TECHNICAL CASE STUDY · NOVEMBER 2026

Hive: a terminal multiplexer for AI coding agents

How a Rust daemon/client split, a strict invariant set and an evidence-first knowledge vault took a solo project from spec to a cross-platform v0.5.9 in about two weeks.

RUST · TOKIO · PTY / VT100 · RATATUI · LINUX · MACOS · WINDOWS · MIT

8	~24k	212	28
workspace crates	lines of Rust	tests (real PTYs)	architecture decisions

github.com/JRH89/hive · hive.codefrontlabs.com · Codefront Labs

01 · THE PROBLEM

Agents are long-running, interactive and fragile

Developers now run several interactive agent CLIs (Claude Code, Codex, Gemini, OpenCode) at once, often one per git branch. These tools are full-screen TUIs, they ask questions mid-task, and they die if the terminal that launched them closes. tmux keeps them alive but knows nothing about agents: which one is waiting for input, which repository it belongs to, or which worktree it owns.

> **Goal.** One persistent place to launch, watch and switch between agents, grouped by repository, with each agent's status (working / waiting / done) visible at a glance, and nothing lost when the UI closes.

Constraints set up front (AGENTS.md spec)

- A daemon owns every process. Closing or crashing the UI must never kill an agent.
- Agents are launched from argv, never through a shell, and defined as TOML data.
- Status must never be guessed: it comes from screen manifests or the agent's own report, otherwise `Unknown`.
- Works without Nerd Fonts, in 16/256-colour terminals and under `NO_COLOR`.
- Out of scope: orchestration, plugins, remote daemons (§18).



The TUI: spaces and agents in the sidebar, the active agent mirrored on the right. Captured from a real isolated daemon in tmux.

02 · ARCHITECTURE

The daemon is the source of truth

Two binaries: `hived` owns PTYs, child processes and the authoritative `vt100` emulator; `hive` is a CLI and a TUI that only mirrors and forwards (ADR-0001). This single rule shapes everything else.

```
hive (TUI / CLI)  ──unix socket / named pipe──▶  hived
  vt100 mirror                                   ├─ Session 1: PTY + child + vt100 (authoritative)
  ratatui render                                 ├─ Session 2: …
  key → bytes                                    ├─ workspaces (git repos, worktrees)
                                                 └─ clients registry, event broadcast
```

CRATE	RESPONSIBILITY
`hive-protocol`	Serde enums framed as u32-LE length + postcard, 8 MiB cap. `Hello` pinned as variant 0 by a test.
`hive-core`	Paths (0700 runtime dir), TOML config, agent launch, client handshake and daemon auto-start.
`hive-daemon`	Three threads per session (reader → emulator → broadcast, writer, waiter); per-connection dispatch; DSR/DA/OSC query responder.
`hive-agents`	Agent catalog, argv-based process identification, herdr-compatible screen manifests, status tracker.
`hive-workspace`	Git discovery by reading `.git` / `HEAD` directly (no subprocess); worktree add/remove.
`hive-notes`	Obsidian-compatible markdown vault: links, backlinks, BM25 search, append-only writes.
`hive-tui`	Pure state machine (`app.rs`), renderer, key encoder, split-pane layout, keymap, theme.
`hive-cli`	The `hive` binary: clap subcommands, `hive context` for agents.

Output path

A reader thread does a blocking 16 KiB `read()` on the PTY master, feeds the bytes into the emulator under the session's `term` lock, and publishes a chunk on a bounded `broadcast` (256). Per-connection tasks forward chunks to clients. The TUI's state machine is pure (events in, `Vec<ClientMessage>` out), so most UI behaviour is unit-tested without a terminal.

03 · HARD PROBLEMS

Correctness under concurrency

The interesting engineering is in a handful of invariants that span files. Each was learned from a bug or a near miss and is written down in CLAUDE.md and an ADR.

INVARIANT	WHY IT EXISTS
Snapshot + subscribe under one lock	Attach takes `state_formatted()` and subscribes to the broadcast under the same `term` lock the reader holds while publishing. Otherwise output between the two is lost or duplicated.
Slow clients resync	`broadcast::try_recv` silently repositions after `Lagged`. A lagging client gets a fresh snapshot (`Resync`) instead; the PTY is never blocked (ADR-0004).
Events under the meta lock	Session broadcasts go only through `Session::publish()/remove()`, ordered under the lock, so nothing resurrects a removed session. Fixed a "ghost sidebar entry" bug (ADR-0008).
Reader never blocks on writer	The daemon answers the child's terminal queries itself; replies use `try_send` so a full input queue can't stall output.
Framing not cancel-safe	`read_frame` lives in its own task, never inside `select!`.
Lock order	`term` before `meta`; no std mutex or `watch::Ref` held across `.await`.

