Annoyance-Driven Development

How the creator of OpenClaw builds the tools that build the tools.

"Being annoyed is the very best way to solve problems."

The Bottleneck Has Relocated

The Old Job

Core Question: How do I build software faster?

Primary Output: Writing code and manual review.

Execution: Linear, single-threaded human effort.

The New Job

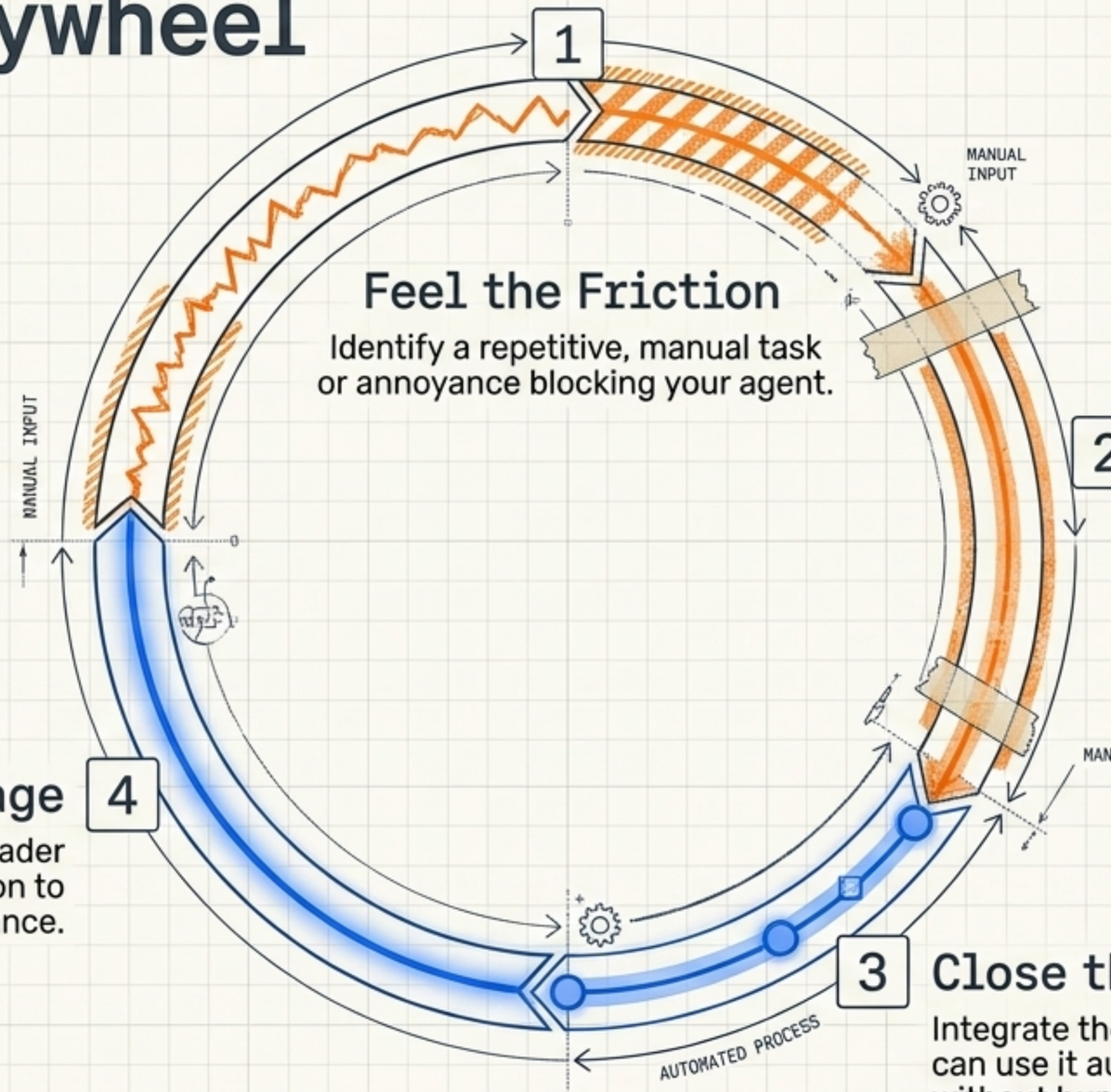
Core Question: How do I help my agent build software faster?

