

The YouChews Operating System

How Brett, a second person and the team work with Claude — the architecture, the game plan, and the five things only Brett can do

Written 27 Aug 2026 · status updated 28 Aug

A living document. Where something has changed since it was written, the change is noted in place rather than the original being quietly overwritten.

The shape of it

Four layers. Each one has a single job, and nobody has to learn a new app.

Layer	What it is	What it holds
The Vault	Our private GitHub repository	Every document, decision, number, rule and memory. The single source of truth.
The Desk	Each person's own Claude session	The only interface anyone touches. Brett dictates. a second person asks. Neither types a command.
The Wire	Files in the Vault, plus one daily email	Messages between people — tasks, notes, what happened — written by the Claude sessions, not by hand.
The Front Door	Cloudflare Pages	Any deliverable becomes a real web address anyone can open, including people with no Claude seat. <i>A login is recommended below and is not yet applied.</i>

The principle underneath: the durable stuff lives in files we own, and everything else is a view onto those files. That is why this survives a change of vendor, plan, model or app — and why nobody can switch it off.

Layer 1 — The Vault

github.com/bretttob2424/beacon, private. It already holds:

- **The house rules** — auto-loaded into every Claude session that opens on it
- **The settled facts** — verified numbers, decisions, what is currently true
- **The running journal** — appended the moment anything happens
- **Every project** and its sub-items, ranked
- **Every prediction** we have made and how it scored
- **Every report**, with its PDF alongside it
- **70 durable memories** about how we work and what has already been proven
- **117 working tools**, 3 skills, 3 specialist agents and 4 automatic hooks

Why this is the centre. A Claude session opened on this repository wakes up already knowing everything. Nobody briefs it. Nobody pastes context. That is the whole trick, and it already works — it just currently works for one person.

Layer 2 — The Desk

Everyone gets the same interface: **their own Claude, pointed at the Vault.**

- **Brett** talks. He dictates a thought and the session does the filing — logging the idea, updating the state, writing the report, generating the PDF, recording the prediction. He has never typed a command and does not start now.
- **a teammate** asks. *"What did Brett decide about analytics?" "What is owed to me?" "Show me the last thirty days of DTC performance."* His session reads the same files and answers from them.
- **Anyone else with a seat** does the same, scoped to what they need.

Two people never have to be in the same conversation to be working from the same brain. **The repository is the shared context; the sessions are just windows onto it.**

Layer 3 — The Wire

This is the part that replaces a chat app, and it is already about eighty percent built.

Messages are files, written by the sessions. When Brett says *"tell a teammate to sort out the Amazon connector,"* his session files it — verbatim, with the reason, ranked against everything else a teammate owes. When a teammate finishes it, his session marks it done and writes what he found. Neither of them opens a task manager. Neither of them types a command. The list is just there, and it is the same list for both of them.

The same holds for everything else: the journal records what happened, the ledger records what we predicted and whether we were right, and the state file records what is currently true.

The one thing missing is a heartbeat — so we add one email. Once a day, an automatic digest goes to a second person and to Brett:

- what changed in the Vault since yesterday
- what is newly owed, and to whom
- any new document, with a link that opens in a browser
- anything that broke

Brett already lives in email. a teammate already lives in email. No new habit, no new app, no new place to check. The machinery for this already exists in the repo — the daily report has been running for weeks.

Layer 4 — The Front Door

Some things have to be *seen* by a human who does not have a Claude account — Carlos, Hector, Sebastian, an accountant, an investor. For that we already have the answer, and it is already live.

Cloudflare Pages, with Cloudflare Access recommended in front of it. *Status as of 27 Aug: the site is live and currently open to anyone holding the link. The Access step below is a recommendation we have not yet applied — stated plainly because a reader can check it in five seconds.* Any report, dashboard or brief becomes a real web address. Access puts a login in front of it: the person types their email, gets a six-digit code, and they are in. **No password to manage, no account to create, no software to install.** Free for up to 50 people.

Three sites are already running on it. This is not a proposal — it is a thing that works today and only needs pointing at the rest of our output.

Why this beats every other sharing route we have tried. A published Claude artifact is private to the person who made it. A Google Drive link needs permissions we cannot set programmatically. An emailed PDF cannot be interactive and goes stale the moment it is sent. **A Cloudflare page is a normal web address that keeps working, keeps updating, and can be locked to exactly the people we choose.**

About Slack — and what we do instead

We are not using Slack. Claude Tag, the version of Claude that lives inside a chat channel, is **Slack-only**. There is no Microsoft Teams version, no Discord version, and no announced date for either. So the honest choice is binary: adopt Slack, or do not have Claude-in-a-channel. **We do not.**

What we lose: an always-on room where Claude sees the conversation and can jump in.

