13 NOVEMBER 5, 2026 3 PP

Shipping our fork of Orca, and merging 9,043 upstream commits back into it

Felipe Arce · Tim Child · Collin Rijock

Forca is Exowatt's fork of Orca, the open-source agent workspace. We built the release feed that ships it and phi on every push, a contract that keeps us pairable with the stock phone app, and the method that merged 9,043 upstream commits in one afternoon.

FORCA ORCA INFRASTRUCTURE RELEASE-ENGINEERING FORKS BROAD-DAYLIGHT OPEN-SOURCE



The phone that stopped pairing

On October 23 the stock Orca Mobile app on our phones stopped talking to our laptops. The new app is a web shell that loads a page from the desktop, and it refuses desktops that do not advertise the capability `mobileWeb.bundle.v1`. Ours, a fork of Orca from July and 9,043 upstream commits behind, did not. At 6:41 that evening a build carrying all 9,043 commits was on our internal feed.

That fork is Forca. Orca is Stably AI's MIT-licensed workspace for running coding agents in terminals, with a phone app that pairs to it. Hundreds of people have contributed to Orca, and almost all of Forca is their work: of the 16,174 commits on our main branch, 237 are ours. Felipe Arce wrote 192 of those, most of them the

release and sync infrastructure. Collin Rijock built the command palette, theme packs and shareable settings presets, and other teammates hardened CI and patched dependencies. Forca is part of Broad Daylight, Exowatt's open-source initiative. The fork is internal for now; Orca's code is public at `stablyai/orca`.

Our own work on the fork is mostly infrastructure. Every push to main becomes a signed macOS build on our feed in about 12 minutes, 62 times since August 1. The same feed ships phi, the coding agent ten of our engineers use, and two other desktop apps. In October our merge procedure took the upstream sync from first command to merge commit in 62 minutes, and that speed is the difference between taking upstream's work and turning into a second product nobody maintains.

Every push to main is a release

In July we set one distribution rule: no build is special. A push to main runs `release.yml`, which builds and signs both Mac architectures and uploads them as a candidate. Each candidate gets an immutable id, like `1.4.214-exo.4-dev-a7995180be0b-r40138145797.1`, and a rerun gets a new one, so no upload overwrites bytes someone may have installed. The feed itself is a small FastAPI service of our own.

The goal was that everyone downloads the bytes a teammate has tested, so promotion to last-known-good (LKG) is a separate workflow that never rebuilds. It requires an annotated tag such as `lkg/forca/v1.4.214-exo.4` that resolves to a commit on main, then copies the archived bytes to the download aliases. A rollback promotes an older archive.

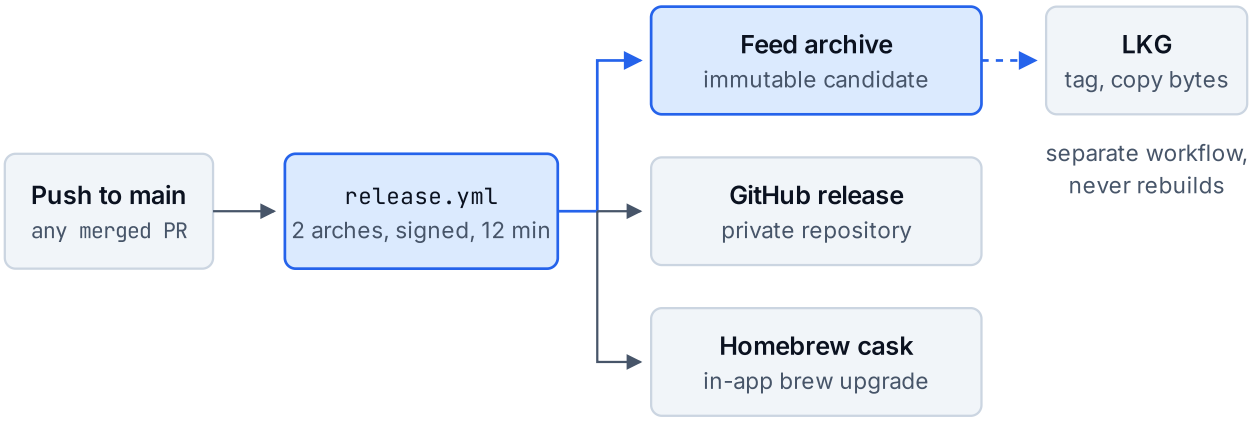


Figure 1. The release path. Every merge produces a signed build in three places; promotion to last-known-good is a separate, tag-gated step that copies archived bytes.

Signing caused the most trouble. Each ad-hoc signed build gets a new code hash, and macOS ties permission grants to that hash, so every update asked for every permission again. We made one internal code-signing certificate and pointed the app's designated requirement at it, so grants survive upgrades. We lack the Apple Developer ID signature Electron's built-in updater needs, so the release workflow bumps a private Homebrew tap on every push and the app runs the upgrade for you.

Two names and one contract

The first fork commit, on July 20, pinned what must never change. Forca shows its own name but still calls `app.setName('Orca')` and keeps the `orca` data directory and `~/.orca` folders. macOS names the Keychain item holding the app's encrypted secrets after the app name, so a rename would orphan every stored credential.