Primary Output: Building loops, context files, and triage tools.

Execution: Parallel, multi-agent autonomous loops.

The human is no longer the engine. The human is the mechanic.

The Slop Flywheel



Feel the Friction

Identify a repetitive, manual task or annoyance blocking your agent.

2 Build the Slop

Write a fast, ugly, unpolished internal helper tool. Polish doesn't matter.

3 Close the Loop

Integrate the tool so the agent can use it autonomously without human babysitting.

4 Compound Leverage

Stack the tool into the broader infrastructure and move on to the next annoyance.

Annoyance #1: The 10,000 Issue Backlog

The Friction

Pull requests and issues piling up faster than maintainers can possibly read them.

The Slop Fix

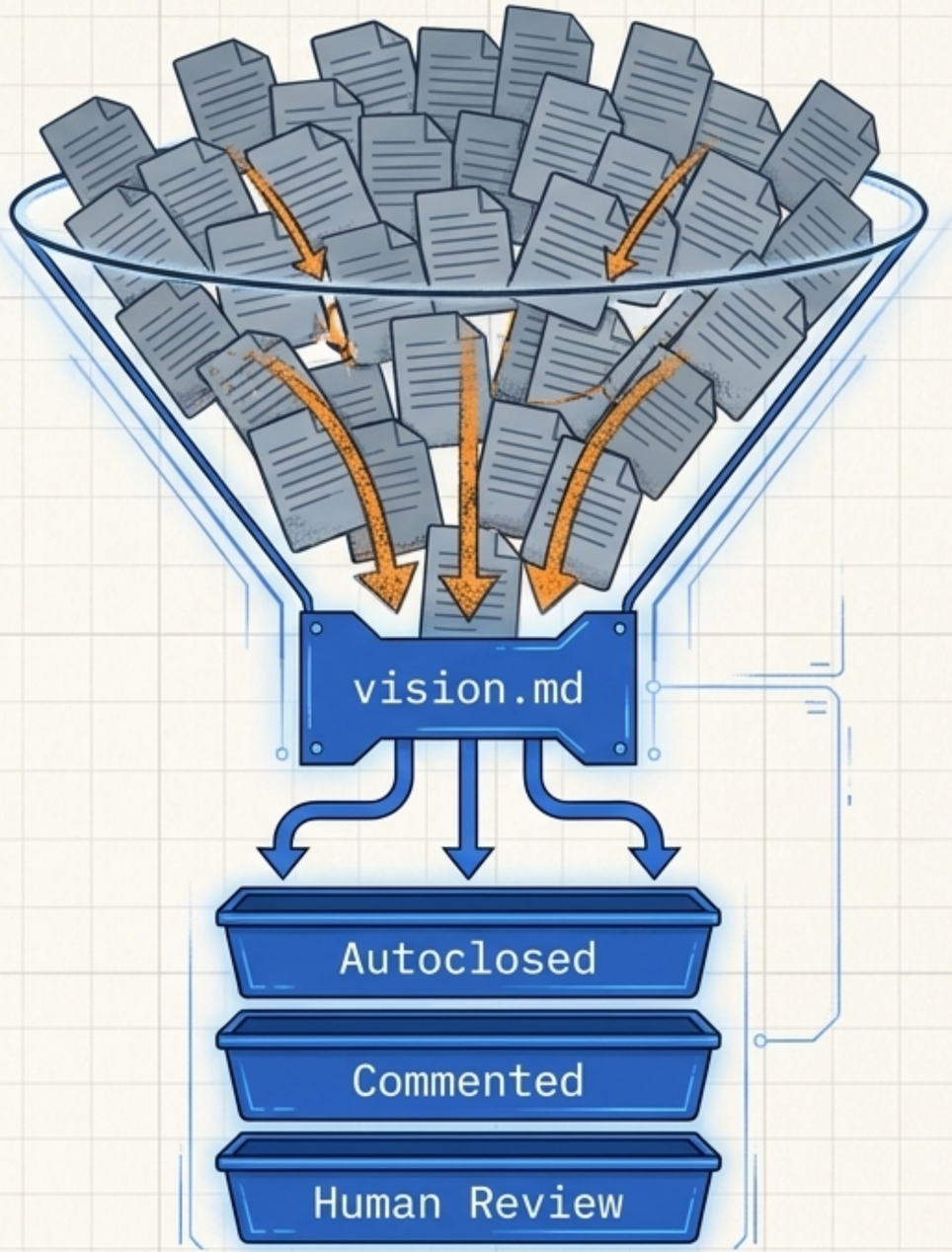
ClawSweeper. A maintenance bot treating issues as prompt requests.

