



SECURITY WHITEPAPER, VERSION 1.0

Security and key management

October 12, 2026 | routexor.com/security | ATERNA AI, Denver, Colorado

ROUTEXOR routes AI model calls on our customers' own provider keys. That one design choice sets most of our security posture. We hold no prepaid balance and resell no tokens. We do hold encrypted provider keys, and our servers can decrypt them to make the calls you ask for.

This paper covers how those keys are protected, how spend caps behave when something fails, what we record and what we don't, and what we have not built yet. Every statement was checked against the production code on October 12, 2026. Anything planned is labeled as planned.

1. What happens to one request

- Your agent calls api.routexor.com over HTTPS with a ROUTEXOR key (sk-at-...).
- We hash the key with SHA-256 and look up the digest. Unknown, disabled or known-compromised keys get a 401, and the account must be active.
- The key's rate limit (60 requests a minute by default) and the plan's daily limit are checked. If the store that holds these limits is unreachable, the request is refused with a 503 and never sent to a provider.
- The spend preflight sets an explicit output ceiling, prices the worst case and reserves that amount atomically against the key's monthly cap and the task profile's daily cap.
- We decrypt your provider key for that call, write a vault audit record and call the provider over TLS. Redirects are refused. Google keys travel in a header, never in a URL.
- We record usage metadata and settle the reservation against actual usage.
- If a provider fails before any output, the request can move to the next model in your task profile's fallback chain. A provider rejecting your key returns a 502 to your agent, so agent harnesses fail over to their next model instead of locking you out. Once output has started, we do not switch providers.

2. The provider key vault

- Envelope encryption. Each stored provider key gets its own random 256-bit data key and is encrypted with AES-256-GCM, which also detects tampering. That data key is encrypted with a key-encryption key derived for your account (PBKDF2-SHA512, 100,000 iterations, from a master secret and your account ID).
- The master secret lives in our hosting platform's environment settings, not in code or the database. The API refuses to start if it is missing or shorter than 32 characters.
- Server-side decryption. The API decrypts a provider key only to make a call you requested or a key check you started. This is not end-to-end encryption, and we do not offer keys that only you can decrypt.

- Every decryption writes a vault audit record: account, provider, key ID, IP address and user agent. Each account can read its own vault log through the dashboard API.
- Optional IP allowlist per provider key, checked before decryption. Denials are logged. It matches exact addresses only.
- More than 1,000 decryptions of one key in a day raises an anomaly flag in the log. It flags; it does not revoke.
- Only the last four characters of a key are ever shown back. You can rotate or delete a provider key in the dashboard at any time, and deleting it stops routing on it immediately.
- Your keys only. Since July 28, 2026, customer traffic never falls back to a platform key. With no key for a provider, the call fails with a clear error.

3. ROUTEXOR keys, sign-in and support access

- ROUTEXOR API keys are shown once, at creation, and stored only as a SHA-256 digest plus a short display prefix. A list of known-compromised digests is blocked at authentication.
- Each key carries its own rate limit and an optional monthly dollar cap. A cap of \$0 blocks all spend.
- Dashboard passwords are hashed with bcrypt (cost 12). Sign-in attempts are throttled per account, 10 per 15 minutes.
- Sessions are httpOnly cookies backed by a signed token that expires in 7 days. Every request re-checks account state and revocation. Logout, password change and "sign out everywhere" revoke existing sessions.
- Browser changes require an exact trusted Origin. Pages send a strict Content-Security-Policy with a fresh nonce per response, refuse to be framed, and use HSTS.
- Support access: an administrator can open a read-only view of an account for at most 15 minutes. It cannot create credentials, change the account or make paid calls, and each start and stop is written to an admin audit log.

4. Spend caps that fail closed

- Enforced today: a per-call output ceiling, a monthly dollar cap per ROUTEXOR key and a daily dollar cap per task profile. All are checked before the call is sent.
- The worst-case price of each call is reserved atomically before dispatch, so a burst of parallel calls cannot all slip past a cap that only one of them fits under. The price envelope includes long-context, cache-write and peak-rate pricing. Reservations settle to actual usage afterward.
- Fail closed: invalid limits, corrupt counters, missing reservation state or an unreachable limit store refuse the request instead of letting it through.
- Limits of this design: the reservation is based on catalog prices and a token estimate, so it is not a guarantee about your provider invoice. Provider-side tools, image billing or a provider ignoring an output limit can cost more. Counters live in Redis with 40-day retention and are not a durable financial ledger. Keep spending limits at your providers as well.
- Not live: cumulative budgets per agent run or job. That backend is built and deployed with new admissions switched off until it has been tested with design partners. A job_id on a request is metadata only.

5. What we record, and what we don't

Recorded for every routed call: timestamp, model and provider, token counts (including cache and reasoning tokens), latency, estimated provider cost, status code, task profile, which of your provider keys was used and the outcome category. Routing decisions are recorded too.

Not recorded by ordinary routing: the text of your prompts and responses. We process them in memory to deliver the result.

