

Security at Revent AI Lab

Encrypted by design, isolated per tenant, least-privilege by default — and honest about what is shipped versus what we are still building. Here's how your data is protected, checked against the platform's own code.

AES-256-GCM at rest

Least-privilege access

Per-tenant + per-user isolation

We train no models on your data

Enforced CSP + HSTS

Card data never touches us

THE SIX PILLARS



Envelope encryption at rest

Every sensitive value — OAuth tokens, Gmail/Calendar content, founder docs, retrieval chunks, attachments — is sealed with authenticated **AES-256-GCM** before it reaches the database. A master key (KEK) wraps a fresh per-workspace data key; plaintext keys are zeroed in memory after use. **One tradeoff we name rather than hide:** a search index of the words in your content is stored unencrypted beside the ciphertext, so keyword search works at all.

crypto/primitives.ts · envelope.ts · ciphertext bytea columns · tamper + wrong-key tests



Per-user, per-workspace isolation

The tenant boundary is re-derived from the signed server session on every request — never trusted from the client — and enforced in SQL. CEO Brain is a private brain: every read, write and retrieval is scoped by **both workspace and user**. No cross-tenant, no cross-user recall.

requireWorkspaceMember() at 200+ call sites · eq(workspaceId) AND eq(userId) on every query · isolation regression test



Least privilege, grant by grant

Each agent connects on its **own separate grant** and asks only for what its job needs, so no agent inherits another's access. Read-only agents are only ever given read access. The two that send on your behalf — Procurement and Sales — carry that permission alone, and a person approves before anything goes out. Connect is admin-gated and CSRF-protected.

google-oauth/scopes.ts — 5 connection purposes, each with its own scope set · signed state + single-use nonce



Prompt-injection-aware AI

On the paths that take in third-party content (product feedback, brand notes, briefs) that content is fenced and explicitly marked as **data, not instructions** before the model sees it, so nobody can smuggle commands inside something they send you. **Not yet on every agent** — extending it is active work, and we would rather say so. Inbound webhooks are HMAC-verified with timing-safe checks and replay windows.

fenceSafe() · [FEEDBACK_DATA] delimiters · slack/verify.ts · stripe webhook constructEvent



Hardened in the browser & on the wire

HTTPS-only with **HSTS**. A **Content-Security-Policy** is **enforced** on every page and minted fresh per request with its own nonce — what stops an injected script running at all. Framing denied, nosniff, Referrer- and Permissions-Policy locked down. TLS terminates at the host; DB connections require **sslmode=require**.

proxy.ts mints the per-request policy · platform/security/csp.ts · violations reported to /api/csp-report



Payments never touch us

Card data is handled entirely by **Stripe** Checkout and Customer Portal — a card number never reaches the platform, keeping PCI scope minimal. Inbound billing webhooks are signature-verified and idempotently de-duplicated.

no PAN handling in src · Stripe-hosted flows · constructEvent + ON CONFLICT dedup

HOW YOUR DATA FLOWS

1

Connect

One purpose-scoped grant, admin-gated



2

Ingest

Only in-scope data, on your behalf



3

Encrypt

Sealed with your workspace key



4

Isolate

Scoped by workspace + user



5

Process

Task-scoped to the listed providers



6

Erase

Self-serve · key destroyed

Straight answers, before you connect a thing.

Where does my data physically live?

Your workspace data sits in managed **Neon Postgres** with TLS-required connections and provider point-in-time recovery; AI processing happens with the providers listed overleaf. Redis is ephemeral (background jobs only) — no durable customer data. The app runs behind an Nginx / Let's-Encrypt host. The authoritative statement of **where each of those sits**, and the safeguards covering any transfer between them, is in our privacy policy at ailab.revent.store/privacy — residency is stated in one place so the two cannot drift apart.

Is data encrypted in transit and at rest?

In transit: HTTPS-only + HSTS, and DB connections require `sslmode=require`. **At rest:** AES-256-GCM authenticated encryption with per-workspace envelope keys for all sensitive fields. One documented tradeoff: full-text search lexemes from brain chunks are stored queryable (strictly less than the source text) for keyword recall.

How is my private brain isolated from other tenants?

Workspace & user are derived from the signed server session on every request and applied as SQL WHERE predicates; CEO Brain is additionally scoped by user, so neither another tenant nor another user in your workspace can read it. **Stated plainly:** today that isolation is enforced in the application layer; a database-level backstop is on our roadmap as defence in depth.

What OAuth permissions do you request?

Only what a given agent needs, on its own grant. Most agents are read-only: CEO Brain takes `calendar.readonly` + `gmail.readonly`, Customer Copilot `gmail.readonly`, the analytics agents `webmasters.readonly` + `analytics.readonly` + `drive.file`. Two send on your behalf because that is their job — Procurement (`gmail.send`) and Sales (`gmail.send` + `calendar`) — and ask for it separately and explicitly; nothing is sent without a person approving it. You see the exact scope on the consent screen before granting, and can revoke it from your Google account at any time.

Is my data used to train your AI models?