What we use instead: **the Vault is the channel, and email is the notification.** It is asynchronous rather than live, which for a three-person company spread across locations is arguably the better fit anyway — nobody is waiting on anybody, and everything said is written down permanently instead of scrolling away.

One experiment worth thirty minutes, not a commitment. Jack Dorsey's **Buzz** is the Slack alternative built for exactly this: AI agents as full members of channels, with Git in the same place. It is free, open source, and there is now a hosted version at buzz.xyz so it can be tried without running any servers. It is early beta and the hosted onboarding reportedly has rough edges. **Worth a look, not worth betting the company's workflow on this month.** If it matures, it is the natural home for Layer 3.

The five things only Brett can do

Everything else in this plan is mine to build. These five are invitations, identities and admin switches that nobody else can press.

#	What	Why it matters	Time
1	Invite a teammate to the GitHub repository and authorise the Claude GitHub connector for the organisation	This single action turns on Layers 1, 2 and 3 for him at once. It is the only true blocker in the entire plan.	10 min
2	Add a teammate to Cloudflare Access and put the login in front of the docs site	Gives him — and later Carlos, Hector and anyone else — a real web address for anything we publish, with no password to manage	10 min
3	Turn on memory for the YouChews organisation	It is switched off by default on the Team plan, so both of you are currently working without it	1 min
4	Say yes to the daily digest email and confirm which address a teammate should receive it at	This is the heartbeat that makes an asynchronous system feel alive	a reply
5	Decide the Cloudflare account question — see below	It is a five-minute decision with a long tail either way	5 min

Total: about half an hour, plus two decisions.

Cloudflare, specifically — you asked, so here is the straight answer

You do not need to sign up. You already have an account.

- Account name: "Brettobrien24@gmail.com's Account"
- Account ID: a6aa0bad3fff5aed0622376f87d27089
- It is on the **personal Gmail** — brettobrien24@gmail.com — the same address the Google Drive and Gmail connectors use. **Not** brett@youchews.com.

To be precise about access: I am not using it. No session of mine has ever logged in to Cloudflare. A token stored on your Mac was created when the account was set up, and the publishing runs as **you** — a script you double-click in Finder, using your own credentials. Nothing about Cloudflare runs under my identity, because it cannot.

Three sites are live right now:

Address	Status	What it serves
youchews-creative-os.pages.dev	Password-protected — returns a login prompt	The full creative library, 533 ad creatives plus thumbnails
youchews-docs.pages.dev	Open to anyone with the link	Mission Control, the daily scorecard, the current brief
youchews-grok-wave.pages.dev	Open to anyone with the link	A video board of unpublished ad concepts

The decision for you, item 5: youchews-docs.pages.dev currently serves revenue, ad spend, margin and profit to anyone who has the address. That was a deliberate trade to unblock a handoff, and it may still be the right trade — but it should be a choice you make knowingly rather than one that persists by accident. **Putting Cloudflare Access in front of it takes about ten minutes and costs a teammate one email code the first time he opens it.** My recommendation is to gate it.

On the account email: the cleanest long-term answer is that company infrastructure lives on brett@youchews.com. The honest counter-argument is that moving a Cloudflare account means moving three live projects and a stored token for no functional gain today. **My recommendation: leave it where it is, write it down so nobody re-derives it, and use brett@youchews.com for anything new.** Consistency matters less than knowing where things are.

A day in the life, once this is on

Brett, 7am, on his phone. *"How did we do yesterday, and what should I worry about?"* His session reads the overnight pull, gives him the number, flags the one thing that moved. He says *"put that in front of our Chief of Staff."* It is filed before he finishes his coffee.

our Chief of Staff, 9am, his own laptop, his own account. He opens his Claude on the repo. *"What do I owe?"* He gets the ranked list, in Brett's own words, with the reason attached. He works one item. When he is done his session writes what he found — and Brett's next session already knows.

Carlos, 2pm, no Claude account at all. He gets a link. Types his email. Gets a code. Sees the live creative performance board, current as of this morning.

Nobody typed a command. Nobody pasted context. Nobody asked anybody to catch them up.

