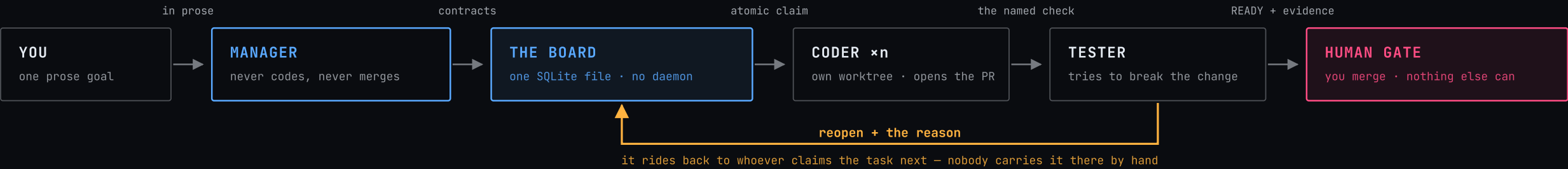


One terminal over every agent you have running.

Type **asq** and you get your projects on the left and, on the right, a **manager** agent you task in prose. It plans, spawns coders, testers and reviewers, and loops until the goal is met — then stops, because merging is yours. Click any agent and you are inside its real Claude Code session, typing at it directly. Nothing is relayed or re-rendered as a chat.

```
$ curl -fsSL https://raw.githubusercontent.com/AISquare-Studio/aisquare-cli/main/install.sh | sh
```

macOS, Linux and WSL2. It installs only what is missing and ends by offering to open the UI. Run it again and it installs nothing.



The manager talks to its agents only through the board. That is the whole coordination model: no relaying, no forwarding, nothing to keep in your head.

Every session starts cold. You re-explain the codebase, the conventions, and the thing you already told it last week — and pay for it again in tokens and in patience. Memory. Five fail-open lifecycle hooks start each session with a packed snapshot of the repo, your context entries and this project's prompt history. Zero commands after setup.	You are the message bus. One agent per terminal. Five agents means five mental models and a human copy-pasting findings between them — which is the part that does not scale. A shared board. Contract-carrying tasks, atomic leased claims, dependency holds, and a per-prompt delta of what teammates did. Nothing needs forwarding.	Nothing is on record. When an agent did something you did not want, you cannot read back what it was shown, what it decided, or what it cost — so the same mistake arrives again next month. Explainability. Every session becomes a Run you can read — prompts, tool calls, tokens, cost, plus your own prompts and board events. Opt-in, and never blocking.
---	--	--

WHY YOU CAN PUT THIS IN A REAL REPOSITORY

Fail-open by construction Any error inside a hook is swallowed and the session continues untouched. A repo that never opts in sees nothing at all.	One file, no daemon All state is ~/.aisquare/context.db in WAL mode, concurrency-tested against racing parallel sessions.	Everything is a command Every action the UI takes is a plain CLI command, and every one takes --json. No script ever meets a full-screen app.	Receipts you can re-prove Writes print ✓ ... seq N; team verify 42 asks the board whether it is really there and exits 0 or 1.	Roadmap stays hidden Unfinished commands are hidden from --help and exit 70 saying so, rather than half-working. What is listed, works.	Contracts, not vibes The server's JSON Schemas are vendored byte-for-byte and every request the CLI can emit is validated against them in CI.
--	---	---	--	---	---

WHERE THIS IS GOING, STATED PLAINLY

● NOW · V0.6.0 — The fleet UI and the manager loop, end to end — Memory, packed snapshots, prompt history — Board: contracts, leased claims, receipts, signals — Model ladder per role, probed · several accounts, one board	● NEXT — Sign in — one identity across your machines — sync: memory that follows you, and your team — On-limit handoff — a seat runs out, the work moves — Remote control — watch and steer off-terminal	○ LATER — Connectors — meeting notes and tickets into the knowledge base — Organisation policy the agents cannot cross — A knowledge graph of findings, decisions and preferences
---	---	--

A PILOT IS ONE REPO AND ONE AFTERNOON

```
$ curl -fsSL .../install.sh | sh # only the gaps get installed
$ asq # click + and point it at a repo
... press Start manager, and type the goal in prose.
```

WHAT IT NEEDS FROM YOU

A Claude Code login your developers already have, tmux 3.2+, git and gh. Nothing to deploy, no account to create, and no traffic leaving the machine unless you switch that on deliberately.

NEXT STEP

Pick one repository with real parallel work in it. We run the loop on a live goal with your team, then hand back the transcript, the board and the Runs.

V0.6.0 · SHIPPINGMIT · LOCAL-FIRSTONE SHEET · A3

PAYMENTS-API · LIVE

🗺️ manager ▶

🔧 coder-idempotency ▶

🔧 coder-webhooks 🔔 NEEDS YOU

🔧 tester-1 ▶

👁️ reviewer-1 ▮

🔧 coder-retries 🔄 PR #412

Six real Claude Code sessions, one project, one window. Close the UI and every one of them keeps running — tmux is the substrate, not a rendering trick.

5roles in the loop

1SQLite file of state

0daemons · accounts