

# SOSA DevOps

Sovereign Systems - Privacy Policy

---

## Privacy Policy -- SOSA DevOps

Effective: 2026-05-15

Applies to: SOSA DevOps desktop application, version 1.0 and later

Controller: Sovereign Systems, Thailand

Contact: [privacy@sovereignsystems.cc](mailto:privacy@sovereignsystems.cc)

This policy explains what SOSA DevOps stores, where it stores it, what it does not store, and what authority you retain over your own data. It is written in plain language because you should be able to read it.

The single most important fact: SOSA DevOps is a local-first desktop application. The two files that hold your AI interaction data -- Vault A and Vault B -- never leave your device. There is no upload path, no sync path, no telemetry path, and no analytics pipeline that touches them. The architecture has no code that does any of these things, and we will not add such code without a material change to this policy and a 30-day notice.

### 1. Scope and Identity

This policy covers the SOSA DevOps desktop application v1.0 and later, distributed by Sovereign Systems (Thailand) for Windows, macOS, and Linux.

It does not cover the marketing site at [sovereignsystems.cc](https://sovereignsystems.cc), which has its own separate privacy policy. The marketing site is a website with the analytics, cookies, and processor relationships typical of a website. The desktop application is none of those things.

For questions about this policy, contact [privacy@sovereignsystems.cc](mailto:privacy@sovereignsystems.cc).

### 2. What We Collect, and Where It Lives

SOSA DevOps stores everything on your machine. There is no user account on a Sovereign Systems server. There is no cloud profile. There is no remote database with your name on it.

The data the application writes lives in your operating system's per-user application data directory:

%APPDATA%\com.sovereignsystems.sosa-devops\ on Windows

~/Library/Application Support/com.sovereignsystems.sosa-devops/ on macOS

~/.config/com.sovereignsystems.sosa-devops/ on Linux

Inside that directory, the privacy-relevant files are:

vault\audit-chain.jsonl -- Vault A. An append-only chain of cryptographic seals. Each

## SOSA DevOps

Sovereign Systems - Privacy Policy

---

entry contains a timestamp, an event type, character counts, and content hashes. It does not contain the raw text of anything you typed or anything the AI said. The hashes are one-way; nothing inside Vault A can be reversed back into your prompts or responses. Vault A is described in detail in §4.

vault\interaction-log.jsonl -- Vault B. The readable record of every AI interaction. It contains your prompts, the AI's responses, and any retrieved RAG (retrieval-augmented generation) context that was used in a turn. It is stored in cleartext on your disk. Vault B is described in detail in §5.

profiles\<id>\\* -- your local profile. The application supports multiple profiles on a single machine (for example, a personal profile and a work profile on the same laptop). Each profile directory contains:

- \* a display name and a hashed authentication identifier (the raw identifier is never stored -- see §7),
- \* persona configuration files,
- \* chat session files (`sessions\<id>.jsonl`),
- \* RAG corpora (the documents you have indexed for retrieval).

Application settings -- what we call "Deep Freeze" state. Locale, theme, accessibility preferences, lockout behaviour, the active workspace, the persona you last selected, the model you last selected. None of this is personally identifying; we list it here for completeness so the inventory above is exhaustive.

That is the full inventory. Nothing else is written to disk by SOSA DevOps in v1.0 outside the standard caches that your operating system or the underlying frameworks (Tauri, WebView2 on Windows, WKWebView on macOS) maintain on their own.

### 3. What We Do Not Collect

This list is short on purpose. SOSA DevOps does not collect any of the following:

- \* **\*\*No telemetry.\*\*** None. The application does not send usage statistics, feature counts, error rates, session lengths, or any other operational signal back to Sovereign Systems.
- \* **\*\*No off-device crash dumps.\*\*** If the application crashes, it crashes locally. There is no Sentry, no Crashlytics, no remote error-tracking service. Whatever your operating system does with crashes (Windows Error Reporting, macOS crash logs) is between you and your operating system.
- \* **\*\*No usage analytics.\*\*** We do not know which features you use, how long you spend in chat, how many sessions you have, or what models you have pulled.
- \* **\*\*No prompt or response content for "model improvement."\*\*** Your prompts and the AI's responses are never sampled, batched, or transmitted for any training, fine-tuning, or evaluation pipeline. The local model you run via Ollama does not learn from you either -- Ollama does not perform online training; the model is a frozen artifact.
- \* **\*\*No third-party trackers.\*\*** The application embeds no advertising SDKs, no

## SOSA DevOps

Sovereign Systems - Privacy Policy

---

marketing pixels, no fingerprinting libraries.

