

Conductor

By Austen Tucker

Why I built it

Agent work gets risky when the plan, authority, execution, and evidence collapse into one conversation. I wanted a local system where a person can see the Story, inspect the plan, approve a specific scope, watch bounded work, and review a receipt before anything is called done.

Conductor formalizes the same staged-authority and evidence model that projects such as WizWor taught me. It did not orchestrate the WizWor launch. It turns those lessons into reusable Lab infrastructure for later Codex and Claude work.

Build snapshot

- Interface: React and Vite with a Jira-shaped backlog, Board, issue detail, Activity, and settings
- Control plane: Local Node and Express service with service-owned SQLite, WAL mode, and backups
- Agent runtime: Per-project Codex or Claude planner, worker, and evaluator roles in bounded workspaces
- Authority: Dispatch off by default, exact plan-hash approval, explicit review, and recorded receipts

What I did

I made work state visible

Projects, Epics, Stories, plans, approvals, runs, receipts, and activity live in a local database rather than disappearing into agent transcripts. The board shows when a Story is waiting for a plan, queued, running, blocked, in review, or accepted.

I bound approval to an immutable plan hash

Moving a Story to To Do creates a read-only planning run. Approval checks that the plan is still the latest version and that its stored body still produces the approved SHA-256 hash. A changed or superseded plan cannot borrow authority from an older yes.

I bounded execution

The supervisor caps concurrent workers, permits only one non-planning run per project, uses role-specific timeouts, supports cancellation, and marks live runs interrupted after a restart instead of silently relaunching them. Repository writes happen in an isolated worktree unless a project explicitly chooses a shared workspace.

I separated completion from acceptance

A successful worker stops in In Review. Higher-risk work can receive a separate, read-only evaluator pass. Only explicit user acceptance moves the Story to Done. Interactive sessions write an atomic completion receipt inside the workspace; a host-side launcher validates and submits it after the agent exits.

I kept consequential actions outside repository-write authority

The current worker grant is scoped to local repository work. Its contract explicitly excludes merge, deployment, publication, material deletion, spending, and external messages. Those actions require a separate action-scope approval instead of being smuggled inside “implement this plan.”

What I learned

Authority needs stages

Approving a goal is not the same as approving a plan, and approving repository edits is not permission to merge or publish them. Naming those stages makes both humans and agents less likely to over-read a vague instruction.

Evidence has to cross the trust boundary

A final chat message is not enough. Structured receipts, verification mappings, persisted run state, and host-side submission make it possible to ask what completed, what was actually checked, and whether the handoff is ready for review.

The project board is part of the safety system

A familiar backlog and Board are not decoration. They expose blocked work, pending approval, queue state, and review status without requiring someone to reconstruct the workflow from terminal output.

Provider abstractions still leak

Codex and Claude differ in sandbox flags, output capture, usage reporting, and interactive-session behavior. A shared role model helps, but provider-specific launchers and receipts still need direct tests and conservative status reporting.

Current best

Conductor is a working local prototype on the feature/claude-code-session-from-board branch. It proves the core loop: plan read-only, approve the exact hash, execute within a bounded workspace, collect evidence, stop in review, and wait for a person to accept the result.

Limits and open questions

- Conductor is a local prototype and internal orchestration system, not a hosted production platform.
- The current approval endpoint issues repository-write authority. The UI and API for granting additional action scopes are not complete.
- Provider token metrics are exercised through fake Codex and Claude harnesses; real-provider metric coverage remains incomplete.
- Browser tests verify the session controls without launching real interactive agents, so the live multi-provider harness still has coverage gaps.
- No public adoption, reliability, or business-impact result is claimed.
- A permission-safe, publication-ready screenshot still needs to be captured from demo data.