Control model

One connection can attach several sessions (split panes, tabs). Control is per session: only the controller's `Input` / `Resize` are honoured; other clients observe until control is free or forced with Leader a (ADR-0010). Hidden tabs detach rather than mirror in the background (ADR-0018).

Status detection without guessing

The daemon identifies agents by the process argv inside each PTY, then applies herdr-compatible screen manifests (Apache-2.0) to the emulator screen to derive working / waiting / done. Agents can also self-report with `hive report-agent`. Anything else is shown as `Unknown` (ADR-0016).

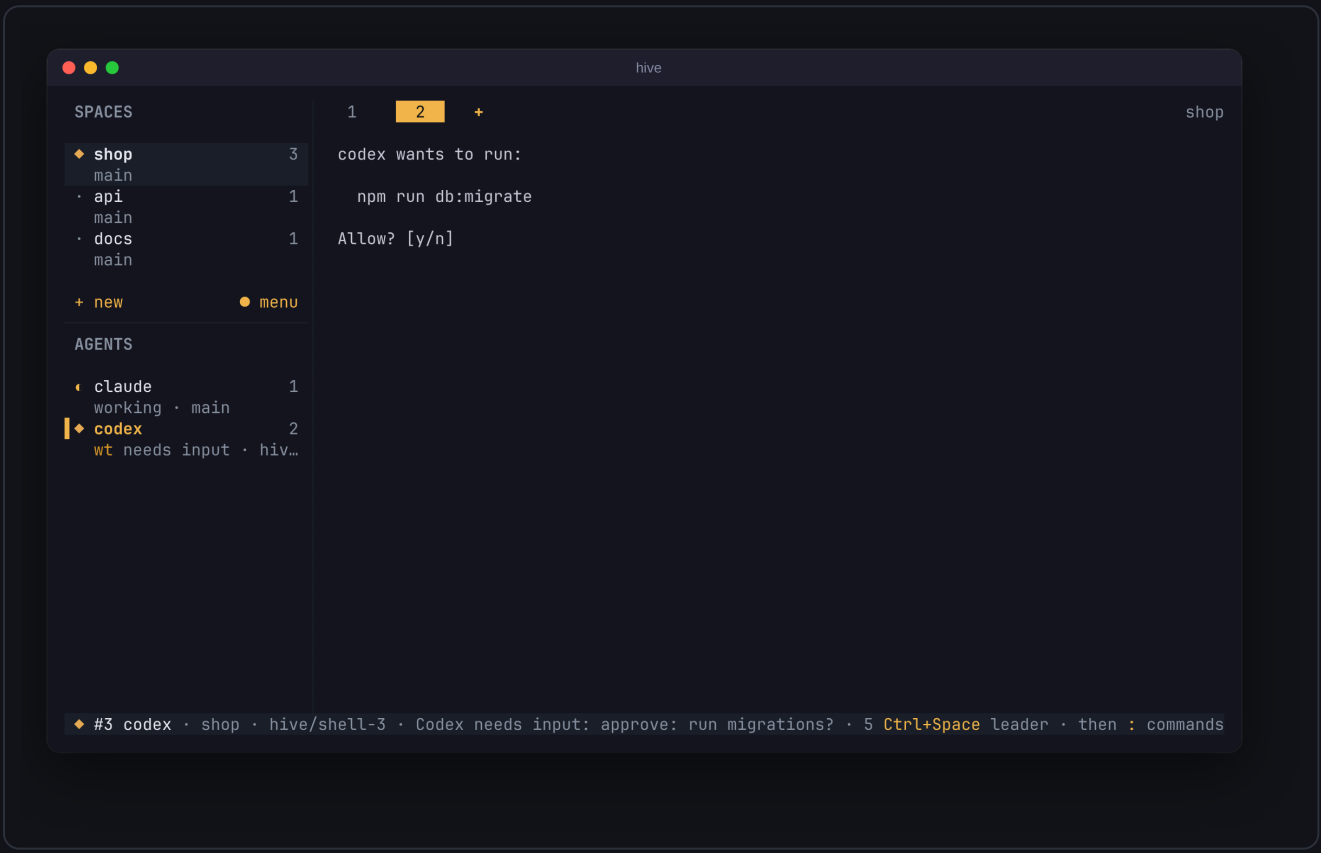
04 · WORKSPACES & PERSISTENCE

Repositories as the unit of work

Every session gets a daemon-owned workspace resolved from its cwd's repository main root (ADR-0011); folders that aren't repos get their own space. Terminals follow `cd` into the right space.

