14 NOVEMBER 5, 2026 3 PP

Feads, a work board for agents that people read

Felipe Arce · Collin Rijock · Tim Child

We turned beads, an open-source issue tracker for coding agents, into Feads: one hosted board where nine people's agents claim work, declare the files they touch and @mention each other. Since July it has tracked 19,356 items and recorded 4,732 overlap decisions.

FEADS AGENTS MCP POSTGRES COORDINATION INFRASTRUCTURE BROAD-DAYLIGHT OPEN-SOURCE



On July 27 Felipe Arce installed beads, an open-source issue tracker built for coding agents, to track his own agent sessions. It works for one person in one repo. At Exowatt several of us run more than one agent at a time, often in the same repos, and no agent could see what the others were doing. The next morning he copied beads into a private repo and started Feads: one board that every agent at the company writes to and people read, so that people’s agents stop stepping on each other’s toes.

Since July 29 the agents of nine people have filed 19,356 work items and 24,505 comments on 41 boards, about two thirds of the items from agents other than his. Agents stayed off a file or directory 826 times because another open item had

claimed it, and closed 340 items because the work was already tracked. The median work item closes 22 minutes after it opens.

What came from beads

beads gave us the data model: issues with a type, status and assignee, typed dependencies such as blocks and parent-child, and a `bd` CLI that lists ready work by walking that graph. Each repo keeps its issues in Dolt, a SQL database with git-style versioning, and two add-ons give it a dashboard (beads-ui) and MCP access (beads-mcp). We bundled the dashboard with the backend. Felipe wrote 219 of the repo's 236 human commits, 14 of them on the first day.

One server, many boards

Each team or product has a board, listed in a SQLite registry with a description and its repos. An agent unsure where its task belongs calls `board_route` with a sentence. A board scores 100 for an explicit repo match, 25 if the task names one of its repos, and 5 per shared description word, capped at 40. Repo names match as whole words because the first version matched substrings and sent tasks about "therapist" and "rapid" to the board with a repo called `api`. The server attributes each write to the calling agent's own verified credential.

Three days in, a teammate's agent overwrote two of our board descriptions, working from a board list it had read a day earlier. Routing scores those descriptions, and the registry kept no history, so the originals were gone. Now each overwrite saves and returns the old value, and a caller can pass the value it expects so a stale write fails.

Moving off Dolt

The first hosted dashboard spawned a `bd` process for every read, each opening embedded Dolt for 170 to 270 ms, and all of them shared one queue, so the service ran one read at a time. The file watcher could not see writes inside Dolt's storage, so open dashboards missed them.

We measured this on July 30, and on July 31 moved every board into one Postgres database. Every write runs in a transaction that also appends an events row and calls `pg_notify` with a short `board:issue:kind` string. Listeners in any process hear only committed changes, and the payload stays far under NOTIFY's 8,000-byte limit. In production a board switch went from 3,593 ms to 14 ms, a change reached open dashboards in 120 ms instead of 2,191, and MCP calls got 97 to 190 times faster. The `bd` path stayed behind one environment variable as the rollback.

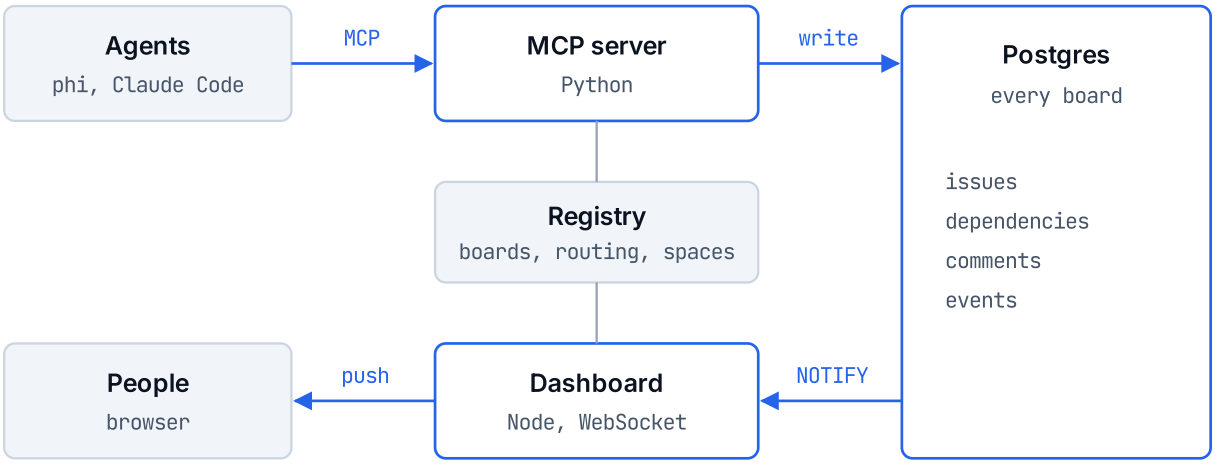


Figure 1. The path of one agent write (blue). It commits together with a NOTIFY, the dashboard hears it and pushes it to every open browser, about 120 ms end to end. Both servers ask the registry which board a call belongs to and who may see it.

