05 OCTOBER 8, 2026 4 PP

Anyone at Exowatt can ship an app

Collin Rijock · Tim Child · Felipe Arce

How a manifest, one reusable workflow and a Claude Code plugin put 205 internal apps on our own servers in under six months, nearly half of them owned by people outside the software team.

SOFTWARE-TEAM AGENTS APPS-PLATFORM

How a manifest, one reusable workflow and a Claude Code plugin put 205 internal apps on our own servers in under six months, nearly half of them owned by people outside the software team.

The problem

Exowatt builds hardware. The people who need internal tools are optics, controls, power and manufacturing engineers, along with finance, HR and operations. Once AI coding tools got good, many of them started writing their own software: a cost-down tracker for the optical subsystem, a lockout/tagout permit system, a viewer for engine test data.

Writing the tool turned out to be the easy part. A tool only helps once other people can open it, and that takes a URL, company sign-in, a database that keeps state, logs, and a way to fix it when it breaks at 7 pm. Without a platform, each of those meant asking the software team.

We decided the software team would stay out of the deploy path. Anyone with a GitHub repo should get a live, signed-in app at its own internal web address without filing a ticket.

What we built

The platform runs on our own Proxmox server in Austin and is reachable from the company network. The core is five pieces:

Part	What it does
Provisioning API	Creates one container per app, adds its route, and sets up a per-app Postgres database, storage bucket or message queue when the manifest asks for one
Reusable workflows	Shared GitHub Actions for first deploy, deploy on push and static deploy, plus logs, env updates and re-setup
Templates	Fullstack (FastAPI and React), FastAPI, Streamlit and static
Apps catalog	Lists every deployed app with its owner, team and live health
`internal-apps` plugin	Claude Code skills to deploy, troubleshoot, improve and update an app, backed by our platform-ops MCP server

