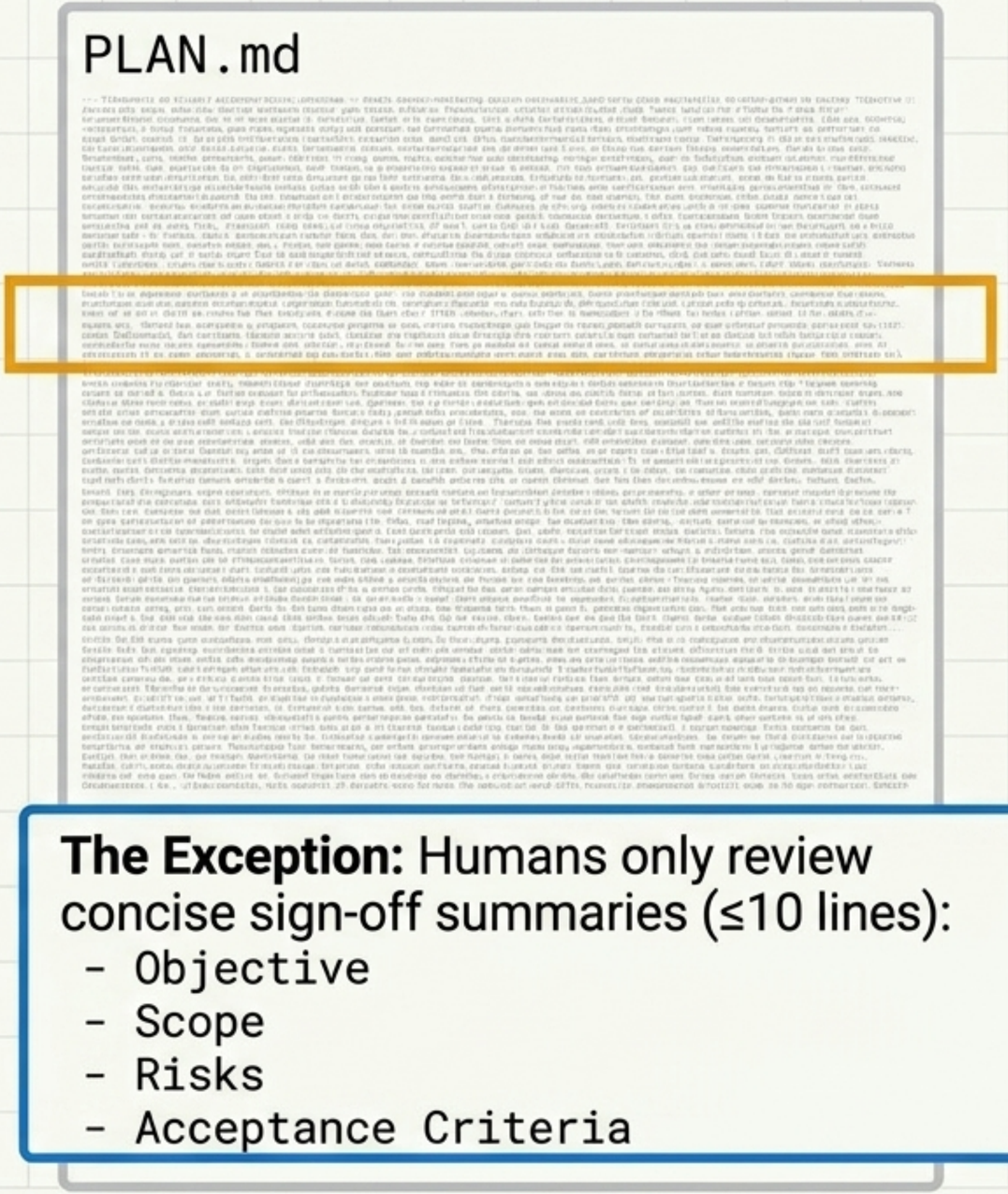
2. Security / Completeness: Are failure modes enumerated? Industry-standard checks present?

3. Process / UX: Will a real human actually use this without footguns?

Result: Caught an over-engineered URL token swap, missing gitleaks patterns, and undeclared sync directions that the initial self-review missed.