The Mechanic's Secret

Equip agents with a **vision.md** file detailing strict project invariants.

Compounding Loop

Run the agent weekly across all old issues to retroactively recognize and close fixed bugs.



Annoyance #2: What Do People Actually Want?

The Friction

Losing the actual user signal inside the noise of infinite PRs and generic feature requests.

The Slop Fix

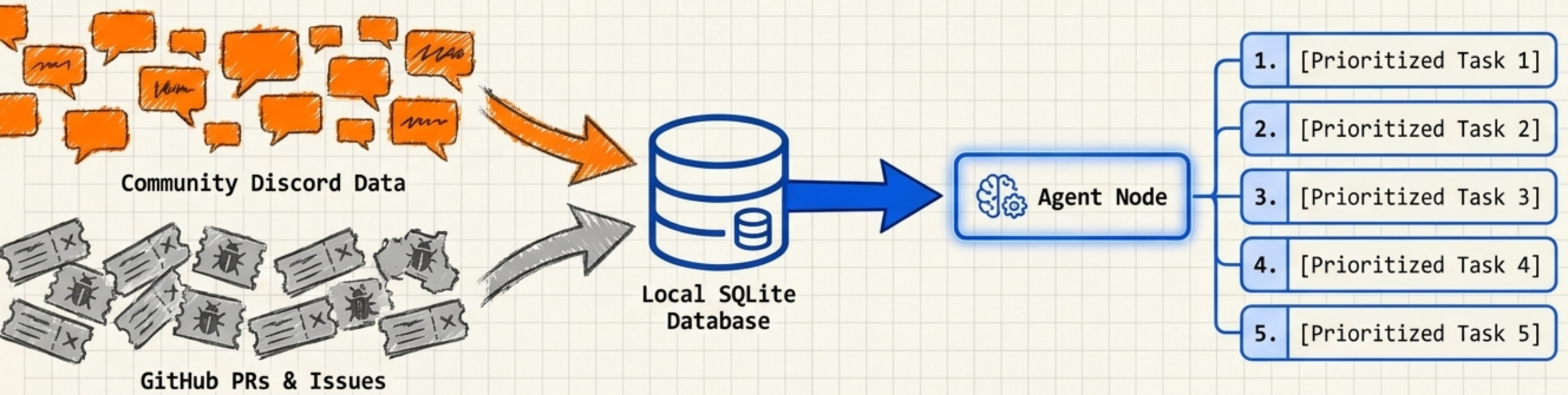
Discrawl. An aggregation crawler that doesn't just look at code repos.

The Mechanic's Secret

Mirror `community Discord` data into `local SQLite` to extract the raw product pain.

Compounding Loop

The agent cross-references user screaming with the `vision.md` file to autonomously output a prioritized daily top-5 task list.



Annoyance #3: The API Rate Limit Wall

The Friction:

Agents hit rate limits, paralyzing the entire development workflow.

The Slop Fix:

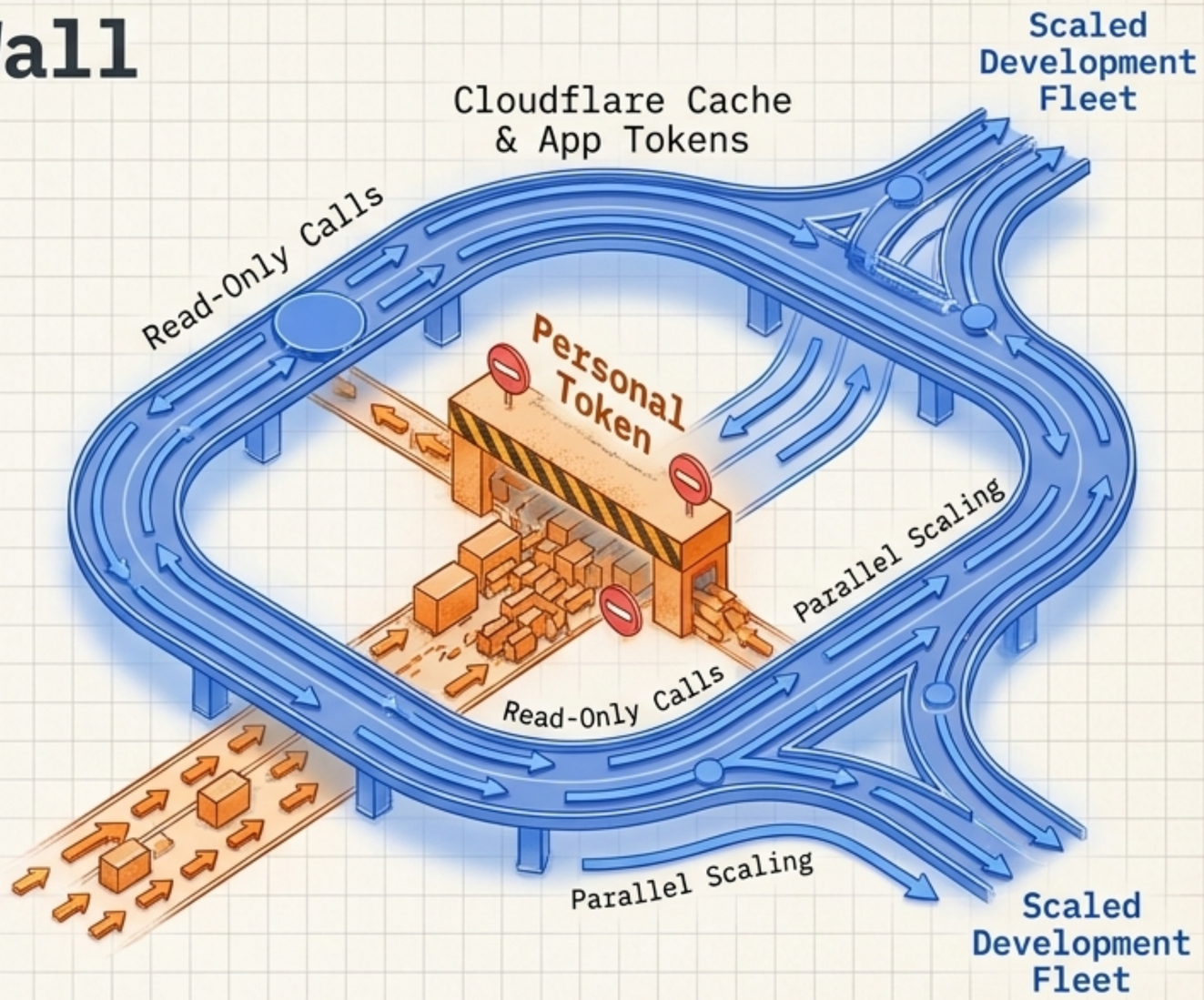
Octopool. A self-hosted relay that pools API identities.

The Mechanic's Secret:

Rate limits are not a law of physics. Route around them. Pool App tokens for read-only calls and save personal tokens for mutations.

Compounding Loop:

All agents in the fleet hit a shared cache, allowing massive parallel scaling without token starvation.



