

How We Work

How the system is meant to work, and where it is still thin

YouChews · written 27 Aug 2026 · status updated 28 Aug · every claim verified against the live accounts

What this describes

We have been building a way of working where **the knowledge lives in files we own rather than in anyone's head or anyone's chat history**. This document lays out how it is meant to work, and then examines it honestly — because the useful version of any system is the one that survives somebody looking for the holes.

Read the second half first if you are short on time. The first half is the design. The second half is where it is still thin.

The model, in one page

Four layers. Each has one job.

Layer	What it is	Who touches it
The Vault	The private repo, <code>bretttob2424/beacon</code>	Both of us, through our own Claude sessions
The Desk	Your own Claude session, pointed at the Vault	Each person separately
The Wire	Messages as files — tasks, journal, decisions	Written by the sessions, not by hand
The Front Door	Cloudflare Pages	Anyone, including people with no Claude account. <i>Currently open to anyone with the link; a login is recommended, not yet applied.</i>

The core claim: a Claude session opened on this repository wakes up already knowing the house rules, the settled facts, what happened yesterday, every open prediction and every project. Nobody briefs it. Nobody pastes context. **You should be able to ask "what do I owe?" or "what did Brett decide about analytics?" and get a real answer with the file behind it.**

Why the repo and not Claude's own memory: Claude's memory only works for one person at a time — Brett's does not reach you and yours will not reach him. The same is true of ChatGPT. There is no team-memory feature to wait for, so the repo is what reaches both of you, and it would survive switching tools entirely.

What is verifiably true today

Not claims — things checked against the live systems while writing this.

- The repo is real and populated:** house rules, settled facts, a running journal, every project ranked, every report with its PDF, and 124 working tools.
- The prediction ledger is not decoration.** As of 28 Aug: **300 predictions written, 36 graded, 17 of 36 correct — a 47% hit rate.** That number is published deliberately. A ledger that only recorded wins would not be worth keeping.

- **Collaborator access works.** An invitation issued 28 Aug carried write permission and matched an existing GitHub account, so onboarding needs no signup step.
- **Publishing works.** Three sites run on Cloudflare Pages today. One is gated, two are open.
- **Every prediction carries a stated falsifier**, written before the outcome is known. This document adds several more.

The stress test

Eight ways this breaks. Each one is a real observation, not a hypothetical.

1. Your work will get stranded on branches — this is already happening

The evidence: there are **four open pull requests** on the repo right now, all created 26 August, all from cloud Claude sessions, **all unmerged two days later**. Device architecture, a four-day handoff document, a cash analysis, a Meta ad expansion brief. Real work, sitting in a queue nobody is emptying.

Why it matters to you: when you work in the browser, that is exactly the shape your output takes — a branch and a pull request. If nobody merges, **your contribution never reaches the shared brain**, and you will reasonably conclude the system does not work.

The mitigation, and it needs an owner: you have write access, so you can merge your own. But somebody has to own the review pass. **Proposal: Brett's Mac sessions work directly on main; your sessions work on branches and open PRs; the PR queue gets cleared at least twice a week.** First test of that: those four existing PRs.

2. Two people, one set of files, 80+ commits a day

The evidence: 579 commits in the last seven days. The settled-facts file is a single document that both of us will want to edit.

The mitigation already built in: the house rules say *prefer appending over rewriting*, and the commit tool takes only named paths so two workers never steal each other's staged files. That was designed for several chats on **one** machine. **It has never been tested with two people on two machines.** Expect friction here first.

3. You can read everything and run almost nothing — day one

This is the most important practical thing in this document, and it is the one that will cost you ten minutes and your confidence if nobody says it first.

You get	You do not get
Every document, report and decision	.env — all four API keys
The full source of all 124 tools	.venv — the Python environment
The house rules and 71 memories	beacon_data/ — the databases, media library, thumbnails
The journal, ledger and project map	—

All three exclusions are deliberate — secrets and multi-gigabyte binaries do not belong in git. **The consequence is that most tools will not execute for you on day one.** That is acceptable, because your job here is to read, decide and follow through rather than to run the ad pipeline. **But it means "it doesn't work" is the expected first experience, and it is**

not a broken system.