The second pass — challenging all of the above

Where this plan is weakest

- 1. There is no heartbeat faster than a day, and that is a real cost.** A daily digest is not a conversation. If a new teammate needs an answer at 11am, this architecture gives him nothing — he texts or calls, like before. **I am explicitly trading immediacy for permanence**, and that is the right trade for work product but the wrong one for urgency. Keep the phone for urgent. If we find ourselves texting constantly about things that should be written down, the trade was wrong and we should revisit a channel.
- 2. I may be dismissing Slack too fast, and it is Brett's call to overrule me.** The objection is "*I will not live in Slack*" — and that is fair. But there is a version nobody has priced: **one channel, used only as a Claude workbench, not as a workplace.** No teams, no threads, no notifications, no culture. Just a room where Claude has shared memory and both of you can see what it did. That is a materially different proposition from adopting Slack as a company. **If the daily digest proves too slow within a month, this is the first thing to reconsider** — and Claude Tag remains the only shipped product with memory genuinely shared between two people.
- 3. The single biggest risk is not software, it is the Mac.** Everything durable is in git and safe. But the *automation* — the digest, the data refresh, the publish — runs on one laptop that sleeps, that cannot read its own repo folder because of a macOS privacy restriction, and that has a scheduled job failing every run right now. **This plan is only as reliable as that machine, and today that machine is not reliable.** Moving the repository off ~/Downloads and onto the Mac Mini is not a nice-to-have; it is the prerequisite that makes Layer 3 trustworthy. I would put it ahead of items 2 through 5.
- 4. a second person's setup is heavier than mine makes it sound.** The repository is large — several gigabytes of history, because ad videos were committed into it. A full clone is a real wait on a home connection. **The mitigation is that he does not need one:** Claude Code in the browser works against the repo with nothing installed locally. But if he wants to run the tools himself, he needs a local clone and a Python environment, and that is an afternoon, not ten minutes. I should not pretend otherwise.
- 5. "Messages are files" is not intuitive to a newcomer.** It works because neither person touches the files directly — their Claude does. But the first time someone hears it, it sounds like homework. The onboarding guide has to carry that weight, and if a third or fourth person joins and it does not click for them, the design has failed and I will have been wrong about it.
- 6. Three identities is two too many.** Claude on brett@youchews.com. GitHub on brettob2424. Cloudflare on brettobrien24@gmail.com. I am recommending we *document* rather than *consolidate*, because consolidating has real migration cost. That is the pragmatic call, but it is not the clean one, and every future confusion about "which account is this under" traces back to this decision.

What is new and genuinely working best

- **Claude Code in the browser** is the quiet unlock. a new teammate needs no clone, no install, no terminal — just the repo authorised and a login. It removes the single biggest onboarding barrier in this plan.
- **Cloudflare Access with email codes** is better than any password scheme we would design. Free to fifty people, nothing to install, nothing to remember, and it works for people who will never have a Claude account.
- **Cowork's scheduled cloud tasks** are the most interesting thing on the horizon for us specifically, because our scheduled work keeps failing on a sleeping laptop. Running the refresh in Anthropic's cloud instead would remove the fragility entirely. **This needs proving with our own connectors before we rely on it** — an earlier attempt found cloud sessions could not see them — but it is the highest-value experiment available right now.

- **Private plugin packaging** means our whole harness — the rules, the tools, the gates — can eventually be installed by a new teammate in one command, in any repository. Not needed today because repo access delivers it, but it is what makes person number five cheap.

Where things are heading, and what to watch

Agents are becoming members of teams rather than tools that individuals use. That is the through-line in everything shipping right now — Claude Tag putting one Claude in a shared room, Buzz giving agents their own identity and permissions, compliance tooling extending to cover agent sessions the way it covers people. **The direction of travel is that an agent will have a seat, not a subscription.**

Our architecture is compatible with that, deliberately: the Vault is where a future agent-with-a-seat would read and write, exactly as our sessions do now. **We are not betting against the trend — we are keeping the memory in a place any future agent can reach.**

Three specific things to watch:

1. **Claude Tag beyond Slack.** Anthropic has signalled Microsoft Teams; nothing is confirmed and there is no date. If it lands somewhere Brett will tolerate, the channel layer becomes worth revisiting immediately.
2. **Shared memory between people.** It does not exist yet in any product, and it is an openly requested feature. If it ships, some of Layer 3 becomes redundant — which costs us nothing, because files never become the wrong answer.
3. **Buzz maturing.** If the hosted version stabilises, it is the only product that puts humans, agents and our code in one room without us running infrastructure. Worth a look each quarter, not a migration this one.

The honest summary

What I am most confident about: the Vault. Keeping durable knowledge in files we own is right regardless of what any vendor ships, and it is already working — it just needs one invitation to work for two people.

What I am least confident about: the Wire. A daily email is the simplest thing that could possibly work, and simple things sometimes turn out to be too simple. **Give it a month. If either of you finds yourself texting the other about something that should have been written down, tell me and we will fix the layer rather than defend it.**

What would change my mind fastest: a teammate trying it and reporting friction. That is the only real test, and it costs one invitation to run.