Annoyance #4: Melting Laptops

The Friction

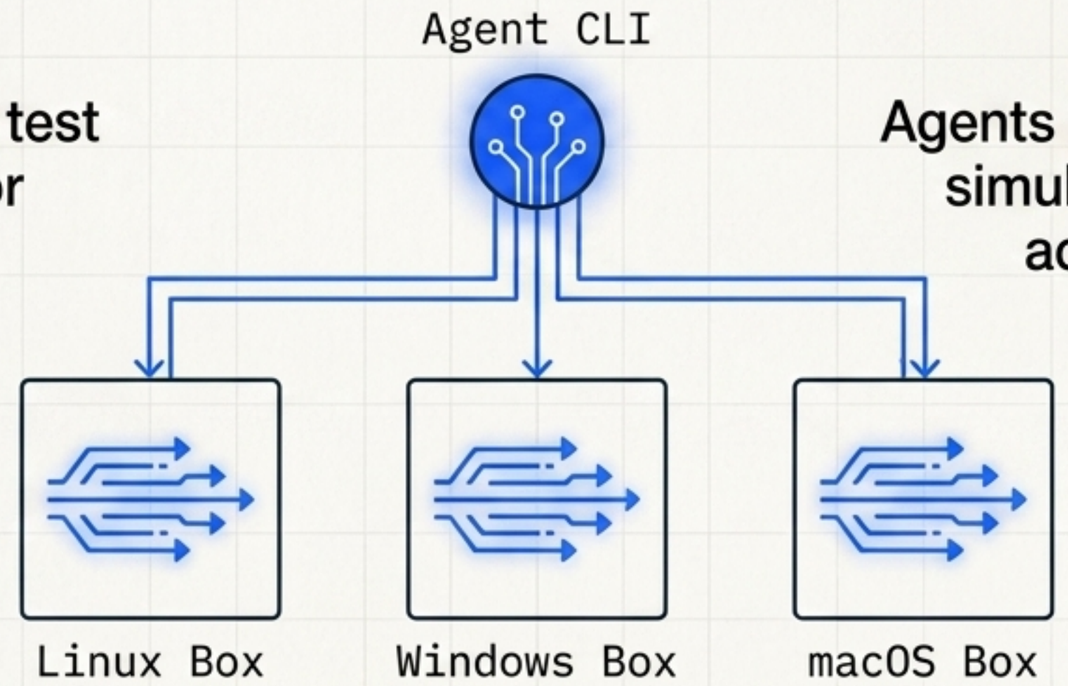
Heavy test execution grinds local machines to a halt, limiting agent speed.

The Mechanic's Secret

Turn testing infrastructure into agent infrastructure. Tell the agent to warm a box, sync the diff, and run the suite remotely.

The Slop Fix

Crabbox. Elastic, remote test boxes built specifically for agents.



Compounding Loop

Agents spin up OS environments simultaneously, proving the fix across platforms in parallel without human setup.

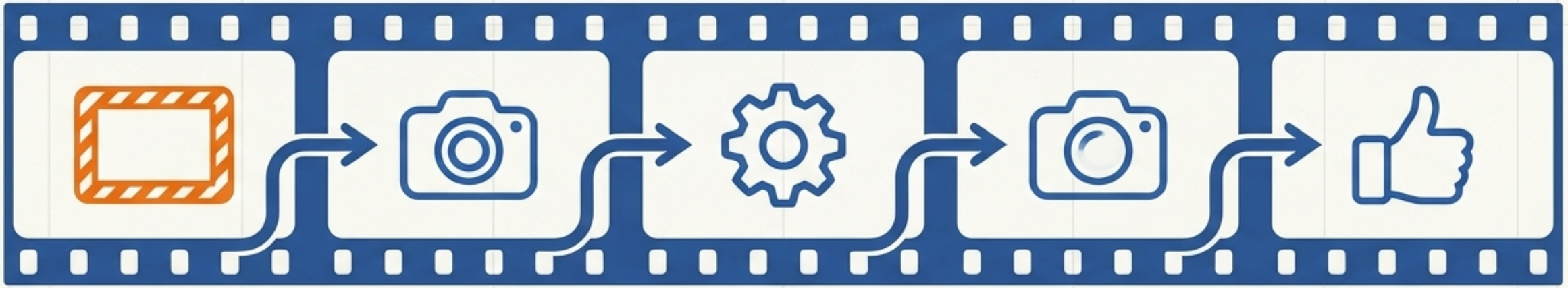
Annoyance #5: Trust Me, The Test Passed

The Friction: If an agent only edits source code, a human still carries the end-to-end UI verification cost.

The Slop Fix: Mantis. UI and browser automation tied directly to the agent workspace.

The Mechanic's Secret: Give the agent browser access and visual tools. Force them to visually recreate the bug and record the fix.

Compounding Loop: The human role shifts from manual reproduction to visual approval.



