

SECURITY

Your code stays on your machine

Redshift Hub is a native macOS app, not a service you upload anything to. We operate no server that receives your repositories. This page lists exactly what does leave your machine, so your security team can check it rather than take our word for it.

RUNS LOCALLY

SIGNED & NOTARIZED BY APPLE

CREDENTIALS IN YOUR KEYCHAIN

01 Where your data lives

YOUR REPOSITORIES

- › Redshift reads and writes the repositories you open, on your disk. Nothing is copied to us.
- › Conversation history and session state are stored locally in your home directory.
- › We hold **no copy of your source code**, because there is nowhere for it to go. Redshift ships with no backend of its own.

YOUR CREDENTIALS

- › Tokens for GitHub, Linear and the AI providers are stored in the **macOS Keychain**, protected by the operating system and your account.
- › They are not marked synchronizable, so they never replicate to iCloud or another device.
- › They are sent only to the vendor that issued them. No credential is transmitted to us, and we have no way to read one.

ENCRYPTION AT REST

- › Local files inherit macOS disk encryption. We recommend FileVault, and require it if you work on client code.
- › Redshift adds no encryption layer of its own on top of the Keychain and the file system, and we would rather say so than imply protection you do not have.

02 What leaves your machine

The complete list. Everything travels over HTTPS with TLS, and the app makes no plaintext connections and applies no certificate exceptions.

TO THE AI VENDOR YOU CHOOSE

- › Redshift launches agent command-line tools: Claude Code, Codex, Gemini CLI, opencode and GitHub Copilot. Whichever you run receives the prompts and whatever repository content it reads.
- › That traffic goes from the agent to its vendor under your own account with them. It does not pass through us.
- › Your organisation controls which agents may run the same way it controls anything else on a developer machine, and for Claude Code specifically through **managed settings**, which Redshift cannot override.

TO SERVICES YOU HAVE CONNECTED

- › GitHub and Linear, using your own token, to show repositories, pull requests, reviews and notifications.
- › Optional commit-message generation sends the output of `git diff --stat`, which is file paths and change counts. Never file contents.

TO US

- › Anonymous product analytics through TelemetryDeck: app launches, which sections are used, and coarse timing. **No repository names, no file paths, no code.** Off with one switch in Settings.
- › Crash reports, harvested from the reports macOS already writes, and only for Redshift itself. Never another application.
- › An hourly update check against our appcast, which sends nothing but the request.
- › Feedback you choose to send, by email to hello@redshifthub.com. There is no in-app upload and no attachment service.

03 What the agent can reach

Worth stating plainly, because it is the part that matters most and the part a marketing page usually skips.

- › Redshift runs outside the macOS App Sandbox. A tool whose job is to run arbitrary build and git commands cannot work inside it. In practice the app can reach what your user account can reach.

- › Every action an agent wants to take goes through the agent's own approval prompt. Redshift renders that prompt; it does not decide it.
- › The approval mechanism is deliberately fail-safe: when a decision cannot be delivered, Redshift returns no decision at all and the agent falls back to asking you.
- › Agents read files in your repository as part of their work, so instructions planted in a file can influence one. This is true of every agentic coding tool. Keep approvals on for code you did not write.

04 Update integrity

- › Distributed as a **Developer ID signed** build, notarized by Apple.
- › Updates are served through Sparkle and signed with an **EdDSA key** whose public half is compiled into the app. An update that is not signed by us is refused.
- › The update feed is served over HTTPS.

05 How we build and release

Redshift Hub is built by one person. We put that here rather than in a footnote, because a supplier assessment will establish it anyway, and because it decides which of the controls below are real and which would be theatre.

GETTING A CHANGE IN

- › Every change reaches the release branch through a pull request. That branch takes no direct pushes.
- › Before every release, the app is compiled from clean on our own machine, in the same Release configuration that ships.
- › The test suite, currently **over 600 tests**, runs on that same machine before a release is cut.
- › Secrets stay out of the repository by design, never in source and never in version-controlled configuration. Credentials are read from the Keychain or the environment at runtime.

