Intelligent Talk™

FOR SECURITY AND PROCUREMENT TEAMS

Security Packet

How Intelligent Talk™ Studio protects your data, what it does not do yet, and how we handle problems. In plain words.



Heir Lifestyle Investments LLC

v1.2 · November 11, 2026

CONTENTS

What is inside

1. **Security White Paper**
2. **Security Questionnaire Answers**
3. **AI Governance Statement and AI System Card**
4. **Incident Response Plan**
5. **Vulnerability Disclosure Policy**
6. **Support Terms**

Every claim is tied to code and an automated test (72 tests passing on Nov 10, 2026). Where we have not done something, we say so. Security and support contact: heirlifestyle@gmail.com. Policy documents are drafts until the approval line is signed.

Signed: this PDF has an Ed25519 signature (.sig file). Our public key is published at /.well-known/intelligent-talk-signing-key.pem on our website. A software bill of materials (CycloneDX 1.6, SHA-256 for every shipped file, 0 third-party packages) comes with this packet.

Intelligent Talk™ Studio: Security Whitepaper

Heir Lifestyle Investments LLC · Studio v0.2.0 · November 10, 2026 · plain-words edition

What this is. This paper explains how the Studio protects your data, in plain words, and what it does not do yet. Every control below is in the code and has an automated test (`npm test`: 72 tests, all passing on Nov 10, 2026). A release check fails the build if any number in this paper drifts from the code..

What we do not claim. We have **no SOC 2 report, no ISO 27001 or ISO 42001 certification, and no outside penetration test**. Nobody outside our company has audited this code yet. Where we point to a standard (OWASP, NIST), we are saying we designed to it, not that someone certified us.

Report a security problem: heirlifestyle@gmail.com. We acknowledge within 2 business days (see the Vulnerability Disclosure Policy).

---

1. The one-paragraph version

The Studio is a single Node.js program that runs **on your own server**. It has no outside software libraries (Node built-ins only), so there is no third-party package supply chain to attack. Your data stays in one folder on that server, encrypted with AES-256-GCM using a key only you hold. People sign in with their own accounts (or your company SSO), and each person gets a role that limits what they can do. Every sign-in, prompt run, approval, export, delete and settings change goes into an audit log that is hash-chained, so a changed or removed record shows up. Nothing is ever sent, published, deleted or spent on its own: a named person approves first.

2. Where your data goes (data flow)

```
Your staff's browsers ──HTTPS──> Your reverse proxy (Caddy/nginx) ──> Studio (Node, your server)
                                                                      │ encrypted data folder <──────────┤ (Brain, queue, history, (AES-256-GCM, your key) │ usage, accounts, audit log)
                                                                      │ Live mode only: prompt text ──HTTPS──> api.anthropic.com (your own API key)
                                                                      SSO only: sign-in redirect ──HTTPS──> your identity provider (Okta/Entra/Google)
```

- **Mock mode** (no API key): nothing leaves the server. Good for trials and demos.
- **Live mode**: the filled prompt and the Company Brain go to Anthropic's API, **under your own API agreement with Anthropic**. That is the only outside call. Anthropic's retention and training terms are between you and Anthropic. Their certifications are theirs, not ours.
- **We (Heir Lifestyle Investments LLC) receive nothing.** No telemetry, no analytics, no phone-home, no fonts or scripts loaded from outside. The page's Content-Security-Policy blocks outside requests even if code tried.

3. Encryption

What	How
Data at rest	**AES-256-GCM** for every file in the data folder: Company Brain, approval queue, run history, usage, approval log, settings, accounts, sessions, and each line of the audit log. Every write gets a fresh random 96-bit IV and a 128-bit authentication tag.
Key	`STUDIO_DATA_KEY`: 32 random bytes, base64. Set in the environment or (better) in a file via `STUDIO_DATA_KEY_FILE`, such as a systemd credential. **In production mode the Studio refuses to start without it.**
Key separation	The master key never encrypts data directly. Separate sub-keys come from HKDF-SHA256: one for files, one for keyed audit hashes and CSRF tokens.
Integrity on read	Each file is bound to its own name (authenticated additional data). A changed byte, a file copied over another file, a wrong key, or an unencrypted file all **stop the read with an integrity error**. They never quietly reset to empty, because a silent reset would let the next save wipe the evidence. `node tools/studio-admin.js verify-data` checks every file.
Key rotation	`node tools/studio-admin.js rotate-key` checks that everything opens with the old key, re-encrypts every file and every audit line with the new key, verifies again, and writes an audit entry. Every file write is atomic. If rotation is cut off halfway, set the old key in `STUDIO_DATA_KEY_OLD` and both keys work while you finish.
Moving from demo to encrypted	`node tools/studio-admin.js encrypt-data` encrypts a plain demo folder in place.
In transit	HTTPS at your reverse proxy (see DEPLOY.md). In production the Studio sends HSTS (2 years). Calls to Anthropic and to your identity provider are HTTPS only; the SSO code refuses any non-HTTPS endpoint.
Exports	The export bundle is encrypted separately, with a key made from a passphrase the admin picks (scrypt, then AES-256-GCM). You can hand it to your records team without sharing the server key.

Honest limits. While the program runs, data is decrypted in memory; anyone with root on the server can read it. The key sits in the server's environment or a credential file; we do not yet integrate with a cloud KMS or an HSM. Encryption protects a stolen disk, a stolen backup or a copied folder. It does not protect a fully compromised running server.