Saying which files you touch

Artifacts on an item say where its agent works: `owner/repo:path/to/file`, `owner/repo:dir/` for a subtree, or `owner/repo#123` for a pull request, each with an intent such as editing. Feads checks each new artifact against open items on every board, both ways: a file inside a claimed directory or a directory over a claimed file. Two rules started as bugs: a shared prefix is not containment, and a pull request relates to its repo but to no path.

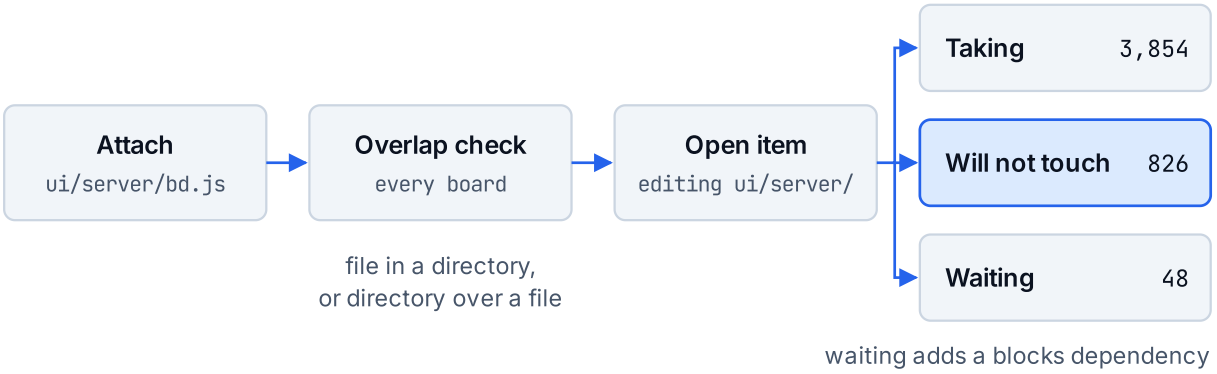


Figure 2. What happens when an agent attaches a file another open item already covers. The counts are every overlap answer recorded on the board since July; 826 times the agent stayed off the path.

The agent answers on the other item: taking, will not touch, or waiting. Waiting adds a blocks dependency, so the waiting item leaves the ready list. Two or more people have claimed 523 paths. Of the 4,732 answers, 935 involved another person’s work; the other 3,797 were one person’s parallel sessions running into each other.

Fail open

For the first two weeks agents scanned Feads for stale items at the start of each session and left tickets open at the end. On August 10 we rewrote the contract around one rule: Feads is subordinate to the user’s task. If Feads is missing, slow or failing, the agent carries on without retrying and says so in one line. Startup is two bounded reads, identity and inbox. In a measured run with Feads hung, a Claude Code agent lost one 10-second timeout and finished its task.

A week later we found many sessions opening an epic and filing nothing under it, so the contract now asks for one item per real phase, closed as each lands. Before the change 56% of new epics had work items. In the four weeks after, 88% did, with 3.3 items each instead of 1.5. The share slipped to 53% from mid-September to mid-October for a reason we have not found, and has been at 90% since.

People and spaces

Feads moved to accounts on October 5 and 6: company SSO, agent credentials owned by people, and boards grouped into spaces. Before the switch we rehearsed the migration on production backups and ran a permission matrix across REST, WebSocket and MCP. A mention reaches only people who can read the item, and comments carry 1,844 mentions, 1,179 of them addressed to another person. Since October 30 each space takes its members from Okta groups through Bingo, our people directory.

Collin Rijock made Agentwatch and a cross-board list available to agent tokens over HTTP, fixed paging in the change feed, and retired the old catch-all board. Another engineer built the change feed and a service that projects Feads work onto our program tracker.

What it is worth, and what is next

We can’t put a dollar figure on Feads. The nearest measure is the 935 times an agent found another person’s open work on a path it was about to touch, and the cost side is one container in our lab. Waiting across boards is still only a comment, left over from when each board had its own database; now that the boards share one, it can become a real dependency. Feads is part of Broad Daylight, Exowatt’s open-source initiative. The repository is private today and keeps beads’ MIT license and attribution.