We do not ship a phone app; people pair with Stably's stock one, so `forca-mobile-compat-contract.test.ts` pins the pairing deep link, the pairing-offer version, the end-to-end encryption's HKDF labels, the runtime protocol version and the default WebSocket port. A change that would break pairing fails CI.

On October 6, with about 15 phi sessions open, Forca's main process sat at 50 to 60% CPU, and the phone lagged and froze over Tailscale. A regex trimming trailing blank rows in quadratic time took about 96% of main-process CPU, and a linear

backwards scan cut that work from 1,228 to 12 CPU-milliseconds per second. The desktop also dropped the phone after one missed pong and discarded terminal output past an 8 MiB queue. Now it waits for three missed probes and resyncs a congested phone with a fresh snapshot, using only messages the stock app understands. The fixes merged four hours after the first report of the lag.

Merging 9,043 commits in an afternoon

In August a trial merge measured the gap: 1,780 upstream commits and 221 conflicts. We wrote the method down as a phi skill, fork-sync. It works in an isolated full clone, never rebases or squashes, and lands a two-parent merge with Forca as the first parent, so every Forca commit stays reachable. Files only Forca touches stay ours without review. For shared files, a manifest lists protected paths and pairs of files that must change together, like the desktop and phone copies of the protocol version. Agents may only propose patches against a checksummed packet of conflicts; a small engine applies them, verifies them and builds the merge commit.

By October the gap was 9,043 commits and 663 conflicted paths: 470 edited on both sides, and 139 that Forca had changed and upstream had deleted, many because upstream had replaced its SSH relay with a new remote server, orcad. The rule we set: what only we changed stays ours, what both changed gets the better feature, and upstream’s restructuring wins on structure. We cut the conflicts into nine disjoint file lists by area and gave each to an Opus agent with a written contract covering storage identity, the phone contract and the relay decision.

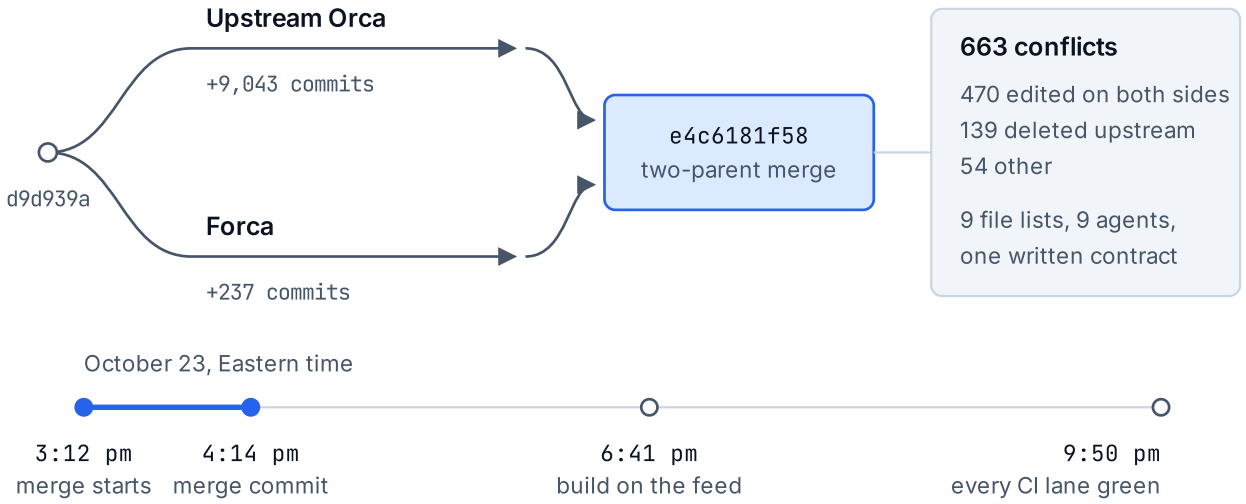


Figure 2. The October 23 sync. Nine agents resolved the 663 conflicts in parallel inside one merge; the blue span is the hour it took.

The 6:41 build advertised the capability the phone wanted, and by 9:50 every CI lane was green, including a new required check that renders the page the stock phone app loads.

What broke

We applied the orcad decision too broadly. Forca spots a devbox shell tab by an escape-sequence marker, OSC 9998, that the devbox CLI prints when it attaches. That code sat next to the relay code, and the sync deleted both. The next day our devbox tabs read “Terminal N”.

The fix shipped about five hours after we noticed. The main process now tracks devbox identity per terminal from the marker, or from the process tree when the marker is gone, so open tabs recover within seconds of the update.

What’s next

Upstream has added 3,251 commits since October 23. Waiting turned 221 conflicts into 663, so I want small, scheduled syncs, with the fork-sync engine doing the bookkeeping we did by hand. Pairing had a contract test and came through the merge untouched; devbox tabs had none. Before the next sync, every Forca feature

that matters needs a contract test like the phone’s, and we need to build orcad’s deployment template so SSH remote hosts work in Forca again.