4. Sign-in, accounts and roles

Passwords. Hashed with **scrypt** (N = 32,768, r = 8, p = 1) with a random 16-byte salt per user, and compared in constant time. A sign-in for a username that doesn't exist still does the same scrypt work, so response time doesn't reveal who has an account. Minimum length: 12 characters (a short sentence works well).

Sessions. A session is 32 random bytes, handed to the browser once in a cookie named `__Host-its_session` with `HttpOnly; Secure; SameSite=Strict; Path=/`. The server stores **only the SHA-256 of the token**, so a copied data folder contains no usable session. Sessions end after **30 minutes idle**

or **12 hours total**, whichever comes first. Signing out, a password change, disabling an account or changing its roles ends that person's sessions at once.

Brute force. 5 wrong passwords in 15 minutes lock that account for 15 minutes, even if the next try is right. Made-up usernames lock the same way, so nobody can test which accounts exist. Sign-in attempts are also limited to 10 a minute per IP address. Failed sign-ins are logged as a keyed hash of the username, never the raw text, because people often type their password into the username box.

Roles (least privilege). The server checks the role on every request; the page only hides buttons for convenience.

Role	Can	Cannot
viewer	Read the library, Company Brain, queue, usage and history	Run prompts (spend money), change anything, approve
editor	Everything a viewer can, plus run prompts and edit the Company Brain	Approve anything, see the audit log, manage users or data
approver	Read, plus approve, reject or mark sent	Run prompts. **Nobody can approve something they created themselves (four-eyes rule).**
admin	Manage accounts, retention settings, audit log, export, and request or run hard deletes	Approve, unless also given the approver role. Delete without an approver.

One person can hold several roles. The system never lets the last active admin be removed or demoted.

Single sign-on (OIDC). Off by default. When turned on, the Studio uses the OpenID Connect authorization code flow with **PKCE (S256)**, a single-use 10-minute `state` bound to the browser by a short-lived cookie, and a `nonce`. Every ID token is checked for: an allowed signature algorithm (RS/PS/ES only; never `none`, never HMAC, which blocks the "public key as secret" trick), a valid signature against your provider's published keys, exact issuer, audience, expiry, issue time and nonce. Optionally it also checks the email domain. Roles come from your provider's groups or app-role claim. **A user with no mapped role gets no access.** Setup for Okta, Microsoft Entra ID and Google Workspace is in DEPLOY.md. SSO users get multi-factor sign-in from your provider.

Two-step sign-in for local accounts (TOTP, RFC 6238). Works with any standard authenticator app. The 160-bit secret is created by the server and stored only inside the encrypted store; it never appears in the user list, exports or logs. Codes are 6 digits on 30-second steps, with one step of clock drift allowed. **A code never works twice**: the last used time step is remembered, so a code seen over someone's shoulder or replayed is refused. Ten single-use recovery codes are shown once and stored only as SHA-256 hashes. Wrong codes count toward the same 5-try account lockout as wrong passwords. An admin can require two-step sign-in for chosen roles (`security.mfa_required_roles`, for example `["admin", "approver"]`). A user in such a role can only finish setup or sign out until it is on, and can't turn it off. Only a *different* admin can reset a lost device. If the only admin loses theirs, someone with server access and the data key runs the break-

glass command `studio-admin.js mfa-reset <user>`, which is audited. Setup, use of a recovery code, failures and resets are all written to the audit log. The code is checked against the RFC 6238 test vectors. Not included: WebAuthn and passkeys, and SMS codes (which we don't plan to add, because SMS is easy to intercept).

Demo and legacy modes. With no accounts and no password, the Studio runs in **local demo mode** and answers only this computer (loopback). A legacy single shared password (`STUDIO_PASSWORD`) still works for small offices with no accounts, but **production mode refuses it**. The four-eyes rule needs real separate accounts, so it applies only in accounts mode.

5. Tamper-evident audit log

- **What is logged:** sign-ins (success, failure, lockout), sign-outs, every prompt run and translation, every approval, rejection and "marked sent", exports, delete requests and executions, retention purges, Company Brain and retention changes, account changes, key rotation, access denials (wrong role, missing CSRF token), and server start.
- **What is never logged:** prompt text, pasted inputs, outputs, Company Brain text, passwords, tokens or keys. Instead the log holds **keyed hashes** (HMAC-SHA256) of the prompt and the output. An auditor who has a document can prove "this exact text was produced in run X", but nobody can read or guess the text from the log. The audit module also drops any metadata field named like a secret, and runs every string through the secret redactor as a backstop. A test plants a marker phrase, an API-key-shaped string and a password, then confirms none reach the log.
- **How tampering shows:** the same design as our `protect/chain.py` ledger. Each record holds the SHA-256 of the record before it, starting from 64 zeros. Changing, deleting or reordering any record breaks the chain from that point. With encryption on, each line is also sealed with AES-256-GCM, and a sealed "head" file records the last sequence number and hash, so cutting records off the end is caught too. Plain demo logs can be checked by the separate Python verifier (`python3 -I protect/chain.py verify data/audit.jsonl`). A test confirms both verifiers agree.
- **How to check:** `node tools/studio-admin.js audit-verify` (exit code 1 if tampered), or the **Admin** tab, which shows "chain intact" or "TAMPERED" with the first broken record.
- **Honest limit:** someone with full control of the server could roll the whole folder back to an older copy. To catch that, copy the latest head hash somewhere the server can't write (`node tools/studio-admin.js audit-head`, daily, into a ticket, email or WORM storage). Built-in forwarding to a SIEM or syslog is not done yet.

