12 NOVEMBER 5, 2026 3 PP

Running a company on a forked coding agent

Felipe Arce · Tim Child · Collin Rijock

phi is our fork of the open-source pi coding agent. Three infrastructure decisions let a whole team run it: ship one binary promoted by tag, supervise subagents as real processes, and put one hub on every machine so a phone or another session can drive any session.

PHI PI CODING AGENTS RELEASE ENGINEERING CGROUPS TAILSCALE BROAD-DAYLIGHT OPEN-SOURCE



Exowatt's LLM gateway carried about one billion tokens in July and 62 billion in September, and phi, the coding agent we forked on July 31, sent two thirds of September's requests. Its parent is pi, which Mario Zechner started in 2025 and now builds at Earendil with Armin Ronacher and many contributors. pi gave us the hard parts under an MIT license: one API across model providers, the agent loop, a terminal UI and an extension system. phi is part of Broad Daylight, Exowatt's open-source initiative; its repository is not public yet.

In August most of our engineers used Claude Code or Cursor, and we could fix neither. phi gave us one place to fix bugs, one model catalog and one set of shared skills. By mid-October ten engineers had sent 1.24 million requests through phi. Tim Child built the gateway phi talks to and co-owns the fork with Felipe Arce, who wrote

205 of its 351 commits and tagged 55 of the 84 builds the team installed between August 17 and November 5.

Ship a binary, promote it by tag

On August 3 an agent worked around a noisy screen in phi with a local extension and called fixing the fork itself a detour. I set the rule that still holds: fix it in the fork, push to main, and cut a build. Landing on main ships nothing; a change counts once a coworker's install command gets it.

phi compiles with Bun into one binary per platform. Every push to main builds a candidate and archives it under an immutable path named `<version>+<sha12>.r<run>.<attempt>`, one per build attempt. The install script and the in-app `/upgrade` read a flat directory that holds the last-known-good (LKG) build, and only one workflow can change it. That workflow takes a version and a full commit SHA, checks that the annotated tag `lkg/phi/v<version>` points at that exact commit on main, and copies the archived files into place without rebuilding. A build takes about three minutes and a promotion about 35 seconds.

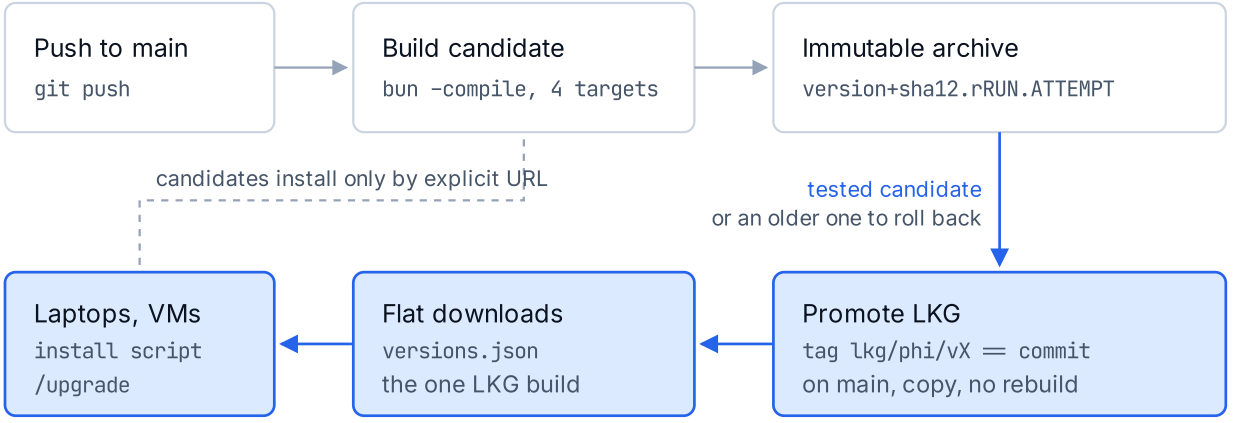


Figure 1. Every push produces an archived candidate; the team's install command only ever sees the one build a tagged promotion copied into place. Rolling back is the same promotion with an older archive.

The first updater checked only that the feed's release id differed from the installed one, and the flat directory is mutable, so `/upgrade` could install an older build. On August 17 we made versions totally ordered, `1.4.0 < 1.4.0-exo.1 < 1.4.0-`

`exo.2`, then workflow run, then attempt. The updater now fails closed on contradictory metadata and pins the download to the vetted archive, so the feed cannot change underneath it.

On October 31 long Opus turns started failing at exactly 300 seconds. Bun's fetch silently aborts a request whose socket has been quiet that long, ahead of phi's own 420-second stream guard, which allows for models that think for minutes without sending a byte. Release 1.4.0-exo.5 hands idle detection back to phi's guard.

Treat subagents as processes

On August 3 we also could not tell whether any subagents were running, because our swarm skills started them as invisible background processes. We made subagents a built-in tool: single, parallel or chained tasks, each with its own model, in the background by default, with a live board and a viewer to watch or kill any child. One laptop alone has logged 6,728 subagent calls since August.