\* **\*\*No browser cookies.\*\*** SOSA DevOps is a desktop application built on Tauri, not a website. It does not use browser storage of any kind for application data.

\* **\*\*No identity leakage in the vaults.\*\*** Vault A and Vault B do not contain your name, your email address, or any account identifier. The internal session and interaction identifiers are random ULIDs.

The application's network traffic in v1.0 is limited to:

1. Ollama localhost (127.0.0.1:11434) -- the local AI runtime that you install separately. This traffic stays on your machine; 127.0.0.1 does not leave the device.
2. Tauri application updater check -- deferred to v1.1 and not wired in v1.0. When wired, it will perform a version-check request against an update endpoint we operate. That request will contain the application version, the platform, and nothing else. There will be no user-identifying payload.

That is the complete network surface of v1.0.

### 4. Vault A -- The Audit Chain

Vault A is the cryptographic record of what happened. It is designed for forensics, not for reading.

What it contains. Each line in audit-chain.jsonl is a sealed event: a timestamp, an event type (for example, a chat turn occurred, a corpus was reindexed, an export was generated), small numeric metadata such as character counts and token counts, and content hashes. A hash is a fixed-length fingerprint produced by a one-way function (SHA-256 in our case). You cannot recover the original content from the hash. Two different prompts will produce two different hashes; the same prompt typed twice will produce the same hash.

What it does not contain. Vault A never contains the raw prompt text, the raw response text, retrieved RAG content, file contents, or anything else that could be read back as a sentence. It is opaque by design.

How you access it. Open the application, then Tools ? Open Audit Vault. The viewer renders Vault A entries with a type-to-confirm gate before showing the chain -- this is a deliberate friction point so that the audit surface is never opened by accident. From the viewer you can also reveal the file in Explorer (Windows) or Finder (macOS).

Why it cannot be deleted from inside the app. Vault A is the chain that proves that something happened. If a user could delete entries from inside the application, the chain would lose its forensic value. The file lives on your disk; if you choose to delete it manually from the operating system you can, but the application itself does not provide a delete-Vault-A surface. This is a deliberate choice.

## SOSA DevOps

Sovereign Systems - Privacy Policy

---

### 5. Vault B -- The Interaction Log

Vault B is the readable record of every AI interaction. It is designed for you, not for forensics. The audience is the user themselves, doing model evaluation, fine-tuning curation, reviewing AI behaviour, or spotting an answer that was wrong.

What it contains. Each line in `interaction-log.jsonl` is a single interaction: a user prompt, an AI response, or a RAG retrieval step. The fields are:

- \* ``interactionId`` -- a random ULID identifying the row.
- \* ``sessionId`` -- the chat session the row belongs to.
- \* ``timestampMs`` -- when the row was written (UTC milliseconds).
- \* ``direction`` -- the kind of event (``user_to_local``, ``local_to_user``, ``rag_retrieval``, and a few others reserved for v1.1).
- \* ``actor`` -- who or what produced the content (``user``, ``local_model``, ``rag``).
- \* ``model`` -- the model that produced the row, if any.
- \* ``content`` -- the readable text: your prompt, the AI's response, or the joined RAG chunks.
- \* ``contentClass`` -- ``raw_user``, ``raw_assistant``, or ``rag_chunk``.
- \* ``tokensIn``, ``tokensOut``, ``latencyMs`` -- numeric metadata for model rows.
- \* ``linkedAuditSeal`` -- the hash of the matching Vault A entry, so you can cross-reference the two vaults.

The full schema is documented in `docs/architecture/INTERACTION_LOG_SCHEMA.md` and is locked at v1. Future application versions may add fields, but they will not break readers of v1 data.

Cleartext on disk. Vault B is not encrypted at rest in v1.0. The file is plain JSONL; if someone with access to your computer opens it, they can read your prompts and the AI's responses. This is a deliberate v1.0 trade-off: Vault B is for you, and we have prioritised your ability to read it (and to delete it) over an encryption story we did not have time to ship correctly. Encryption-at-rest is on the post-launch roadmap as part of the Sovereign Scribe feature; until then, treat the application data directory the way you treat the rest of your machine -- protected by full-disk encryption (BitLocker on Windows, FileVault on macOS, LUKS on Linux) and a strong account password. We recommend you turn full-disk encryption on regardless; it is a baseline best practice.