Rule: AX doesn’t review code.

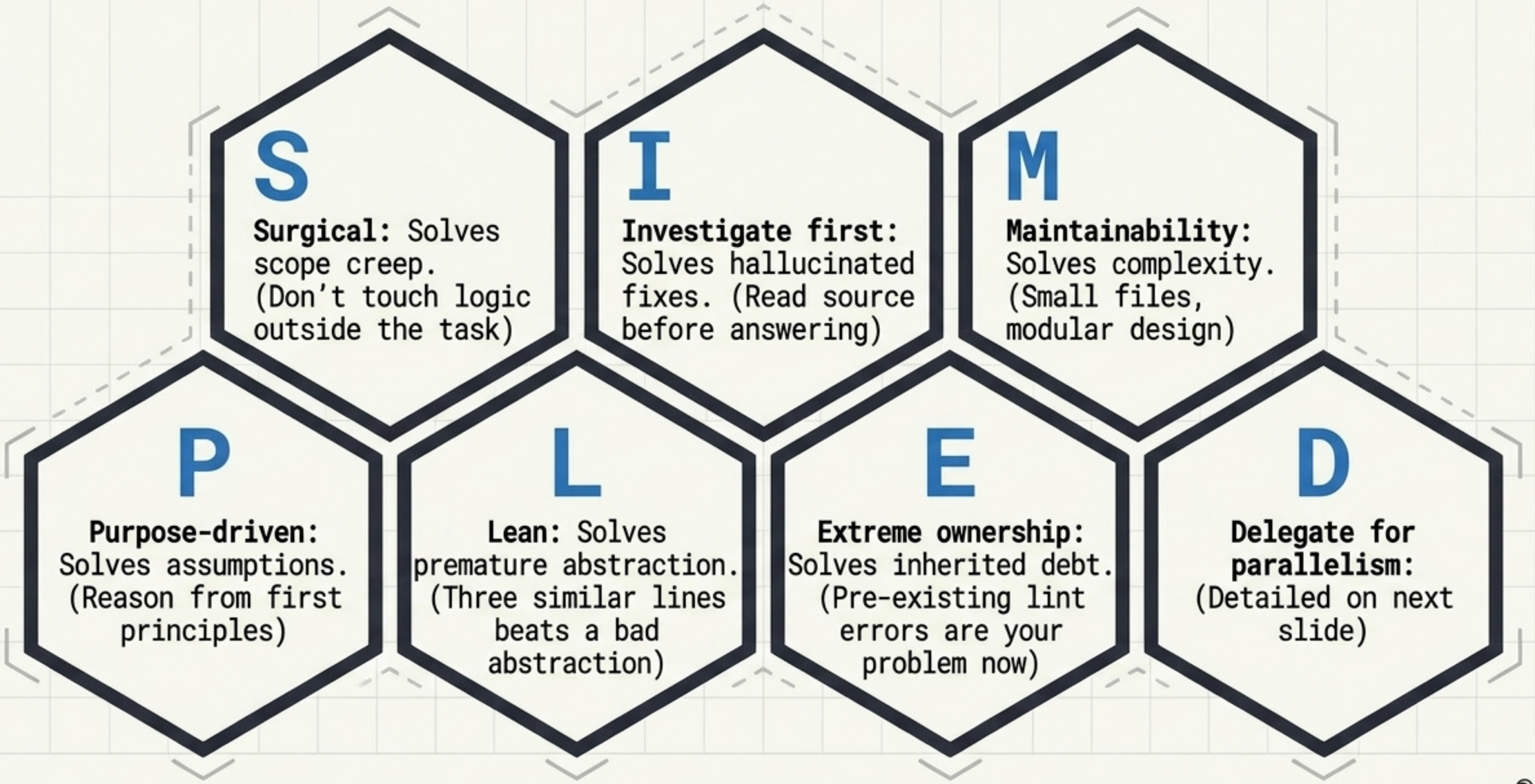
The agent runs multipass review on everything itself. There is no human safety net, forcing the agent's own review to be stricter.