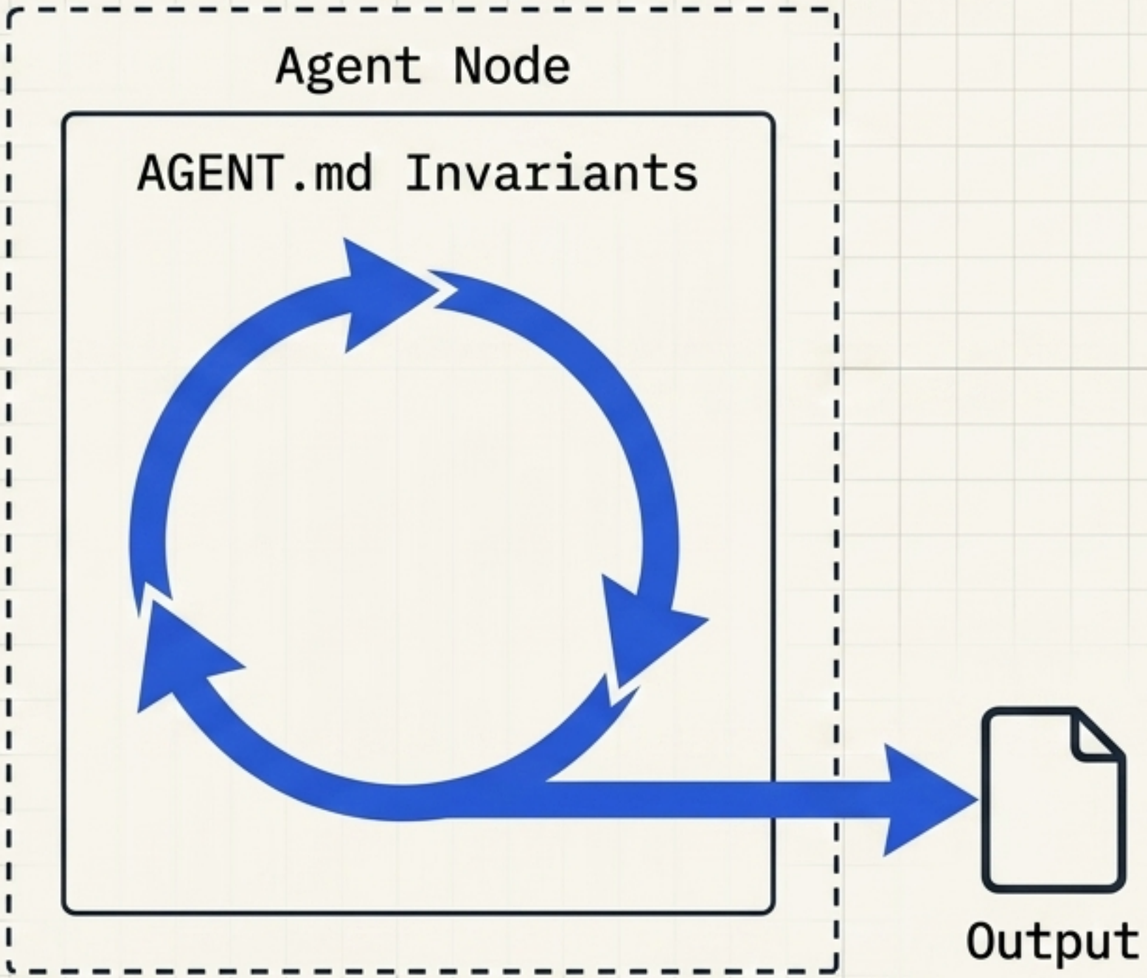
Annoyance #6: The Infinite /Review Loop

The Friction
Humans manually prompting agents to /review their code over and over, waiting 20 minutes for serial fixes.

The Slop Fix
Autoreview. Automated, iterative self-review via CLI.

The Mechanic's Secret
Treat review as a multi-pass automated process. Force the agent to repeatedly call a fresh-context review on itself before committing.

Compounding Loop
The agent filters out defensive, useless suggestions and only surfaces a hardened, pre-reviewed PR.



Annoyance #7: The Black Box PR

The Friction: Reviewers cannot tell if a PR came from a sloppy four-word prompt or a rigorous two-hour iterative session.