How you view it. Open the application, then Tools ? Open Interaction Log. The viewer shows entries grouped by session, in reverse-chronological order. For each row you can see the prompt, the response, the RAG chunks if any, the model, the latency, and the linked audit seal.

How you export it. From the Vault B viewer you can export any session in three formats:

- \* `**JSONL**` -- the raw v1 schema. Best if you want to pipe the export through a script or import it into another tool.
- \* `**TXT**` -- a human-readable transcript with the SOSA DevOps brand header and footer.

## SOSA DevOps

Sovereign Systems - Privacy Policy

---

\* **\*\*PDF\*\*** -- the same transcript rendered as a portable document, including an integrity certificate that lists the row count, the time range, and the cryptographic hash of the export.

You can also export from the session list directly (Synopsis ? ? menu on a session ? Export) which gives you the same three-way picker without going through the viewer.

How you delete it. From the Vault B viewer, select a session and choose Delete. The application asks you to type the session identifier to confirm -- this is the type-to-confirm gate the documentation refers to. When you confirm, the deletion is performed by an atomic rewrite: the application reads the file, drops every row that belongs to that session, and writes the result back. Other sessions are preserved. The deletion is logged to Vault A as `audit.interaction_log.session_deleted` with the row count, but the audit entry contains only the count and the session identifier hash, never the deleted content.

You can also delete the entire `interaction-log.jsonl` file by deleting it from your operating system file explorer. The application will create a fresh, empty file next time it writes a row.

What v1.0 does not yet ship. A bulk "delete everything" command from inside the application is on the v1.1 roadmap. Until then, the per-session delete is the in-app surface and the OS-level delete is the bulk surface.

### 6. v1.0 Known Limitation -- Plaintext Prompts and the PII Scrubber

You should know about one specific limitation before you use the application.

The architecture includes a component called the PII Scrubber. Its job is to detect credentials, personal identifiers, and other sensitive strings in your prompts and replace them with redaction markers before they are written to Vault B (and, in a future version, before they are sent to an external API). The Scrubber is a v1.1 deliverable. It is not wired in v1.0.

The practical consequence: in v1.0, your prompts are written to `interaction-log.jsonl` as you typed them. If you paste a password, an API key, an OAuth token, a database connection string, or any other secret into a chat, that secret will land in Vault B in cleartext. It will stay there until you delete the session or the file.

We have decided that you should hear this directly rather than have it buried in a footnote. The mitigations available to you in v1.0 are:

1. Don't paste secrets into chat. This is the simplest and most effective control.
2. If you do paste a secret, delete the session. The Vault B viewer's per-session delete is the surface for this. The deletion is atomic; other sessions are preserved.
3. Use full-disk encryption on your machine. It does not change what is in Vault B, but it changes who can read it.

## SOSA DevOps

### Sovereign Systems - Privacy Policy

---

The architecture takes a deliberate "log first, scrub later" position rather than "scrub first, ship later." Logging is reliable and visible to you; scrubbing is hard to get right and we will not ship a Scrubber that gives you a false sense of security. When the Scrubber ships post-launch, this section of the policy will be revised.

## 7. Local Profile Authentication

If you set a profile lockout in SOSA DevOps, the authentication identifier you use to unlock the profile is hashed at the Rust boundary using Argon2id before it touches any storage. The raw identifier is never written to disk, never logged, never appears in audit entries, and never appears in UI state.

If you forget your authentication identifier, there is no recovery path. There is no cloud backup, no email reset link, no support ticket that can recover the profile. The hash is one-way by design. We recommend using a password manager.

If you lose the device, the profile is gone with it. This is the cost of a no-cloud architecture.

## 8. External API Providers -- Not Wired in v1.0

The application's architecture supports user-keyed external API providers -- services like DeepSeek, OpenAI, Anthropic, Google, and others. The "External Providers" tab in the application's settings exposes this surface.

In v1.0, no external providers are wired. The tab shows the catalog with every provider marked "Disabled." There is no code path that sends a prompt, a response, a piece of RAG context, or any other content to an external service. The application's external network surface is limited to the Tauri updater check described in §3.