The Exception: Humans only review concise sign-off summaries (≤10 lines):

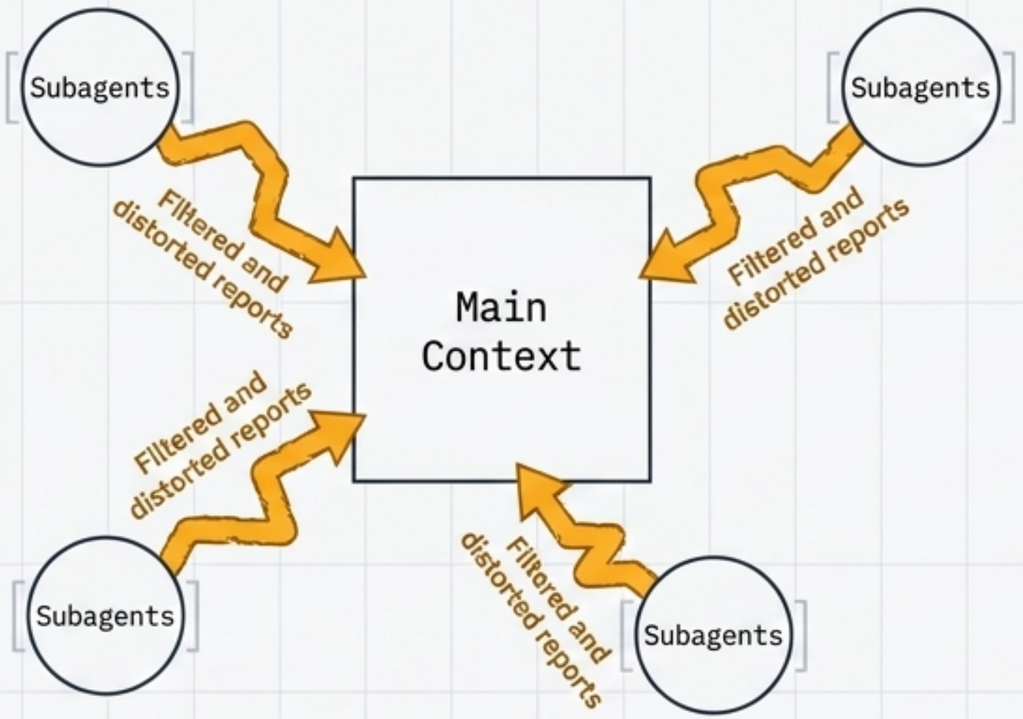
- Objective
- Scope
- Risks
- Acceptance Criteria

SIMPLED: A philosophy calibrated by past mistakes.



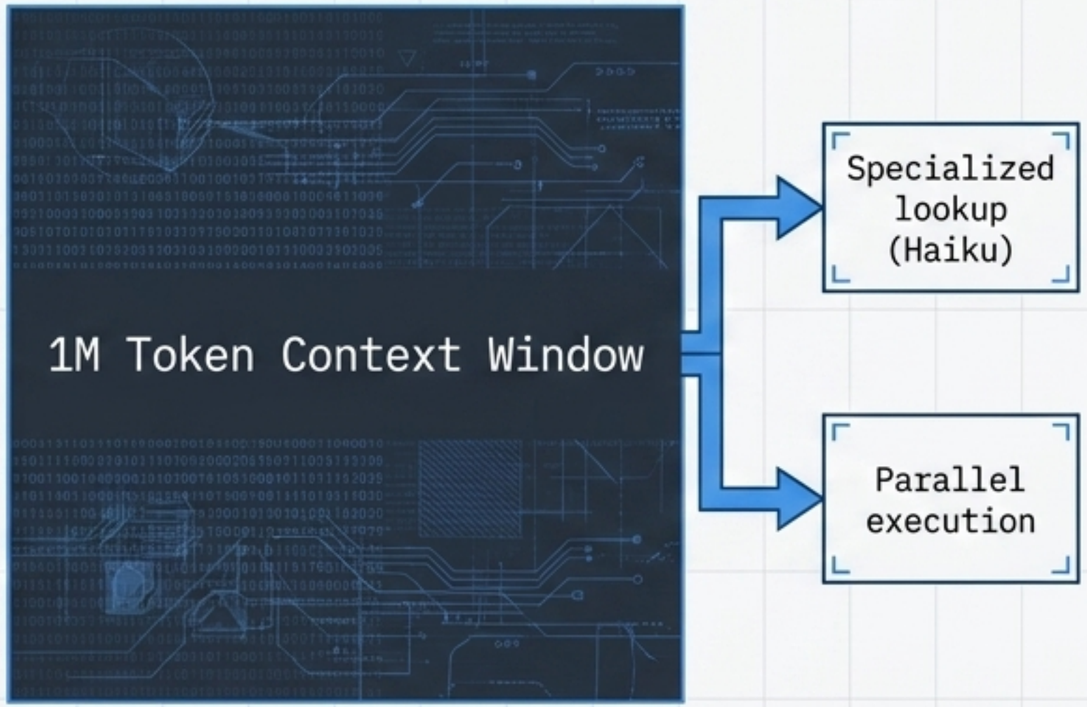
The Context Protection Fallacy.