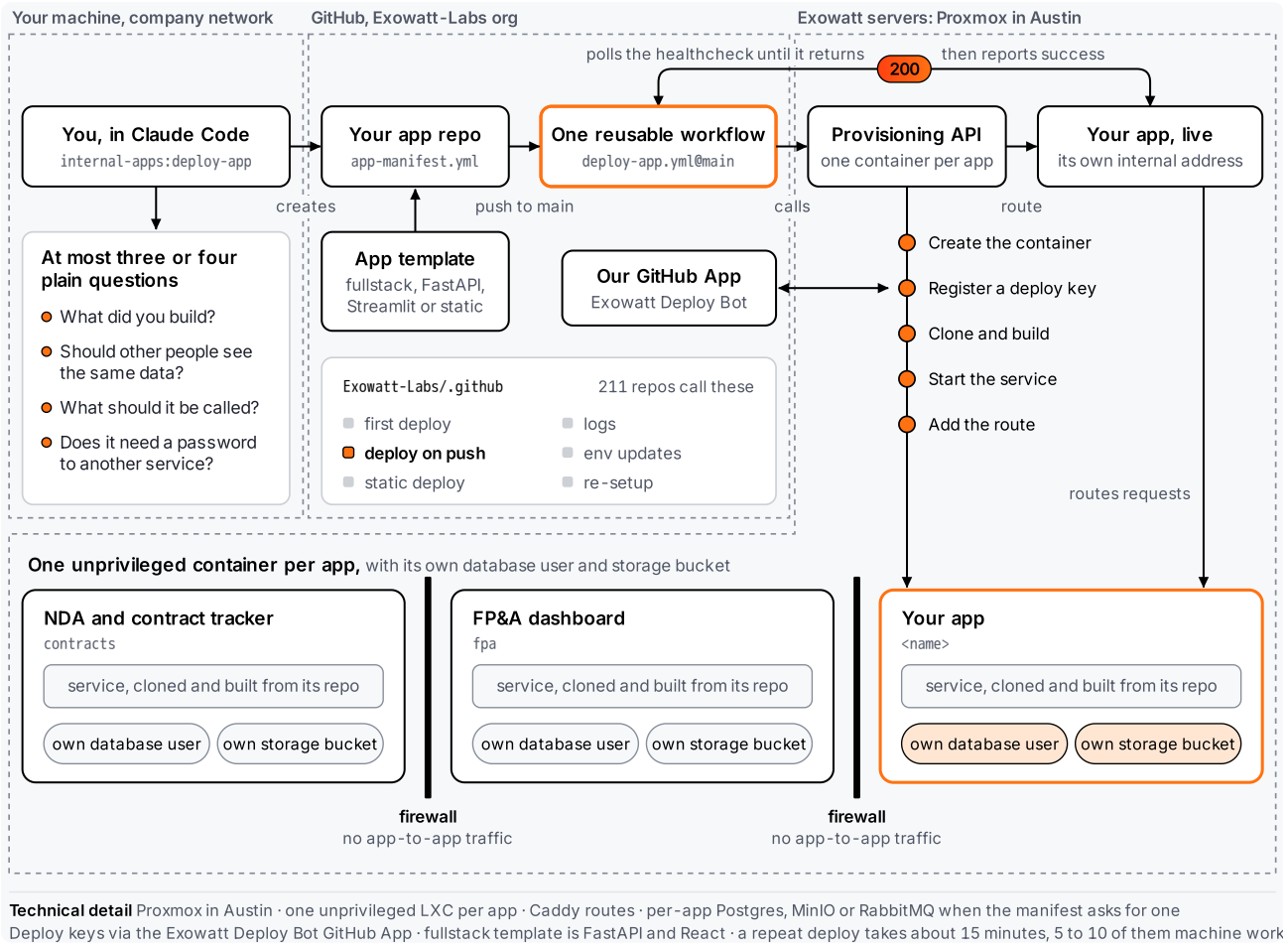


Figure 1. The whole path from a question in Claude Code to a live app. Orange pulses follow one first deploy through the reusable workflow and the provisioning API. The bottom row shows how each app is

kept apart from the others.

A fleet needs tools of its own. Kombucha shows each app's health and logs and gives you `exo-ssh <app>` with a short-lived SSH certificate scoped to that one app. The unused-apps auditor gives every app a staleness score from 0 to 100 and runs a governed teardown that archives the app's database instead of dropping it. A secrets service in front of OpenBao handles agent credentials, and Hypergoose gives an agent one key for many internal and SaaS connectors.

How it works

Every app has an `app-manifest.yml` at the repo root with its name, runtime, entrypoint, port, healthcheck, resource tier, and any shared services or secrets it needs. The workflow that deploys on push is one job:

```
jobs:
deploy:
uses: Exowatt-Labs/.github/.github/workflows/deploy-app.yml@main
secrets: inherit
```

The first deploy creates the container, registers a deploy key through our GitHub App, clones and builds the repo, starts the service and adds the route. After that, every push to `main` redeploys, and the workflow polls the healthcheck for a 200 before it reports success. Every app calls the same reusable workflow, so a fix to the deploy path reaches all of them on their next push.

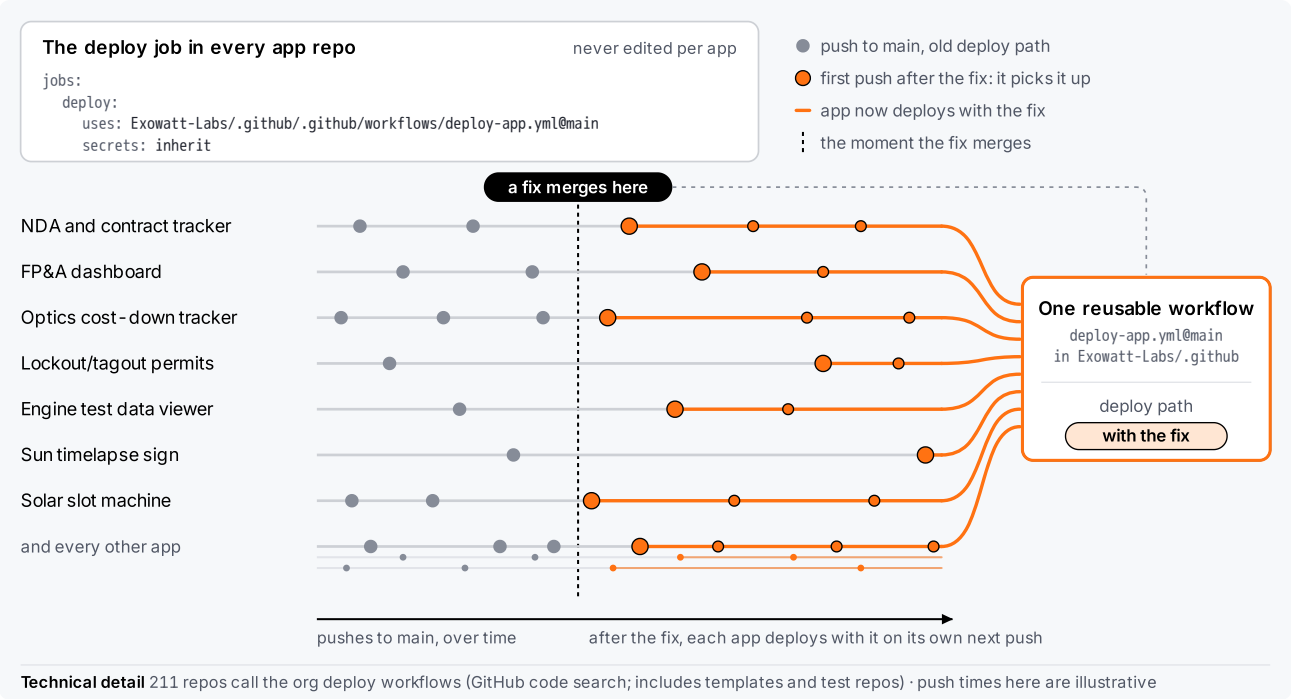


Figure 2. Every app repo carries the same short deploy job, so the deploy path lives in one place. When a fix merges there, each app picks it up on its own next push to main. The push times are illustrative.