We train no model of our own on your data, store no weights and run no fine-tuning on it. For the providers we send content to we claim only what their terms say: **Anthropic** is contractually prohibited from training on what it processes for us and deletes it within 30 days by default; **Voyage** is configured so your content is not used for training and not retained after processing. Our speech-to-text and Arabic-language providers run under their standard commercial terms, which we are confirming in writing — until that is done we will not make the same promise on their behalf.

What happens when I disconnect — can I delete everything?

Yes, and you can do it yourself from your workspace settings without asking us. Deleting a workspace destroys that workspace's encryption key — so the stored data becomes cryptographically unrecoverable — and removes the records. We also call Google's server-side revoke endpoint for your tokens, but that call runs after the deletion and is **not retried** if Google is unreachable, so the certain way to end access is to remove it from your own Google account too. Encrypted backups then age out on their retention schedule — which we would rather state plainly than imply is instant.

How do you defend against prompt injection & app-sec risks?

On the paths that take in third-party content, that content is fenced and labelled **data-not-instructions** before the model sees it, so a message someone sends you cannot issue commands to your agents — though that fencing is **not yet on every agent**, and extending it is active work. Inputs and env are Zod-validated; server actions are auth-gated; session cookies are `httpOnly` + `Secure` + `SameSite`; an **enforced per-request Content-Security-Policy** with its own nonce is the main browser-side control. Every pull request is scanned for vulnerable dependencies and leaked secrets, with a full-history sweep weekly, and the scanner binaries are version-pinned and checksum-verified so a compromised release cannot be swapped in.

Are you SOC 2 / ISO 27001 certified? Is the Google app verified?

Not yet, and we would rather say so than imply otherwise — no SOC 2, no ISO 27001, no pen-test attestation. We're a security-by-design build, pre-certification: the controls in this brief are real and in the code today, but they have not been audited by an independent third party. Google app verification and the associated CASA security

Who touches your data, and what they get.

SERVICE	PURPOSE	DATA IT RECEIVES
Anthropic (Claude)	LLM reasoning for agent tasks — no training on your data	The specific content sent for a task (excerpts, prompts), task-scoped
Voyage AI	Text embeddings for CEO Brain semantic retrieval	Plaintext text chunks, to generate embedding vectors
Groq	Speech-to-text transcription	Audio you submit for transcription
Hugging Face · Cohere Labs	Arabic-language reasoning, via a hosted Arabic model	The prompt content for an Arabic task
Google APIs	Source for the data you connect	Scope-limited access to what you grant; tokens & content stored encrypted
Neon (Postgres)	Managed primary database, TLS-required, PITR	All app data — sensitive fields stored as AES-256-GCM ciphertext
Stripe	Billing via hosted Checkout + Portal	Card data (handled entirely by Stripe — never touches us) + subscription metadata
Resend	Transactional email delivery	Recipient address + message content for that email
Cloudflare R2	Storage for media you publish or upload	The image and media files themselves
Sentry	Error monitoring (server & edge only)	Server-side error context; personal-data capture off, no browser tracking or session recording
GitHub	Turning your in-product feedback into fixes	Feedback you submit in the product, written into our private issue tracker
DigitalOcean	Application hosting	Technical connection data such as IP addresses — not your business content
Slack / Teams (optional)	Opt-in chat — the bot posts notifications & takes commands	Bot-scoped channel/user data + messages with the bot; tokens encrypted
Fireflies (optional)	Opt-in meeting-notes ingestion via your API key	Transcripts you ingest; API key stored encrypted, never returned in plaintext
Google Ads (optional)	Opt-in ad-account management	Access to the ad account you connect; nothing is spent without your approval
LinkedIn (optional)	Opt-in publishing of approved posts	The post text and media you approve for publication
Meta (optional)	Opt-in publishing to Facebook & Instagram	The post text and media you approve for publication
X (optional)	Opt-in publishing of approved posts	The post text and media you approve for publication
WhatsApp (optional)	Opt-in messaging on a number you connect	Messages sent and received on that number

Where we are, stated plainly.

● SHIPPED (IN CODE)

AES-256-GCM envelope encryption, key per workspace

Per-user + per-workspace isolation from the signed session

Purpose-separated, least-privilege connections

Enforced per-request CSP (nonce-based) + HSTS + hardened headers

Self-serve deletion with cryptographic erasure

Google token revoke called on erase (best-effort)

Dependency + secret scanning on every change, weekly full sweep

HMAC-verified webhooks (Slack, Stripe); Stripe-hosted payments

Prompt-injection fencing (data-not-instructions)

Server-only error monitoring, PII capture off

SHA-pinned CI (typecheck · lint · test · build)

● OPERATIONAL (CONFIG)

TLS in transit (Let's Encrypt at host)

DB sslmode=require

Key material held in the host environment

Rate limiting on authentication

Managed Postgres with point-in-time recovery

Backup retention on a fixed schedule

● IN PROGRESS

Independent certification. We do not hold SOC 2 or ISO 27001 today. The controls at left are real and in the code; they have not been third-party audited, and we will not imply otherwise.

Google app verification and the associated security assessment.

Continued hardening — key management, database-level defence in depth, and automated retention tooling.

Evaluating us seriously? We keep a detailed engineering ledger — what is shipped, what is configuration, what is still open, with the evidence behind each line. Ask and we share it under NDA.