On September 22 a child ran a bare `npm test` on a cloud dev box, and Jest forked 47 workers at about 770 MB each: roughly 22 GB on a 30 GB machine with no swap. The out-of-memory killer took the terminal pane with phi and ten subagents in it. The second kill that day reached `systemd -user` and took every tmux session on the box.

Now each child starts through `systemd-run -user -scope` in its own cgroup, with a memory ceiling of 30% of RAM, clamped between 2 and 16 GiB, and a CPU quota of four cores or a sixth of the machine, whichever is larger. Node reads the cgroup's CPU limit, so test runners size their pools to the quota. A child that hits its ceiling dies alone, and the parent tells the model which flag would have prevented it. A watchdog kills silent children and spares busy ones.

One hub per machine

By late October phi ran on laptops, cloud dev boxes and a Mac mini, and the phone terminal we used to reach them kept dropping. On October 29 we started on an iPhone app like Claude Code's remote control. Its core is one small hub per user per

machine, started at login by launchd or systemd. Each phi session registers with it over a unix socket and never opens a network port. The hub listens only on loopback and the machine's Tailscale addresses, pairs a phone with a single-use QR code, speaks a versioned WebSocket protocol to it, and keeps only a SHA-256 hash of each device token. Agents wrote the 12,596-line pull request against a protocol we specified, and it merged about two hours after the request.

Bun broke this too: in a compiled binary, its `node:http` accepts a WebSocket upgrade and then never sends a byte, so the hub carries its own small HTTP and WebSocket layer on `node:net`. Later releases added session history, resume, and an index over the 2,404 sessions (2.2 GB) on one laptop that answers in under a millisecond once warm, and dropped the cap on headless sessions per hub: each costs about 100 MB, and memory is the only limit.

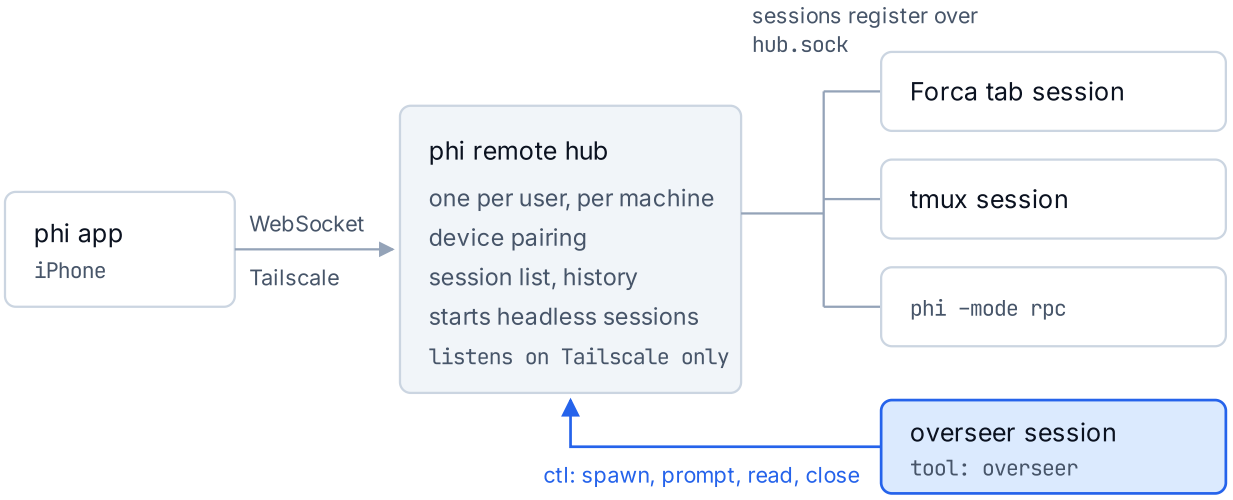


Figure 2. One machine. Sessions never listen on the network; they dial a local hub. The phone and an overseer session both act on sessions through that hub, so a session can start and manage other sessions with no new plumbing.

At 21:58 UTC on November 4 we put agents to work on an overseer built on the hub: a phi session that starts other full sessions with a pinned model, thinking level and subagent policy, in a Forca tab, tmux or headless, then monitors and steers them through control frames on the hub socket. It reached the team's install command 96 minutes later, and the next release passed an end-to-end test on three of our hubs. The next morning one overseer ran eight long writing sessions in parallel.

What comes next

Tim’s MCP and gateway-error work runs in every phi session, and on October 29 Collin Rijock merged ten upstream pi releases, 0.84 through 1.4.0, in one day. With no download counter on the feed, we count users by gateway keys and miss anyone who calls Anthropic or Bedrock directly. The overseer board is on the hub but not yet in the phone app. Several fork changes, such as the stream idle guard, batched tool registration and image fitting, belong in pi, and sending them upstream is the cheapest way to keep the next merge small.