Traceability. Each run's audit record also stores:

- the model and auditor model
- the Studio version
- a SHA-256 fingerprint of the exact prompt text and auditor text that ran

So any output can be traced months later to the precise instructions and model behind it. Start-up records the version and models.

6. Approvals stay human

- The Studio **has no send, publish or payment feature at all**. Emails and texts (prompts 4, 11 and N1, plus anything run with an email or text channel) are saved as **drafts**. A person sends them outside the Studio, then clicks "mark sent". The server refuses "mark sent" unless the item was approved first.
- With accounts, **the approver is the signed-in person** (not a typed name), and **the creator of a draft can never approve it**.
- **Hard deletes need two people**: an admin requests (typing DELETE), a different person with the approver role approves, then an admin runs it.
- Text produced by the model ("approve and send this now") cannot change any of this. Tests confirm that approving and marking sent make no outside calls.
- Spending: running prompts calls your Anthropic key. That is limited by role (editors and admins only), by rate limits (12 model calls a minute per IP and per user), and by the cost display on every run. An admin can also set a **monthly spending cap** in US dollars (`monthly_cap_usd` in Settings; empty means no cap, 0 means no model calls). Month-to-date spend is kept in its own encrypted counter, so trimming or retention of the usage history can never hide spend. Once the cap is reached, Translate and Run are refused before any model call, and the block is written to the audit log. Honest limit: the check runs before each call, so the call that crosses the cap can go over it by the cost of that one run. We still recommend a matching limit in your Anthropic Console as a second layer. **Kill switch:** set the cap to 0, and every model call stops at once. That change is audited too.

7. Data rights

- **Export:** Admin → "Download encrypted export" (or `node tools/studio-admin.js export`). One encrypted file holding the Company Brain, queue, run history, usage, approval log, settings, the account list (no password hashes) and the full audit log with its verification result. It opens only with the passphrase (`open-export`).
- **Hard delete:** scopes are all company data, run history, usage, queue, Brain, approval log, or one history or queue item, through the two-person flow above. The audit log is kept (it holds no content) and still verifies afterwards. Files are overwritten, then removed. **Honest limit:** on SSDs, copy-on-write file systems and backups, an overwrite may not truly erase the old data. The reliable way to destroy everything is **crypto-shredding**: rotate to a new key (or stop using the data), then destroy the old key and any backups made with it.

- **Retention:** run history **90 days** by default, usage 365, decided approvals 365. Admins change these in the Admin tab (1 to 3,650 days). The purge runs at start-up, hourly, and on every change, and each purge is logged. Drafts nobody has decided on are never auto-deleted.

8. Web and input safety

- **Content-Security-Policy:** only this server's own script and styles; no inline code, no `eval`, no plugins, no framing (`frame-ancestors 'none'`), `base-uri 'none'`, and **Trusted Types enforced**, so text can never be parsed as HTML. The page builds everything with `textContent`, and a test fails the build if `innerHTML`, `eval` or similar ever appear. Also sent: `X-Content-Type-Options: nosniff`, `X-Frame-Options: DENY`, `Referrer-Policy: no-referrer`, `Cross-Origin-Opener-Policy` and `Cross-Origin-Resource-Policy: same-origin`, `Cache-Control: no-store`, and HSTS in production.
- **CSRF, three layers:** (1) every state-changing request must be JSON, which a plain HTML form cannot send; (2) cross-site `Origin` or `Sec-Fetch-Site` is refused; (3) with accounts, each request must echo the session's CSRF token in an `X-CSRF-Token` header (HMAC of the session, checked in constant time).
- **Limits:** request bodies up to 200 KB, URLs up to 4 KB, each field up to 40,000 characters, a 15-second header timeout and a 60-second request timeout (against slow-client attacks), 120 requests a minute per IP and per user, 12 model calls a minute, and 10 sign-in attempts a minute per IP.
- **Safe-Paste Screener (built in):** before any model call, every pasted value is screened:
- **Invisible characters are removed**:
  - zero-width spaces and the soft hyphen
  - bidirectional embeddings, overrides and isolates ("Trojan Source")
  - Unicode tag characters ("ASCII smuggling")
  - supplementary variation selectors ("emoji smuggling")
- **Real languages and emoji are kept:** the joiners Persian, Hindi and family emoji need, right-to-left marks, and genuine flag emoji. Tests prove each one passes untouched.
- Hidden commands are also looked for on a **normalized copy**, so look-alike letters (fullwidth or styled "math" letters) can't disguise them.
- The screener runs on **every** pasted field: prompt values, the Translate sentence and its follow-up answers, and the Company Brain. The Brain feeds the system prompt, so invisible characters are always removed there, whatever the mode.
- **Hidden-command patterns are flagged.**

- **Personal data** (email addresses, phone numbers, US Social Security numbers) is noted on the result, so a person sees it before using the output.

The default mode flags and continues. `security.screen_mode: "block"` refuses the run, before anything is sent, whenever hidden commands or invisible characters are found.

- **Secrets in pasted text:** API keys, card numbers (checked with the Luhn rule) and `password: ...` lines are replaced with `[REDACTED]` before anything is stored or sent.