Most people never see the YAML. They open Claude Code and run `internal-apps:deploy-app`. The skill checks their network and GitHub access, then asks at most three or four plain questions: what did you build, should other people see the same data, what should it be called, and does it need a password to another service. It picks a template, creates the repo and starts the first deploy in the background while it converts the code. Those changes go through the same automated code review as the rest of our code. Our docs put a repeat deploy at about 15 minutes end to end, with 5 to 10 of those minutes being machine work after the questions.

Each app runs in its own unprivileged container with its own database user and storage bucket, and the network firewall stops apps from reaching each other. Our architecture doc calls that last rule the one that made it safe to open the platform to the whole company, and it is the reason we were willing to.

Results

These numbers come from the live catalog API on October 8, 2026. That morning the catalog said its provisioning API was unreachable and served last-known-good data, so counts can lag by a deploy or two. Owners are counted by Exowatt email, falling

back to GitHub login, with duplicate accounts merged and the 4 unowned apps left out. “Software team” means the 5 members of our GitHub software-team group.

Measure	Value
Live apps	205, all reporting up
Container apps / static sites	169 / 36
Apps with an MCP endpoint	45
Apps with an agent surface	16
Named owners	35 people
Owners outside the software team	30 people, owning 95 apps
Owners on the HR, finance, operations or commercial GitHub teams	6 people, owning 14 apps
Repos that call the org deploy workflows	211 (GitHub code search; includes templates and test repos)