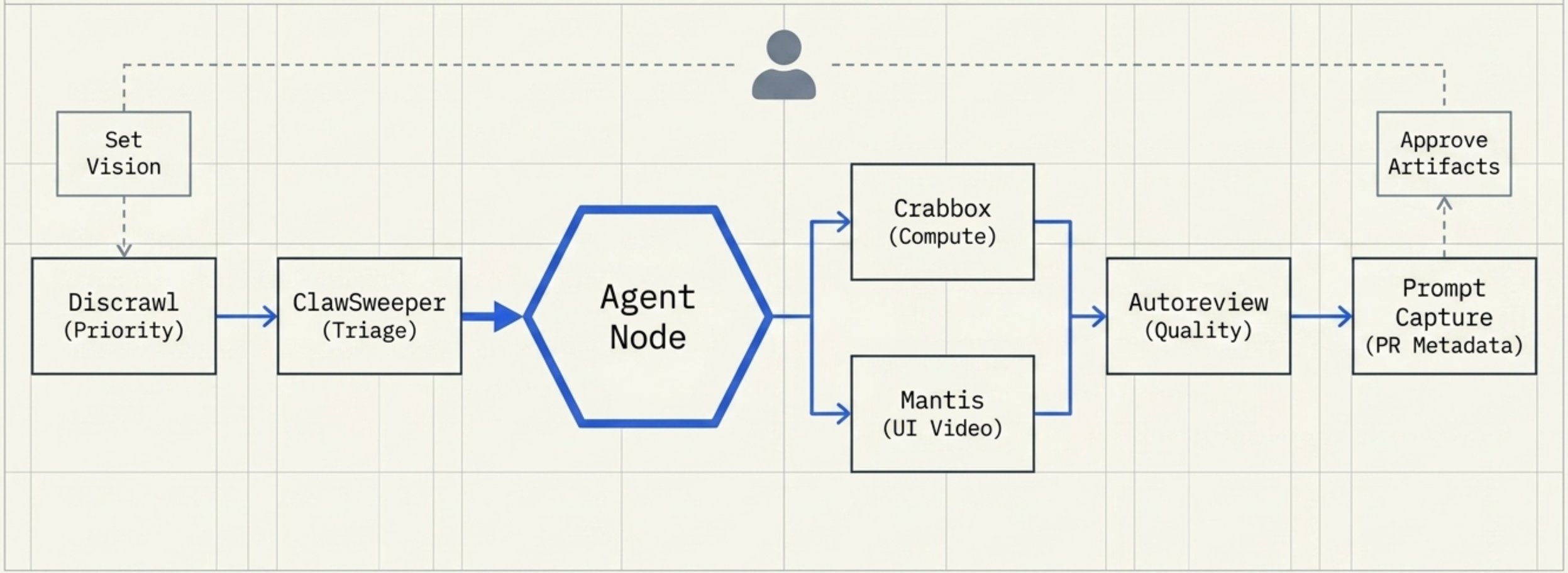
The Slop Fix: Prompt Capture. Auto-attaching local session histories directly to commits.

The Mechanic's Secret: Treat agent session provenance as first-class engineering metadata, not disposable chat exhaust.

Compounding Loop: Maintainers instantly calibrate their scrutiny based on the depth of the attached agent transcript.

Synthesis: The Agent Operating System

Agent-era leverage does not come from better prompts. It comes from better loops.



Every tool compounds on the others. The human sits at the edge, defining tracks and watching telemetry.

The Reality Check

Do This / The Tracks

- Turn recurring pain and manual reviews into agent infrastructure.
- Use explicit, strict vision.md and AGENT.md files to bind agent behavior.
- Build local-first, durable state (SQLite, local caches) to route around SaaS rate limits.

Not That / The Trap

- Don't expect raw LLM code generation to scale without verification loops.
- Don't apply the 'slop is fine' rule to production customer-facing code.
- Don't rely entirely on hosted chat environments for complex, multi-agent workflows.

Build the thing that builds the thing.

Your job is no longer to be the engine. Your job is to build the tracks. Pay attention to wherever the human is acting as the glue between systems—and automate it.