9. LLM safety layers (mapped to OWASP Top 10 for LLM apps, 2026)

Layer	What it does	Where it runs	OWASP
Untrusted-data wrapper	Every pasted value goes inside `<untrusted_data>` tags; the system prompt says that text is never an instruction; text that tries to close the tag early is defused	Studio server, every run	LLM01 prompt injection
Injection flags	Pasted text that looks like hidden commands ("SYSTEM: ignore rules", `<script>`) is flagged on the result	Studio server	LLM01
Integrity block	Inside each of the 20 library prompts: pasted material is data, no invented facts, links only from the user, `[REDACTED]` for secrets	Inside the prompt, read by Claude	LLM01, LLM02, LLM09
A1 Output Auditor	A second Claude pass checks every claim against the brief and the original prompt, fixes or marks `[NEEDED]`, and shows the fixes	Studio server, every run	LLM09 misinformation
Company Brain banned terms	The server checks the final output for banned terms	Studio server	LLM05 output handling
Human approval gate	No send, publish, delete or spend on its own (section 6)	Studio server	LLM06 excessive agency
Rate limits, cost display, monthly cap	Model-call limits per IP and per user, cost on every run, an admin-set monthly dollar cap	Studio server	LLM10 unbounded consumption
Safe-Paste Screener	Code version, built in: removes invisible and Unicode tag characters, flags hidden commands, notes personal data; optional block mode stops the run before any model call. The fuller S1 prompt (it also judges risky links and context) stays available in the library for manual use	Studio server, every run and translate	LLM01, LLM02

Honest limits: no prompt-injection defense is complete (Anthropic says the same). Our red-team numbers (0 safety failures across 40 tests in prompt version 8) come from **our own simulated graders**, not an independent lab. The Studio's own injection tests cover the server's handling, not the model's judgment.

10. Supply chain and code

- **Zero runtime dependencies**: Node.js 22+ built-ins only (`crypto`, `http`, `fs`, `node:test`). `package.json` lists no packages, so there is no `npm install` step to poison.
- Every release runs one gate script: prompt lint, `npm test`, brand and page checks, a workspace secret scan, and a proof-of-creation hash refresh.
- The server removes the data key, the SSO client secret and the API key from its environment once it has read them at start-up.

11. What we do NOT have yet

Gap	What it means for you	What closes it
SOC 2 Type I/II	No independent audit of our controls	A readiness program and an auditor (costs money; only needed if we host)
Outside penetration test	Only our own tests have looked for holes	A pen test by a CREST/OSCP-certified firm before the first large deployment
ISO 27001 / 42001	Not certified	An alignment statement first; certification later
SAML and SCIM	SSO is OIDC only; accounts are not provisioned automatically from your directory	SCIM 2.0 endpoint; SAML if a buyer requires it
Passkeys / WebAuthn	Local two-step sign-in uses authenticator-app codes (TOTP); phishing-resistant passkeys aren't built yet	WebAuthn for local accounts, or use SSO with your provider's passkeys
KMS/HSM key storage	The key is in an environment variable or file	AWS KMS, Azure Key Vault or GCP KMS envelope encryption
External audit anchoring / SIEM export	Rollback of the whole folder is caught only if you keep the head hash elsewhere	Syslog/SIEM forwarding and scheduled head anchoring
High availability	One process, one server; the lockout counter resets on restart	A shared store and multi-node support

Gap	What it means for you	What closes it
Uptime SLA, cyber insurance, DPA	Incident response plan, vulnerability disclosure policy and support terms are **written as drafts awaiting owner approval** (INCIDENT_RESPONSE_PLAN.md, VULNERABILITY_DISCLOSURE_POLICY.md, SUPPORT_TERMS.md). No uptime SLA (self-hosted), no insurance, no signed DPA	Owner approval of the drafts; insurance and DPA are owner decisions
Live SSO test	SSO was tested offline against a simulated provider with real signed tokens, not yet against a live Okta, Entra or Google tenant	One live test per provider before the first SSO customer

Enterprise AI Security Questionnaire: Pre-filled Answers

Heir Lifestyle Investments LLC · Intelligent Talk™ Studio v0.2.0 (self-hosted) · November 10, 2026 · DRAFT for owner review

How to use this. These are the 40 questions enterprise security and procurement teams ask most often about AI software. They are drawn from our buyer research (`prompts/security/PRO_STANDARD.md`: OWASP LLM Top 10 2026, CSA AI-CAIQ/SIG, the "80-question wall"), `business_suite/enterprise_readiness.md`, and the controls in `SECURITY_WHITEPAPER.md`. Copy an answer into the buyer's form. **Never upgrade a "No" or "Partial" to a "Yes".** Answers apply to the **self-hosted Studio**, where the customer runs it on their own server. If we ever host it for a customer, rows 1–4, 6, 22, 24 and 31–36 change and SOC 2 becomes necessary.

Status key: Yes = built and tested · **Partial** = some of it, the gap is stated · **No** = not today · **N/A** = does not apply to self-hosting · **Owner** = needs an owner decision or money.

A. Company, data and privacy

#	Question	Status	Answer
1	Where is our data stored and processed?	Yes	On your own server, in one data folder, encrypted. We (the vendor) receive nothing. In live mode, prompt text goes to Anthropic's API under **your** Anthropic agreement.
2	Does the vendor have access to our data?	Yes	No. There is no remote access, telemetry, analytics or phone-home. The page's CSP blocks any outside request. Setup help is screen-share only, on your terms.
3	Is our data used to train models?	Yes	We train nothing and receive nothing. Anthropic states it does not train on commercial API data by default. Confirm in your own Anthropic terms.
4	What sub-processors are used?	Yes	None from us. Anthropic (live mode) and your identity provider (SSO) are your own vendors, under your contracts.
5	What is the data retention period, and can we change it?	Yes	Run history 90 days by default, usage 365, decided approvals 365. Admins set 1–3,650 days. The purge runs hourly and is audit-logged. Anthropic API retention is set in your Anthropic account.
6	Can we export all our data?	Yes	Admin → Export: one AES-256-GCM bundle locked with a passphrase you choose (scrypt key derivation). It includes Brain, queue, history, usage, approvals, settings, accounts (no password hashes) and the full audit log.
7	Can we permanently delete our data?	Partial	Yes, through a two-person flow (admin requests, a different approver approves, admin runs it), and it is logged. Files are overwritten and removed. On SSDs and in backups, an overwrite may not truly erase the old data, so for certainty, rotate the key and destroy the old one (crypto-shredding).