The Fallacy



Key Insight: Spawning subagents outside to protect the main context backfired, requiring the user to re-derive missing details.

The Reality



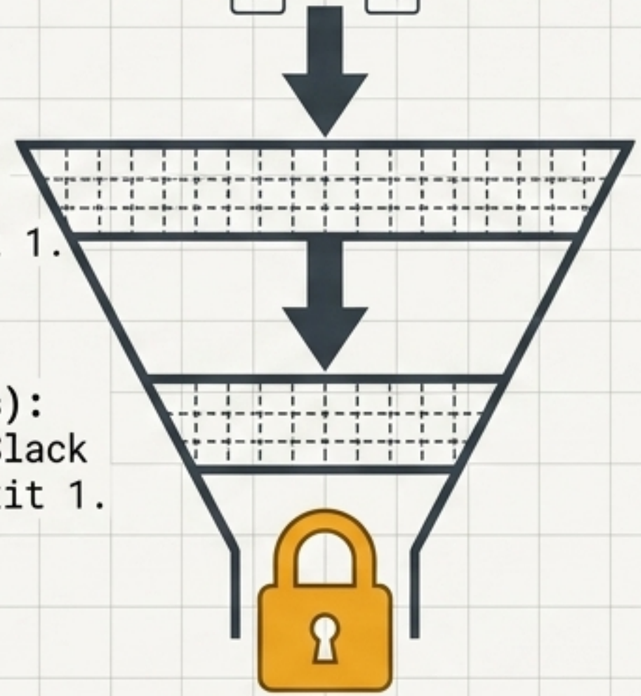
The Fix: The 1M context window is designed to hold whole sessions of noise. Protect it less, use it more.

The Sync Infrastructure: A two-tier fail-closed audit.

Input: sync-public.sh (325 files)

Filter Tier 1 (Identity Grep): Searches for xingfanxia, /Users/xingfanxia. Any hit = exit 1.

Filter Tier 2 (Gitleaks): Scans for ghp_*, sk-*, Slack tokens. Any finding = exit 1.



Fail-Closed Lock

Result: Caught 3 personal infrastructure leaks and a hidden bytecode path on day one. Refuses to push if either tier reports a single anomaly.

A reference architecture, not a turnkey product.