4. There is no way to give you the API keys yet

Following from the above: if you ever *do* need to run the data tools, the four keys live only in an ignored file on Brett's Mac. **There is no secure distribution path and we have not designed one.** They must not go by email.

Recommendation: do not solve this yet. Nothing in your first two weeks needs them. When it does, the answer is probably a shared secret manager, and it is a decision to make deliberately rather than in a hurry.

5. The repository is about 5.5 gigabytes

Ad videos were committed into the history. A full clone is a real download on a home connection.

Mitigation: do not clone it. Claude Code in the browser works against the repo with nothing installed. Only clone if you specifically want to run tools locally — and see point 3 for why that is a bigger project than it looks.

6. All automation depends on one laptop that sleeps

The morning report, the data refresh, the dashboard updates — every scheduled job runs on Brett's MacBook. **It sleeps, and the jobs are late or missing.** One scheduled job is currently failing on every single run because of a macOS file-permission restriction on where the repo lives.

Two fixes are queued: moving the repo onto an always-on Mac Mini, and moving the data pull into a Cloudflare Worker on a schedule so it runs in the cloud regardless. **Until one of those lands, treat any automated number as "usually there" rather than "always there."**

7. There is no notification layer yet

Nothing pings anybody. If Brett files something for you at 9pm, you find out when you next look.

The plan is one daily digest email — what changed, what is newly owed, what broke. It is not built. **Until it is, this is a pull system, not a push system.** For anything genuinely urgent, use the phone like a normal person; this architecture is for work product, not emergencies.

8. Nobody has tested any of this end to end

You are the test. The entire model rests on one prediction, written before you had access so it could not be fudged after the fact:

*Prediction #283: with repo access and his existing Claude seat, a second person can answer "where is the Mixpanel analytics document?" from his own account, in one session, with nothing sent to him and no new tooling. **Wrong** if he needs a file forwarded, or the Claude GitHub connector cannot be authorised for his account.*

It is open and ungraded. Please do not be gentle with it.

Your first hour, concretely

1. **Accept the GitHub invitation** in your email.

2. **Open [claude.ai/code](#)** and connect it to `bretttob2424/beacon`. You will be prompted to install the Claude GitHub App on your own account — approve it. Brett cannot do this part for you.
 3. **Ask your session, in plain language:** *"Read STATE.md and tell me what this company is doing and what is currently true."*
 4. **Then run the actual test:** *"Where is the Mixpanel analytics document and what did it conclude?"*
 5. **Then:** *"What is on the Chief of Staff's list?"* — there are nine items, ranked by Brett.
 6. **Write down every point of friction.** Not to be polite about it. The friction list is the most valuable thing you can produce this week, and it is worth more to us than any of the nine items.
-

What we would most like you to attack

Six questions where an outside opinion is worth more than our own:

1. **Is "messages are files" actually workable between two people**, or is it a clever idea that only survives because one person built it?
 2. **Should the repo move to a GitHub Organization now** rather than later? It is owned by Brett's personal account, which means the company's entire written history depends on one individual's login. We deferred it to avoid delaying you. Was that right?
 3. **Is a daily digest email a sufficient heartbeat**, or does an async two-person team need something faster? Brett will not use Slack, so a channel is off the table — which may be a mistake we should revisit.
 4. **How should secrets reach you** when the time comes?
 5. **What breaks when a third person joins?** Everything here is designed for two.
 6. **What did we not think of?** That is the real question.
-

The predictions this document adds

Written now, before the outcome, so they can be scored rather than argued about later.

#294 — The four stranded pull requests are a process gap, not an accident: with a named owner and a twice-weekly pass, the open-PR count stays at or below two for the next four weeks. *Wrong if* PRs older than seven days accumulate again, or if merging them surfaces conflicts that make branch-based contribution impractical.

#295 — the first week with a second person produces a friction list with at least three items we did not anticipate in this document. *Wrong if* everything he hits is already named in the eight failure modes above — which would suggest we are marking our own homework rather than genuinely testing.

Verified 27–28 Aug 2026 against the live GitHub repository, the Cloudflare account, the Claude organisation, and the repo's own commit and prediction history. Every number in this document was read from a system, not recalled.