GETTING A RELEASE OUT

- › Releases are cut by a scripted pipeline, so the same steps run every time and the shipped state can be reconstructed from the repository.
- › Every build is signed with our Developer ID identity, hardened at runtime, and notarized by Apple before it is published.
- › The publishing script refuses to write an update feed it could not sign. A release that cannot be signed does not ship.

WHAT WE DO NOT HAVE

- › A second engineer reviewing the first one's code. There is no second engineer.
- › What stands in its place: automated review on every change, and dedicated regression coverage on the approval layer, including a test that proves it **fails closed** rather than open. That is a compensating control, not an equal one, and we would rather label it correctly than let a questionnaire read it as a yes.

06 Dependencies and patching

- › The app ships 19 third-party Swift packages, every one pinned to an exact version rather than a range.
- › The full inventory is the Package.resolved file in the build, and we will provide it, or a formal SBOM, on request.
- › Our working targets once a vulnerability is known: **critical and high within 30 days**, medium within 60, low when it next makes sense. A fix reaches you through the normal update channel.
- › Where a dependency issue could affect data on your machine rather than ours, it goes in the release notes rather than shipping quietly.

07 If something goes wrong

- › One address handles it: **hello@redshifthub.com**. There is no tier one to get past.
- › We aim to acknowledge within two working days, and faster than that for anything with a live security impact.
- › Where an incident affects personal data we hold, we notify affected users without undue delay and within 72 hours of becoming aware, per GDPR Article 33.
- › Our fastest remedy is an emergency release. A signed build reaches every install through the same update channel described above, without anyone having to go and fetch it.
- › Because we hold no copy of your code, an incident on our side cannot expose it. That is the point of the architecture, and it bounds what any incident here can cost you.

08 Personal data, and which role we hold

WHAT WE ACTUALLY HOLD

- › An account identifier, and only if you choose to sign in. Sign-in is optional and the app works fully without it.

- › Any email you send us, for as long as we need it to answer you.
- › Anonymous analytics carrying no identifier we could resolve back to a person, which stop entirely when you switch them off.
- › That is the complete list. There is no user profile, no usage history tied to a person, and no record of what you worked on.

CONTROLLER OR PROCESSOR

- › For the account identifier and your correspondence we are the controller, and the basis is performing our contract with you.
- › We are **not a processor of your customers' data**, because none of it reaches us. If your process needs a DPA regardless, ours is at [/dpa](#) and we will sign it.
- › Every provider that processes anything on our behalf is listed at [/subprocessors](#), with its location. We give 30 days' notice before adding one.
- › Ask us to delete your account data and we will, then confirm it. There is very little to delete, which is the intended outcome.

WHO ON OUR SIDE CAN REACH IT

- › One person, on accounts protected by multi-factor authentication.
- › Our database is closed to client access at the rules layer, so the app itself cannot read or write it even while you are signed in.
- › We keep no server-side record of what you do in the app, because there is no server for it to reach.

09 If Redshift Hub goes away

The question every assessment of a small vendor arrives at, and one we would rather answer directly than leave you to infer.

- › The app on your machine keeps working. It needs no server of ours to open a repository, run an agent, or use git, and nothing it depends on is licence-checked against us at launch.
- › Your work is already where it belongs: in your repositories, on your disk, under your own git remote. There is no export to run and no data to retrieve from us, because we never held it.
- › What you would lose is future updates and support. Not access, and not your work.

10 Reporting a vulnerability

Email **hello@redshifthub.com** with enough detail to reproduce it. We aim to acknowledge within two working days and will tell you honestly what we can fix and when. Redshift Hub is a small product; you will be talking to the person who wrote the code.

We hold no third-party security certification. If your process needs one, say so early and we will tell you plainly whether we can meet it rather than let a questionnaire imply otherwise.

Questions from your security team?

Send the questionnaire. We would rather answer a hard question accurately than return a page of badges.

Last updated: September 10, 2026