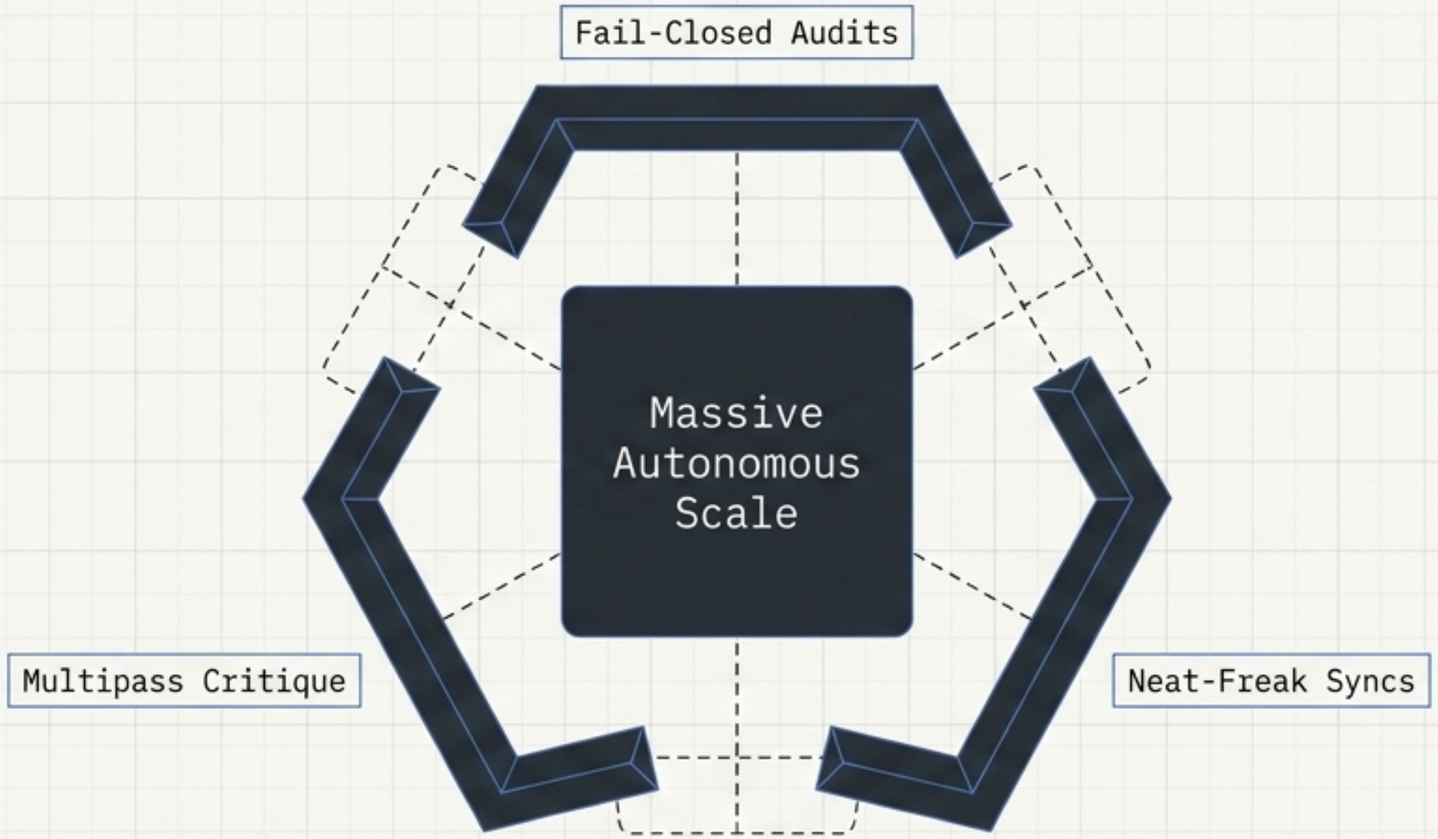
IS FOR

- Developers curious about heavy daily-driver Claude Code usage.
- Researchers studying emergent skill/subagent composition patterns.
- Builders looking for a reference configuration to fork and steal from.

IS NOT FOR

- People expecting git clone to yield a working setup (depends heavily on external APIs and MCP servers).
- People seeking universal ‘best practices’ (this is highly personal taste).
- People needing a beginner Claude tutorial.

The safety net is what enables the scale.



“High-volume autonomous output is only possible when surrounded by ruthless, automated friction.”

Key Insight: By building systems that explicitly protect the user from their own carelessness, AX successfully removed the human from the review loop. Strict automated friction is the prerequisite for massive AI output without catastrophic bugs.

The new shape of our tools.

The CLI Era

~/.vim
~/.zshrc

The Agent Era

~/.claude

Input Speed

Autonomous Output Quality

When the floor for AI tool access drops to zero, differentiation moves entirely to how you configure the tool.

~/.claude is to the agent era what ~/.vim was to the command-line era.
It is not a tool. It is the digital workshop that compresses six months of work into an hour.