



See what every CLI agent is doing, while it runs

A hook script, a local event hub, and a dashboard that nests runs by repository, session and subagent.



Sytze P.
Solution Engineer

The problem

An agent run is a terminal scroll. Two runs in two repositories are two terminal scrolls, and neither tells you which one is stuck or which one has quietly spawned three subagents.

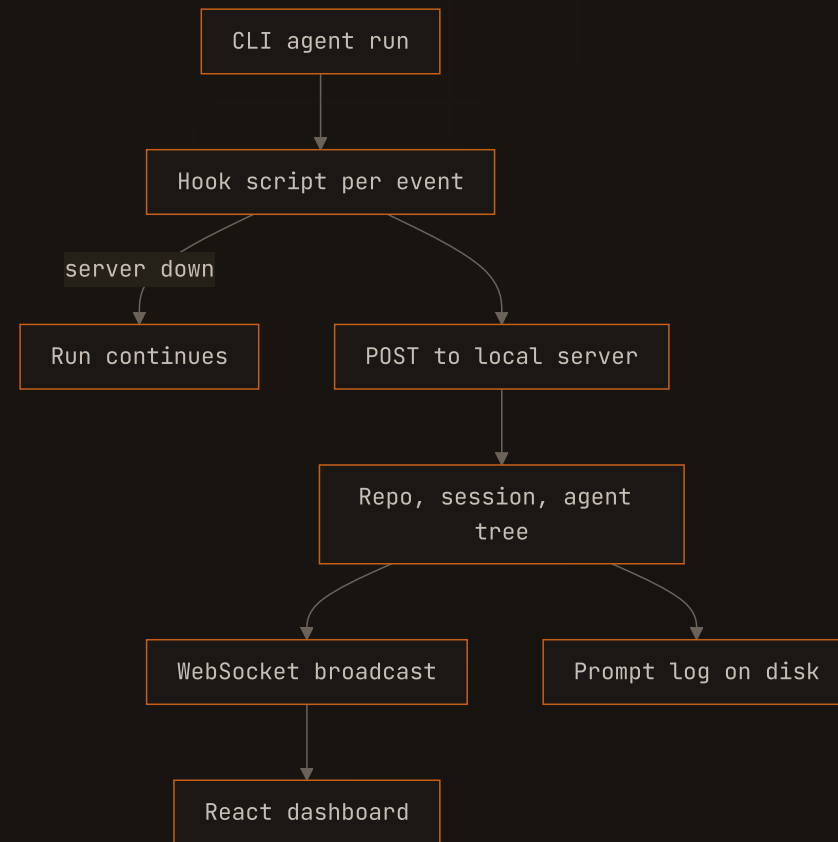
The agent is not your program either. You cannot add logging inside a CLI that someone else ships, so any visibility has to come from outside it.

• APPROACH

Hook event to live dashboard

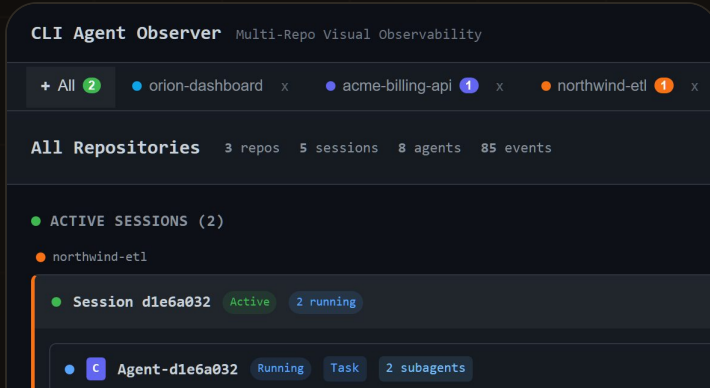
The CLI already fires hooks on its own lifecycle, so the observer registers a small script against them rather than wrapping or proxying anything. Each event posts to a local server that holds the repository, session and agent tree in memory and pushes updates to the dashboard over a WebSocket.

That send is deliberately fire and forget. If the server is down the hook errors are swallowed and the run carries on, so the monitor can never take out the tool it monitors. Claude Code and Gemini CLI name their events and tools differently, and both are mapped to one vocabulary before anything is stored.



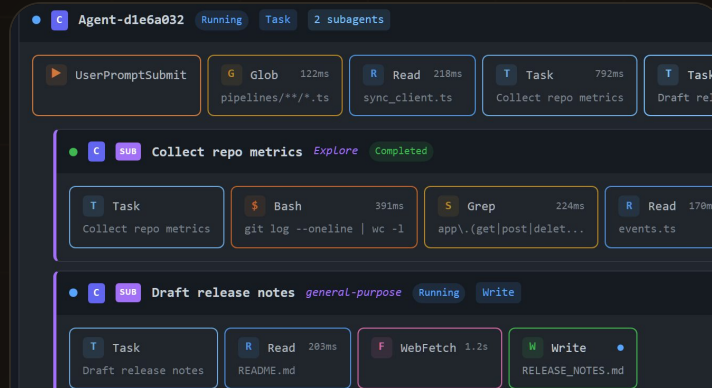
HIGHLIGHTS

How it works



■ NOTHING WRAPS THE AGENT

The installer adds a command hook to the CLI's own settings file, and the script reads the payload the CLI hands it. It never sits between the agent and its model, and uninstalling strips the hook back out of that file.



■ SUBAGENT TREE REBUILT FROM TOOL CALLS

There is no subagent start event to listen for. The hook spots a Task call, synthesizes a start event carrying a parent id, and the dashboard renders that child as a nested lane.

■ A DEAD SERVER CANNOT STALL A RUN

Every event is posted fire and forget with a five second timeout, and hook errors are swallowed. Start the dashboard halfway through a run and it picks up from the next event.

Results & impact

no wrapper on the agent

reads hooks the CLI already fires

runs with the server off

hook errors are swallowed, the agent continues

two CLIs, one timeline

event and tool names normalized on the way in

• STACK

Tech stack

TypeScript

server and dashboard

React

live swimlane UI

Node.js

hook scripts and installer

Express

event intake API

WebSocket

push to the browser

Vite

dev server and build



Sytze P.

Solution Engineer

A hook script, a local event hub, and a dashboard that nests runs by repository, session and subagent.