The daemon runs `git worktree add/remove` so each agent can work on its own branch. Dirty or in-use worktrees are never removed without force, and branches are never deleted (ADR-0012).

`sessions.json` stores metadata only, never environment. Sessions alive when the daemon stops come back as `Lost` with the agent that was running; `Ctrl+Space R` restores them and Claude Code / Codex resume their conversation (ADR-0009, ADR-0025). Ids never repeat.



Cross-platform

PLATFORM	TRANSPORT	NOTABLE FIX
Linux	Unix socket, 0700 dir	Static musl binaries, x86_64 + aarch64 (ADR-0015)
macOS	Unix socket	CI on macos-14; arm64 + x86_64 builds
Windows	Named pipe + token auth	`hived` inherited the caller's stdout so `hive | Out-String` hung; std handles now non-inheritable (ADR-0021)

05 · PROCESS

An evidence-first working memory

The repository carries an Obsidian vault (`vault/`) that acts as the project's memory across human and AI-agent sessions: a Status page, a roadmap with acceptance criteria per phase, 28 ADRs and a dated log for every work session. The rule is **record evidence, not claims**: each phase lists how every criterion was checked (test names, tmux captures, CI run ids).

Bugs the process caught

SYMPTOM	CAUSE AND FIX
TERM-trap test passed	SIGTERM arrived before `trap` ran, so SIGKILL was never exercised; a 0.07 s suite gave it away. Now waits for an `armed` marker and asserts elapsed ≥ timeout.
14 stray `hived`s	A CLI test ended with a broken config and `hive daemon stop` loaded config first. Daemon commands now run before config load.
Flood test didn't lag	Per-client queue allowed 16 MiB; test now asserts `Resync` actually happens.
Reused branch failed	Hand-deleted worktree left a "prunable" registration; `add_worktree` prunes first.
`shell` "not defined"	`check_config` read user overrides only; now uses built-ins + overrides.

Testing approach

- Integration tests drive real PTYs with `/bin/sh -c` scripts and wait on output markers, never on sleeps.
- Replies and broadcasts are unordered, so tests fence on the events channel rather than on a reply.
- Gate on every change: `cargo fmt`, `clippy -D warnings`, `cargo test --workspace`, plus a smoke script against an isolated daemon.

06 · OUTCOME

Shipped, and what's next

MILESTONE	DATE	WHAT LANDED
v0.1.0	2026-10-30	Phases 1–6: daemon, TUI, workspaces, worktrees, distribution; static Linux builds
v0.2.0	2026-10-30	Agent detection and status, mouse, split panes, tabs
v0.4.0	2026-10-31	macOS and Windows binaries, `install.ps1`
v0.4.16	2026-11-01	Sessions survive daemon restarts; agents resume
v0.5.0	2026-11-02	`hive notes` and `hive context` for agents
v0.5.9	2026-11-12	Clickable, draggable sidebar scrollbars

Design system

One palette, degraded by terminal capability (ADR-0013): truecolor → 256 → 16 → mono with reverse video, and Unicode → ASCII glyphs. Amber is reserved for accents.



bg #111318 · surface #1c2028 · accent #f2b84b · accent-dim #d99a2b · text #e8eaf0 · muted #8992a3 · success #79c99e · error #e57373

Known gaps

- No scrollback viewing in the TUI (the daemon has it; the client mirrors only the visible screen).
- Shift+Enter is indistinguishable without the kitty keyboard protocol.
- Processes don't survive a daemon crash by design; only metadata does.

Takeaway. Putting the one non-negotiable rule (the daemon owns the terminal) first, writing every cross-file invariant down, and refusing to call anything done without recorded evidence let a small project move very fast without losing correctness.