When external providers are wired in v1.1+, the following constraints will apply, and this policy will be revised to document them in detail:

1. You bring your own API key. The application does not proxy through any Sovereign Systems server. Your key, your account, your billing relationship with the provider.
2. Each provider's policy applies. When you send a prompt to OpenAI through the application, OpenAI's privacy policy and terms of service govern that request. We do not (and cannot) override or extend their handling.
3. The Privacy Filter is the boundary. Before any request leaves the device, it passes through the Privacy Filter -- a preflight stage that inspects the outbound payload and either redacts or refuses based on rules you control. The Privacy Filter is currently a Stage 1 stub in v1.0, present in the architecture but not enforcing redaction; full enforcement ships with the first concrete provider in v1.1+. Until that ships, the absence of any external provider wiring means the Filter has nothing to gate.
4. Vault A and Vault B continue to log. When a request goes to an external provider,

## SOSA DevOps

Sovereign Systems - Privacy Policy

---

Vault A seals the event and Vault B records the outbound prompt and the response, with the direction field marking it as external (`user_to_api`, `api_to_local`). You retain the same view, export, and delete authority over external-routed content as you do over local-routed content.

5. No external traffic is automatic. A request goes to an external provider only because you, in the moment, picked an external model in the chat composer or asked the agentic router to use one. The application has no background sync, no idle phone-home, no "warm-up" request.

The first concrete external provider on the roadmap is DeepSeek, scheduled for v1.1. Adding a provider is a code change with its own commit, audit entry, and policy revision. The set of available providers is bounded by what we have wired; you cannot add an arbitrary new provider yourself without modifying the application.

### 9. Local AI Models -- What We Provide and What We Don't

SOSA DevOps connects to a local Ollama runtime that you install separately. We do not bundle Ollama, we do not host Ollama, and we do not distribute Ollama on your behalf. Installation and updates of Ollama are between you and the Ollama project ([ollama.com](https://ollama.com)).

You also choose the model Ollama serves. Llama 3.1, Qwen 2.5, Mistral, Phi 3, Gemma -- the model catalog inside the application is descriptive metadata, not bundled weights. When you decide to use a model, you (in v1.0) drop to a terminal and run `ollama pull <model-name>`, and Ollama downloads the weights to your machine. (An in-application "pull" surface is on the roadmap as a v1.0 ship gap; the privacy posture does not change when it lands.)

The consequences for your data:

- \* **The model lives on your machine.** Inference happens locally. Your prompts are sent to `127.0.0.1:11434`, which is the Ollama HTTP server running on your loopback address. They do not leave your machine.

- \* **The model does not learn from you.** Ollama does not train models. The weights are a frozen artifact. Your prompts and the AI's responses do not become training data for any future version of any model -- not by Sovereign Systems, not by Ollama, not by the model's original author.

- \* **Each model has its own license.** Llama 3.1 is governed by the Llama 3.1 Community License. Qwen has the Qwen Research License or the Tongyi Qianwen License depending on size. Mistral has Apache 2.0 or the Mistral Research License. Gemma has the Gemma License. Embedding models like `nommic-embed-text` have Apache 2.0. **You are responsible for license compliance with the model you choose.** We provide model-card metadata in the catalog, but we do not enforce licensing at the runtime layer.

- \* **AI output quality is the model's responsibility, not Sovereign Systems'.** We do not curate, validate, or warrant the output of any model. See §6 of the Terms of Service for the AI disclaimer.

## SOSA DevOps

Sovereign Systems - Privacy Policy

---

### 10. Cross-Border Data Transfers

For v1.0, there are no cross-border data transfers. All your data is local. Nothing crosses any border because nothing crosses the boundary of your device.

When external API providers are wired in v1.1+, requests routed to those providers will involve a cross-border transfer (most providers operate from the United States, the European Union, China, or other jurisdictions). At that point this policy will be revised to document each provider's transfer mechanism, the safeguards in place, and your rights.

### 11. Your Rights -- PDPA (Thailand) and GDPR (European Union)

The Thai Personal Data Protection Act and the European General Data Protection Regulation grant you a defined set of rights over personal data. SOSA DevOps's local-first architecture changes how those rights are exercised, but does not diminish them.

Right of access. Everything we have about you is on your disk. Tools ? Open Interaction Log (Vault B) and Tools ? Open Audit Vault (Vault A) are the access surfaces. There is no remote database to query.

Right of correction. Generally not applicable -- we do not curate or transform the content you produce. If you believe a row in Vault B is incorrect (for example, your prompt was logged with a transcription error), you can delete the session and recreate it. We do not edit your logs on your behalf.

Right of erasure. Per-session delete from the Vault B viewer (atomic rewrite, type-to-confirm). For complete erasure, uninstalling the application removes the application binaries; you should also delete the application data directory (%APPDATA%\com.sovereignsystems.sosa-devops\ on Windows and the equivalent on macOS/Linux) to remove the vaults, profiles, and corpora. A v1.1 release will add an in-application "delete everything" command for convenience; until then, the OS-level delete is the bulk surface.

Right to data portability. Export your sessions in TXT, PDF, or JSONL format from the Vault B viewer or the session drawer's three-way picker. JSONL is the machine-readable format and conforms to the schema in INTERACTION\_LOG\_SCHEMA.md.

Right to withdraw consent. Withdrawing consent for SOSA DevOps's processing means uninstalling the application and deleting the application data directory. There is no "consent withdrawal request" to send anywhere because the application does not phone home; nothing remote will continue to process your data after you stop using the application.

Right to object / restrict processing. Equivalent to "stop using the application and delete the data directory" in our architecture, since processing happens on your machine



## SOSA DevOps

Sovereign Systems - Privacy Policy

---

under your control.

Right to lodge a complaint. Thai users may complain to the Personal Data Protection Committee (PDPC) at [pdpc.or.th](https://pdpc.or.th). EU users may complain to the data protection supervisory authority of their member state. We will cooperate with any lawful inquiry from these authorities.

If you wish to exercise any of these rights and have a question we have not anticipated here, write to [privacy@sovereignsystems.cc](mailto:privacy@sovereignsystems.cc).

### 12. Data Breach Notification

A data breach in the conventional sense -- unauthorised access to a database under our control -- is structurally limited by SOSA DevOps's architecture. We do not operate a database that holds your interaction data. There is no Sovereign Systems server with your prompts on it. A compromise of our infrastructure cannot expose your local vaults because your local vaults are not on our infrastructure.

If, despite this, we discover a security incident affecting any data we do hold (for example, the marketing-site mailing list, or the application's signing keys), we will notify the Thai PDPC and the relevant EU supervisory authority within the regulatory window (72 hours for both PDPA and GDPR), and we will notify affected users without undue delay if the incident materially affects them.

A breach against your local machine -- malware, theft, an unattended unlocked laptop -- is an incident on your side, not ours. We cannot detect it from here, and we have no notification obligation to you for it. The mitigations are the standard ones: full-disk encryption, a strong account password, and not leaving your machine unlocked.

### 13. Children

SOSA DevOps is intended for adult developers and professional users. We do not knowingly direct the application at children. Under the GDPR, processing personal data of children under 16 (or the lower threshold set by a member state, down to 13) requires parental consent; under PDPA, children under 10 require parental consent. The application has no child-specific surfaces and we do not solicit information about a user's age. Parents and guardians who wish to apply parental controls should rely on their operating system's family safety features.

### 14. Changes to This Policy

Material changes -- for example, adding telemetry, changing the local-only invariant, wiring an external provider, or introducing a new processor relationship -- will be announced 30 days in advance via an in-application banner, and the version of this

## SOSA DevOps

### Sovereign Systems - Privacy Policy

---

policy in the application's About panel will increment.

Minor changes -- clarifications, typo fixes, restructuring without semantic change -- will be documented in the project rollback ledger (docs/ROLLBACK\_LEDGER.md).

The "Effective" date at the top of this document is the date the current version was published.

## 15. Contact

- \* General privacy questions: ``privacy@sovereignsystems.cc``
- \* AI-specific questions: ``ai@sovereignsystems.cc`` (see also the AI Disclosure document)
- \* Legal questions: ``legal@sovereignsystems.cc``
- \* Security disclosure: ``security@sovereignsystems.cc``

Sovereign Systems  
Thailand

## SOSA DevOps

Sovereign Systems - Privacy Policy

---

### Document Integrity Certificate

---

**Document:** Privacy Policy

**Application:** SOSA DevOps v1.0.0

**Publisher:** Sovereign Systems, Thailand

**Effective Date:** 2026-05-21

**Generated:** 2026-05-21

**Source SHA-256:** 9b87f5db993f0392afab64a6d236a16c

**(cont):** 8393d210b49c727712980486c63b4fc2

This certificate records the SHA-256 digest of the markdown source from which this PDF was generated. It is provided for audit and tamper-evidence purposes. Sovereign Systems reserves the right to update these documents; the current version is always available in-application.