- Evaluation datasets you create store the messages and expected answers you upload.
- Outcome feedback stores whatever you put in its metadata.
- Ensemble sends your prompt to several models, and the judge model also sees their answers.
- RUDY, the dashboard analyst, sends your question and a compact usage summary to a model on your own key.

Errors: provider error bodies are discarded; we keep a status and a category. Application logs use fixed messages and allowlisted error codes, and request logs record the route pattern rather than the path you called.

Browser diagnostics carry fixed categories only, with no error text, stack traces or URLs. They are capped at 500 reports and kept for at most 30 days.

There are no advertising pixels on routexor.com, including the signup and key-entry pages. We do not sell personal data or use prompts or outputs to train models.

6. Infrastructure and transport

- The web app runs on Vercel. The API, PostgreSQL and Redis run on Railway. Billing runs through Polar as merchant of record, so card numbers never reach us.
- HTTPS everywhere, with HSTS on routexor.com and api.routexor.com. Database connections outside the platform's private network require verified TLS (certificate and hostname), and settings that would skip verification are rejected.
- The API runs as a non-root Node 24 container. Schema changes ship as reviewed migrations applied before the new version starts, with lock and statement timeouts.
- A scheduled availability check runs from GitHub Actions, and the API records its own self-check every five minutes for routexor.com/status.

7. How changes ship

Every change to the main branch runs through CI, and a release needs it to pass:

- secret scanning of the full Git history with a pinned Gitleaks release;
- dependency advisory gates for the web app and the API, each recording a software bill of materials;
- a scan of the actual API container image with Trivy, whose binary provenance is verified before it runs;
- type checks, unit tests and integration tests against fresh PostgreSQL and Redis, plus a boot test of the real image with its migrations.

CI actions are pinned to reviewed commit IDs. A green CI run is release evidence, not a certification.

8. Third parties that touch data

SERVICE	WHAT IT DOES	DATA IT SEES
Vercel	Hosts the web app and dashboard	Web requests, session cookies in transit
Railway	Hosts the API, PostgreSQL and Redis	Everything ROUTEXOR stores, including encrypted provider keys
Polar	Checkout and billing, merchant of record	Billing identity and payment details (never sent to us)
AI providers you choose	Run your model calls on your own keys	Your prompts and outputs, under their own terms and retention
GitHub	Source code and CI	Code only, no customer data
Cloudflare Turnstile	Signup bot check	Not active yet; would see signup browser signals

Model metadata (names, context sizes, list prices) is refreshed every 60 seconds from a public model catalog. It contains no customer data.

9. What we don't have yet

- No SOC 2 report, ISO 27001 certificate or third-party penetration test. Nothing on this page has been audited by an outside firm.
- The master secret is an environment variable, not a cloud KMS or hardware security module, and there is no automated master-key rotation. Rotating it today means re-encrypting every stored key.
- Our servers can decrypt your provider keys. There are no customer-managed keys and no bring-your-own-KMS option.
- No multi-factor authentication, SSO or SAML for dashboard accounts, and no self-service email verification or password recovery yet.
- Bot protection on signup is built but not switched on.
- IP allowlists match exact addresses only, and trusted client-IP provenance behind every proxy is not established.
- Cumulative per-run budgets are built and switched off. Caps are not a guarantee about your provider invoice.
- Status history starts October 12, 2026. There is no uptime SLA, no 24/7 on-call rotation and no published recovery objectives (RPO and RTO) or tested restore drill.
- One hosting region, with no data-residency choice.
- Account deletion and data export happen by request, not self-service, and older logs and backups follow their own retention.
- The vault audit log is readable through the API but not yet shown in the dashboard.

10. SOC 2 roadmap: targets, not achievements

Today ROUTEXOR has no SOC 2 report. The dates below are targets. They assume our pre-seed round closes by the end of 2026; its use of funds includes \$80,000 for the SOC 2 Type 1 audit and a security review.

TARGET DATE	STEP	WHAT IT COVERS
Target: Q1 2027	Readiness	Choose a compliance platform. Write the core policies: access, change management, incident response, vendor management, data retention. Add MFA for staff and admin access. Move the master secret into a managed KMS. Run a first external penetration test.
Target: Q1 2027	Auditor selection	Engage a licensed CPA firm. Fix the scope: the Security criterion across the API, dashboard, key vault and the infrastructure behind them.
Target: Q2 2027	SOC 2 Type 1	Point-in-time audit of control design. Report shared with customers under NDA.
Target: Q3 2027 to Q1 2028	SOC 2 Type 2 window	At least six months of controls operating under observation, with the Type 2 report targeted for Q1 2028.

We will update this page as each step happens, with the date it happened.

11. Reporting a security issue

Email team@aterna.ai with "Security" in the subject line. Include steps to reproduce. Please don't include live credentials or customer data.

Version 1.0, October 12, 2026: first publication.

This document describes ROUTEXOR as built on October 12, 2026. It is not an audit report or a certification. The current version lives at routexor.com/security; live measurements are at routexor.com/status.