#	Question	Status	Answer
8	Do you sign a DPA (GDPR/CCPA)?	N/A / Owner	Self-hosted: we are not a processor because we receive no personal data. If a setup service touches your data, an owner-approved DPA is needed first.
9	Can it handle regulated data (PHI, PCI, privileged legal)?	Partial	Card numbers, API keys and `password:` lines are auto-redacted before storage or sending. PHI only if you have a BAA with Anthropic for the API. No industry certification is claimed. Counsel should review.
10	Is data encrypted at rest?	Yes	AES-256-GCM on every data file and every audit line, with a fresh IV per write, HKDF-separated sub-keys, and the file name bound in as authenticated data. Production refuses to start without a key.

B. Identity and access

#	Question	Status	Answer
11	Do you support SSO?	Partial	Yes for **OIDC** (authorization code + PKCE S256), with documented setup for Okta, Microsoft Entra ID and Google Workspace. Off by default. **No SAML.** Tested offline against a simulated provider with real signed tokens; not yet against a live tenant.
12	Do you support SCIM provisioning?	No	Not yet. Accounts are created in the Admin tab or the CLI, or just-in-time through SSO with roles from group claims. Disabling a user in your IdP blocks new sign-ins; an existing session lasts at most 30 minutes idle or 12 hours total.
13	Is MFA supported?	Yes (TOTP) / Partial (passkeys)	SSO users: your provider's MFA. Local accounts: built-in two-step sign-in with any authenticator app (TOTP, RFC 6238), with replay protection, 10 single-use hashed recovery codes, lockout on wrong codes, and an admin setting to **require** it for chosen roles. Resets need a different admin and are audited. **No passkeys/WebAuthn yet**; use SSO for phishing-resistant MFA.
14	What roles exist (RBAC)?	Yes	viewer, editor, approver, admin. Checked on every server route. A user can hold several roles. Admin is not an approver unless explicitly granted. The last active admin cannot be removed.
15	How are passwords stored?	Yes	scrypt (N=32,768, r=8, p=1), 16-byte random salt per user, constant-time comparison, 12-character minimum. Hashes are never returned by any API or included in exports.
16	How are sessions protected?	Yes	32 random bytes in a `__Host-` cookie with `HttpOnly; Secure; SameSite=Strict`. Only the SHA-256 of the token is stored. 30-minute idle and 12-hour absolute timeouts. Sign-out, password change, disable and role change all end sessions at once.
17	Is there brute-force protection?	Yes	5 failures in 15 minutes locks the account for 15 minutes (unknown usernames too, so accounts can't be discovered). Sign-in is limited to 10 tries a minute per IP; the API to 120 requests a minute per IP and per user.

#	Question	Status	Answer
18	Is there separation of duties?	Yes	The creator of a draft can never approve it (four-eyes rule). Hard deletes need an admin and a different approver. Applies whenever accounts are on.

C. Logging and monitoring

#	Question	Status	Answer
19	What is audit-logged?	Yes	Sign-ins (success, failure, lockout), sign-outs, prompt runs, translations, approvals, rejections, marked-sent, exports, delete requests and executions, retention purges, Brain and settings changes, account changes, key rotation, access denials and server start.
20	Are logs tamper-evident?	Yes	SHA-256 hash chain (same design as our `protect/chain.py` ledger). With encryption on, each line is also GCM-sealed and a sealed head catches truncation. Checked by `studio-admin.js audit-verify` (exit code 1 if tampered) and in the Admin tab. Limit: a full rollback of the folder is caught only if you keep the daily head hash elsewhere (DEPLOY.md §8).
21	Do logs contain prompts, outputs or secrets?	Yes (they don't)	No. Metadata plus HMAC-SHA256 hashes of prompt and output only. Secret-shaped fields are dropped and strings are redacted. Tested with planted markers.
22	Can logs go to our SIEM?	Partial	Not built in. The audit log is a JSONL file and the export bundle includes it; the server's stdout logs (method, path, status, time only) go to journald. Syslog/SIEM forwarding is on the roadmap.
23	How long are audit logs kept?	Yes	Indefinitely by default; hard deletes deliberately keep them because they hold no content.

D. Application and infrastructure security

#	Question	Status	Answer
24	Do you have SOC 2 / ISO 27001 / ISO 42001?	No	**None.** We claim no certification. Anthropic holds its own certifications for its API, under your contract with them; we do not inherit them.
25	Has an independent penetration test been done?	No	Not yet. Our own automated suite (72 tests) covers auth, CSRF, encryption, audit tampering, injection and limits. A third-party pen test is planned before the first large deployment (Owner: budget).
26	What protects against XSS and clickjacking?	Yes	Strict CSP (self only, no inline, no eval, `object-src 'none'`, `frame-ancestors 'none'`, Trusted Types enforced), `textContent`-only rendering (enforced by a test), nosniff, COOP/CORP, no-store caching, HSTS in production.

#	Question	Status	Answer
27	What protects against CSRF?	Yes	JSON-only state changes, cross-site Origin and Sec-Fetch-Site refused, and a per-session CSRF token in a custom header, compared in constant time.
28	What input limits exist?	Yes	200 KB bodies, 4 KB URLs, 40,000 characters per field, 15 s header and 60 s request timeouts, rate limits per IP and per user.
29	What third-party components are used?	Yes	None at runtime: Node.js 22 built-ins only. `package.json` has zero dependencies.
30	How are keys and secrets managed?	Partial	The data key comes from a root-only file (systemd credential) or an environment variable, and is removed from the process environment after start-up. Rotation is built into the CLI and tested. **No KMS/HSM integration yet.**
31	How do you handle vulnerabilities and patches?	Partial	Every release passes one gate script (tests, secret scan, page checks). A vulnerability disclosure policy is written (VULNERABILITY_DISCLOSURE_POLICY.md: 2-business-day acknowledgement; fix targets 14/30/90 days by severity; good-faith safe harbor), **published at intelligenttalk.pages.dev/security.html, with machine-readable /.well-known/security.txt (RFC 9116); final owner sign-off pending**. No CVE numbering authority.
32	Do you have an incident response plan?	Partial (draft)	Written (INCIDENT_RESPONSE_PLAN.md): roles, four severity levels with start times, six steps (detect, contain, assess, fix, notify, review), signing-key compromise steps, customer notice aimed within 72 hours of confirming impact for S1/S2, yearly drill. **Draft awaiting owner approval.** A live drill of the signing-key compromise scenario ran on Nov 10, 2026 with throwaway keys and passed all 5 steps (record kept in `engineering/drills/drill_log.jsonl`).
33	What are your uptime SLA and support terms?	Partial	No uptime SLA: self-hosted, so uptime follows your server. Support terms are written (SUPPORT_TERMS.md): email, Mon–Fri 9–5 ET, first reply within 2 business days, Critical security issues started within 4 business hours. **Draft awaiting owner approval.**
34	Do you carry cyber or E&O insurance?	No / Owner	None today. Owner decision; usually required by enterprise MSAs.
35	Is there a backup and disaster-recovery process?	Partial	Your backups of an already-encrypted folder (DEPLOY.md §8), with `verify-data` after restore. Single server; no built-in high availability.

E. AI-specific risk (OWASP LLM Top 10, 2026)

#	Question	Status	Answer
36	How do you defend against prompt injection (LLM01)?	Partial	Layers: pasted text wrapped as `<untrusted_data>` with a guard in the system prompt (attempts to close the tag early are defused); injection flags; the integrity block in every prompt; the A1 auditor second pass; and the human approval gate, so a hijacked draft still cannot act. The built-in Safe-Paste Screener removes invisible and Unicode tag characters, flags hidden commands and notes personal data on every run; an optional block mode stops the run before any model call. No defense is complete.
37	Can the AI take actions on its own (excessive agency, LLM06)?	Yes (it can't)	No. The Studio has no send, publish, payment or delete tool for the model. Emails and texts are drafts; a named approver must approve; tests confirm approval triggers no outside call.
38	How do you limit hallucination or misinformation (LLM09)?	Partial	Every run gets an A1 Output Auditor pass that checks claims against the brief and marks gaps `[NEEDED]`, plus the Company Brain's facts and banned terms (checked by code on the final). People review before anything is used. Model output can still be wrong.
39	How do you prevent runaway cost (LLM10)?	Partial	Only editors and admins can run prompts; 12 model calls a minute per IP and per user; cost is shown on every run and in the Usage tab. An admin-set **monthly dollar cap** blocks Translate and Run once reached (checked before each call, so one run can cross it); blocks are audited. Also set a limit in the Anthropic Console.
40	Do you publish AI safety test results by version?	Partial	Yes, for the prompt library: v8 had 0 safety failures across 20 red-team and 20 adversarial tests. **These come from our own simulated graders**, not an independent lab, and are labeled that way. Live-model re-runs after each model release are planned.

Owner actions before the first enterprise questionnaire is returned

1. Done Nov 11, 2026: the security and support contact is heirlifestyle@gmail.com; security.txt and the disclosure policy are published on the website (/security.html).
2. Approve the drafted incident response plan, disclosure policy and support terms (rows 31–33), and sign the approval line.
3. Decide on cyber/E&O insurance (row 34) and get a lawyer to review the license and any DPA (row 8).
4. Budget an independent penetration test (row 25) and one live SSO test per provider (row 11).

AI Governance Statement and AI System Card

Heir Lifestyle Investments LLC · Intelligent Talk™ Studio v0.2.0 · November 10, 2026 · plain-words edition Not legal advice. Your counsel decides how these rules apply to your own use.

1. AI system card (what this AI system is)

What it does	Turns a person's plain words into structured prompts, runs tested prompts from our library on Anthropic's Claude, and has a second AI pass (the A1 Output Auditor) check the draft
Intended use	Drafting business text (emails, sales pages, summaries, support replies, scripts) that **a person reviews before use**
Not intended for	Decisions about people's jobs, credit, insurance, housing, education, health or legal rights; anything sent or published without human review
Model	Anthropic Claude through *your own* API key. The model and version are named in `config/models.json`, and every run's audit record stores them
Who runs it	You, on your own server. We receive none of your data
Human oversight	Nothing is sent, published, deleted or spent on its own. Messages are drafts until a named approver approves, and nobody can approve their own draft. An admin can stop all AI calls instantly (spending cap = 0)
Known limits	AI can be wrong or invent facts. The auditor reduces this but doesn't remove it. No prompt-injection defense is complete. Our prompt red-team results come from our own graders, not an independent lab
Traceability	Each run logs the model, auditor model, app version and SHA-256 fingerprints of the exact prompt and auditor text, in a hash-chained, tamper-evident audit log
Data	Encrypted at rest (AES-256-GCM, your key). Retention is set by you (default: 90 days for run history). Export and two-person hard delete are built in

2. NIST AI Risk Management Framework (AI RMF 1.0): how we line up

NIST AI RMF has no certificate. This is our own mapping, with evidence you can check.

Function	What NIST asks for	What we have (evidence)
Govern	Policies, roles, accountability, risk culture	Named roles and least-privilege permissions (white paper §4); incident response plan; disclosure policy; support terms; owner accountability. Policies are dated drafts awaiting sign-off
Map	Context, intended use, impacts, limits	This system card; the "not intended for" list; data-flow diagram (white paper §2)
Measure	Testing, evaluation, monitoring	72 automated software tests that gate every release; prompt test suite (normal, missing-info and trick inputs); a release check that fails if any documented number drifts from the code; cost and usage tracking

Function	What NIST asks for	What we have (evidence)
Manage	Respond, recover, improve	Human approval gate; spending cap and kill switch; Safe-Paste Screener; incident plan with a recorded drill; change log per version

Gaps we state openly: no independent evaluation of model outputs; no bias testing, because the Studio drafts text and makes no decisions about people; no formal AI risk register yet (planned).

3. ISO/IEC 42001 (AI management system)

We are not certified. We follow its main ideas where they fit a small company:

- a defined scope (this system card)
- roles and accountability
- risk treatment (the controls in the white paper)
- documented changes (CHANGELOG.md)
- incident handling
- a yearly review

Certification would come later, if buyers require it.

4. EU AI Act: where things stand (checked November 10, 2026)

The AI Omnibus amending the Act was adopted in June 2026 and published in the Official Journal on July 24, 2026.

- **Already in force:**
- prohibited practices and AI literacy (Article 4) since February 2, 2025
- general-purpose AI model rules since August 2, 2025
- general application, including the Article 50 transparency rules, since August 2, 2026
- **Moved later:** high-risk rules for Annex III uses now start **December 2, 2027** (originally August 2, 2026). For Annex I products they start **August 2, 2028**.
- **AI literacy (Article 4):** the duty was softened to supporting the development of AI literacy among staff.

What that means for the Studio:

- **Risk level depends on your use.** Drafting business text is not on the high-risk list. If you used AI output to screen job applicants or score credit, that use could be high-risk, and the Studio is not intended for it (section 1).

- **Transparency (Article 50).** When AI-generated text is published to inform the public on matters of public interest, the deployer must disclose it, unless a person has reviewed it and takes editorial responsibility. The Studio's approval gate makes that human review a recorded step.
- **We are not the model provider.** Anthropic carries the general-purpose AI model obligations for Claude.
- **AI literacy.** Our free plain-words guides and the method kit help your staff understand what AI is and how to check it.

5. Review

This statement is reviewed every 12 months, and whenever the Studio's model, purpose or the law changes. Approved by: ____ (Heir Lifestyle Investments LLC) · Date: _____

Sources for section 4: Future of Privacy Forum, "The AI Act implementation timeline: what changes under the AI Omnibus?" (2026); Cloud Security Alliance research note on the EU AI Act Omnibus delay (2026).

Intelligent Talk™ Studio: Incident Response Plan

Heir Lifestyle Investments LLC · v1.0 DRAFT for owner approval · November 10, 2026 · plain-words edition Not legal advice. Have a lawyer review the notice rules for the states and countries where customers are.

1. What this plan covers

The Studio is **self-hosted**: each customer runs it on their own server, and we receive none of their data. So an "incident" for us is one of these:

1. **A security flaw in our software** (found by us, a customer or a researcher).
2. **Tampering with what we ship**: a changed release file, a fake download, or a stolen signing key.
3. **A compromise of one of our own accounts**: email, Payhip, PayPal, Stripe, Cloudflare, YouTube or the build workspace.
4. **A problem at a customer's site that involves our software**, where they ask us for help.

The customer's own security team runs incidents on their own servers. We support them (section 6).

2. Who does what

Role	Who	Job
Incident lead	The owner (Heir Lifestyle Investments LLC)	Makes the decisions, signs off on every notice
Technical responder	The owner, or a contractor under NDA	Investigates, fixes, ships the patch
Outside help (as needed)	A lawyer; a security firm	Notice rules; forensics

Contact for reports: heirlifestyle@gmail.com (see the Vulnerability Disclosure Policy).

3. How serious is it?

Level	Meaning	Examples	Start work within
S1 Critical	Customer data could be read or changed by someone who shouldn't, or our signed releases can't be trusted	Login bypass; encryption flaw; stolen signing key	4 hours
S2 High	A serious flaw that needs special conditions to use	Privilege escalation that needs an existing account	1 business day

Level	Meaning	Examples	Start work within
S3 Medium	A real flaw with limited impact	A bypassable rate limit; an information leak in an error message	3 business days
S4 Low	A hardening idea, not an active risk	A missing header on a non-sensitive page	Next planned release

4. The six steps

1. **Detect and log.** Write down the time, who reported it, what was seen, and the evidence. Open an incident record with a running timeline.
2. **Contain.** Stop the harm first: - Pull or replace a bad release. - Rotate any key or password that may be exposed. - Sign out sessions and turn off a compromised account. - For a stolen **signing key**: publish a new Ed25519 public key on the website, mark the old one revoked, and re-sign current releases.
3. **Assess.** Answer four questions: what happened, which versions it affects, which customers are affected, and whether any data was exposed. Set the severity level. The Studio's hash-chained audit log (`studio-admin.js audit-verify`) helps a customer check their own server.
4. **Fix.** Write the patch with a test that proves the flaw is closed. The release must pass the full release gate (`engineering/release.sh`) before it ships.
5. **Notify** (section 5).
6. **Review.** Within 10 business days, write a short post-incident review: cause, timeline, fix, and what changes so it can't happen again. Add the lesson to the engineering log.

5. Who we tell, and when

- **Affected customers:** tell them without undue delay. For S1 and S2, the aim is within **72 hours** of confirming they are affected. GDPR gives data controllers 72 hours to report personal-data breaches, and our notice helps them meet that clock. The notice says: what happened, which versions, what to do now, and when the fix ships.
- **All customers:** a security advisory with the fixed version, once a fix exists. For S1 and S2, also email every licensed customer.
- **The reporter:** keep them updated (see the Vulnerability Disclosure Policy).
- **Authorities:** if our own systems held personal data that was exposed (for example, the buyer list on Payhip), follow the breach-notice laws that apply. A lawyer confirms which ones.
- **Payment partners:** if a payment account is compromised, tell Payhip, PayPal or Stripe right away.

6. Helping a customer with an incident on their server

- We answer within the support terms. For a suspected S1 involving our software, we start within 4 hours on business days.
- We never ask for their data or a copy of their data folder. We work from logs they choose to share, the audit-chain results, and version numbers.
- Any remote session happens by screen-share, on their terms, and is logged by them.

7. Practice and upkeep

- Run a tabletop drill once a year (example scenario: "our signing key was stolen"), and after any major change.
- Review this plan every 12 months, or after any S1 or S2 incident.
- Keep an offline copy of the signing-key backup and the account recovery codes (owner step).

Approval

Approved by: ____ (Heir Lifestyle Investments LLC) · Date: _____

Vulnerability Disclosure Policy

Heir Lifestyle Investments LLC · Intelligent Talk™ · v1.0 DRAFT for owner approval · November 10, 2026

We welcome reports from security researchers and customers. If you find a flaw, tell us first so we can fix it before it harms anyone.

What's in scope

- The Intelligent Talk Studio software (all released versions).
- Our website, intelligenttalk.pages.dev, including the free translator.
- The signed release files and our published signing key.

Out of scope:

- Payhip, PayPal, Stripe, Cloudflare, YouTube, Anthropic and OpenAI systems. Report those to the company that runs them.
- Denial-of-service tests, spam and social engineering of people.
- Reports from automated scanners with no proof of a real problem.

How to report

Email heirlifestyle@gmail.com. Please include:

- what you found and where (version or page)
- steps to reproduce it
- what an attacker could do with it
- your name or handle, if you'd like credit

Please don't include anyone's real personal data. If you happen to see some, stop, don't keep it, and tell us.

What we promise

Step	Our target
Confirm we received your report	Within **2 business days**
First assessment and severity	Within **5 business days**
Fix for a Critical flaw	Within **14 days**

Step	Our target
Fix for High	Within **30 days**
Fix for Medium or Low	Within **90 days**

We'll keep you updated, tell you when it's fixed, and credit you in the advisory if you want.

Safe harbor

If you act in good faith under this policy, we will not take legal action against you or ask anyone else to. Good faith means you:

- test only against your own copy of the Studio, or pages that are open to the public
- don't access, change or delete other people's data
- don't harm service for others
- give us a reasonable time to fix the flaw before telling anyone else; we suggest 90 days, or sooner once a fix ships

We can't authorize testing of systems other companies own.

Rewards

We don't run a paid bug bounty today. We say thank you publicly, with your OK.

A machine-readable copy of this contact lives at `https://intelligenttalk.pages.dev/.well-known/security.txt` (RFC 9116). This policy is published at `https://intelligenttalk.pages.dev/security.html`.

Support Terms: Intelligent Talk™ Studio

Heir Lifestyle Investments LLC · v1.0 DRAFT for owner approval · November 10, 2026

What support covers

- Questions about installing, setting up and using the Studio.
- Help understanding a prompt's results.
- Bug reports and security reports. Security reports also follow the Vulnerability Disclosure Policy.
- Updates and security fixes for the version you licensed, for as long as your license or maintenance term runs.

What it doesn't cover

- Your own servers, network, identity provider, or your Anthropic API account and bills.
- Writing custom prompts or doing your work for you. That's available as a separate paid service.
- Results from the AI models. AI can make mistakes; people review before anything is used.

How to reach us

Email: heirlifestyle@gmail.com. Hours: Monday–Friday, 9 AM–5 PM US Eastern, except US federal holidays.

Response targets

Plan	First reply	Period
Personal products (Kit, Library)	Within 2 business days	90 days from purchase
Business, Agency and Enterprise	Within 2 business days; Critical security issues are started within 4 business hours	For the license term

These are response targets. They are not an uptime promise. The Studio runs on your own server, so uptime depends on your hosting. We don't offer an uptime SLA (service level agreement) today.

Severity, for software problems

- **Critical:** a security flaw or data at risk. Handled under the Incident Response Plan.
- **High:** the Studio won't start, or a main feature fails for everyone, and there's no workaround.
- **Normal:** a feature misbehaves but there's a workaround.

- **Question:** how-to and setup help.

Remote help

Done only by screen-share, on your terms. We never ask for your data, your data folder, your keys or your passwords.

Changes

We'll give 30 days' notice before reducing support for a current license.