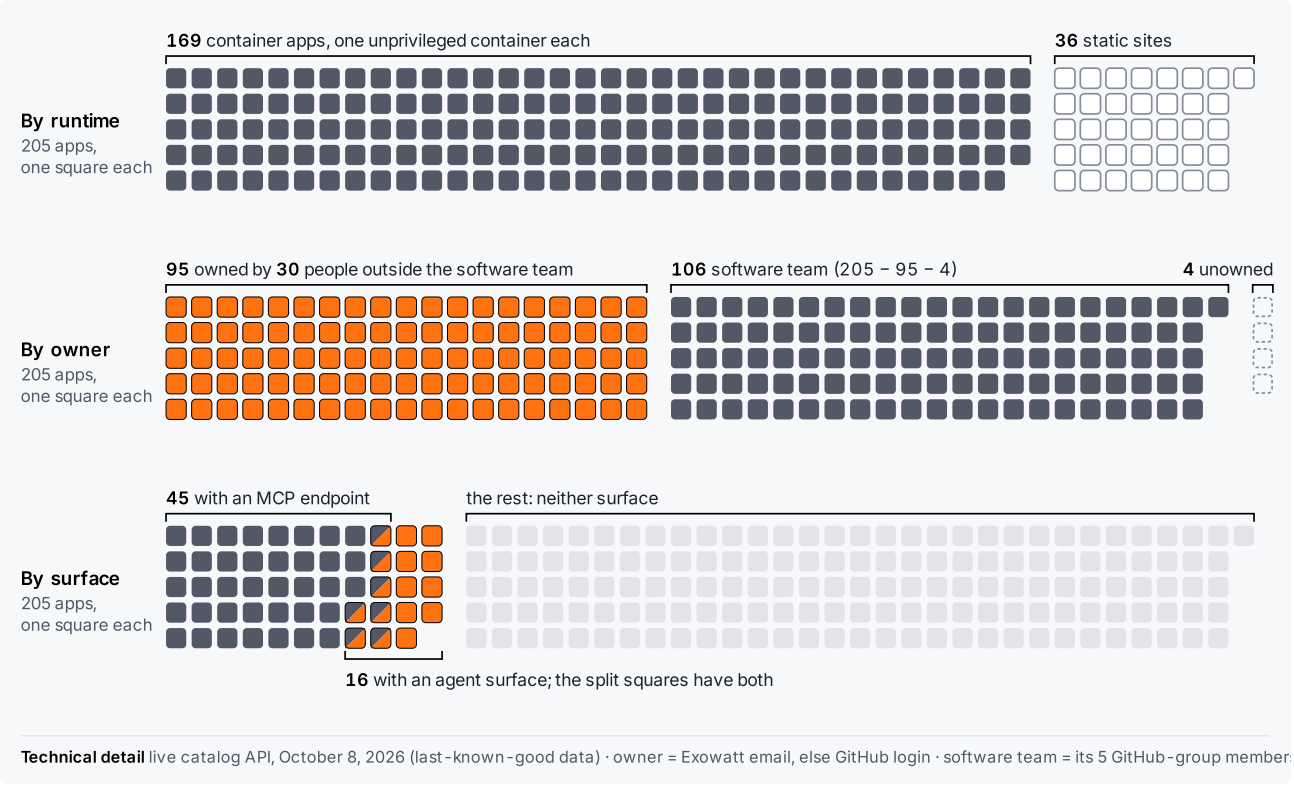


Figure 3. The 205 live apps on October 8, 2026, one square per app, sorted three ways. People outside the five-person software team own 95 of them. The surface row overlaps: some apps have both an MCP endpoint and an agent surface.

The first provisioning API commit landed in mid-April 2026, so the fleet is under six months old. Tim Child wrote that commit, and he authored 310 of the 378 pull requests merged into the platform repo.

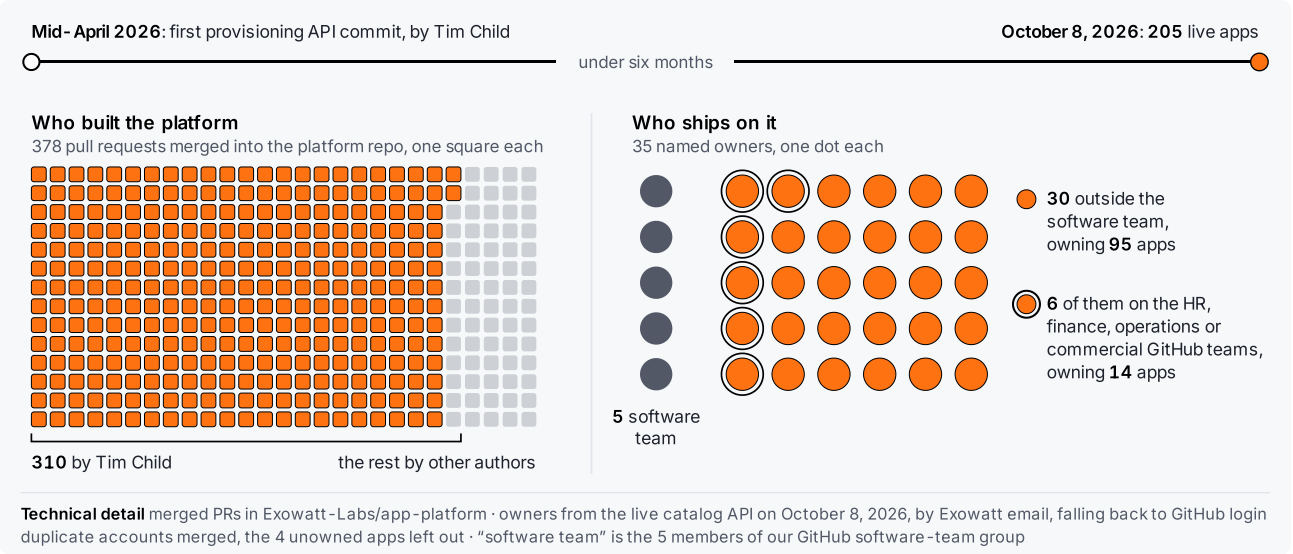


Figure 4. Tim Child authored 310 of the 378 pull requests merged into the platform repo. The people who ship on it mostly work elsewhere in the company: 30 of the 35 named owners sit outside the software team.

The apps are ordinary company work. One tracks NDAs and contracts through legal review and DocuSign. Finance has an FP&A dashboard for actuals, budget and cash flow. Optics tracks levelized-cost reductions for the optical subsystem. A digital sign loops a 24-hour timelapse of the Sun from NOAA's SUVI imager, and Tim built a solar-themed slot machine as a demo of the platform.

What's next

Deploying still needs Claude Code on a machine connected to the company network. The next phase gives the platform-ops MCP server tools to create the repo, make the first commit and trigger the first deploy itself. Those tools are built and stay switched off until we turn on their flags. The GitHub org-seat check already runs from Cowork and Claude Desktop.

The secrets service gives agents scoped, audited reads of their credentials from OpenBao, and its hook into provisioning is also off by default for now. With 205

apps, removing dead ones matters as much as shipping new ones, and the unused-apps auditor already scores every app for exactly that.

About Exowatt — Exowatt is a next-generation renewable energy company providing modular dispatchable energy solutions for energy-intensive applications. Inquiries:
