

Jents - Security & Trust Overview

Last updated: 2026-08-11 · Security contact: security@jents.io

Jents is the governance, cost, and control layer for an organization's AI agents. Because that job means handling sensitive operational data, security and tenant isolation are designed into our foundation, not bolted on. This document explains how we protect your data. We're an early-stage company building toward formal certification (see **Compliance**), and we're glad to complete your security questionnaire, sign an NDA/DPA, and walk your team through any of this.

1. Tenant isolation - your data is never visible to another customer

This is our most important guarantee, enforced at three independent layers:

- **Every record is stamped with your organization ID.** All application data carries an `organizationId`, and every database query is automatically constrained to the signed-in company through a single, centralized data-access layer - code cannot issue an un-scoped query for company-scoped data.
- **Database-level backstop (defense-in-depth).** Row-Level Security is **enabled on every table** in the database, with per-company policies on the core company-scoped tables, proven in CI against a restricted role that cannot bypass them. Policy coverage is still being extended across the remaining tables. To be precise about what this does and does not do today: the application connects with an administrative role that Postgres exempts from those policies, so RLS is a *configured* backstop rather than the layer currently doing the enforcing, and moving the application onto the restricted role is in progress. The primary, verified guarantee remains the centralized query scoping and CI enforcement described here.
- **Gateway isolation** (for routed AI traffic): each customer's vendor credentials are registered as a private, per-organization model on the gateway, and each agent's key may only call *that organization's* models. One customer's keys cannot reach another's.

We verify this automatically. Our continuous-integration pipeline includes (a) a build-blocking guardrail that fails any change introducing a query that isn't company-scoped, and (b) automated cross-tenant tests that assert one company cannot read, update, or delete another's records. Isolation cannot silently regress between releases.

2. Your API keys and secrets

- **Bring-Your-Own-Key (BYOK).** Your LLM vendor API keys (OpenAI, Anthropic, Google) are held only on the metering gateway, **encrypted at rest**. They are **never stored in, logged by, or returned to** the Jents application - we keep only a non-secret reference (e.g. the model name).
 - **Secrets we do store** are encrypted at rest with **AES-256-GCM** and never returned to the browser. These are credentials for things you choose to connect: a Slack bot token, a cloud provider's access secret or service-account key (AWS, Azure, Google Cloud), a vendor org-admin key used to read your own usage and seat counts, and the admin key for your own gateway if you bring one.
 - Administrative/gateway master credentials are server-side only and never exposed to the browser or to customers.
-

3. What data we process and store

By default, Jents stores only metadata about your AI calls - not the content of your prompts or responses. We practice strict data minimization:

- **Stored in Jents:** your agent inventory and metadata; per-call **cost, token counts, model, vendor, latency, and status** (the metering data); the budgets, alerts, and subscriptions you configure; your user directory (names, emails, roles) from your identity provider.
- **Your content (prompts & responses):** passes through the gateway to your chosen vendor and is **not retained** by Jents by default (see optional trace capture below).
- **Tracing:** tracing is held **entirely within Jents** - no third-party tracing provider receives your prompt or response text. By default a trace records metadata only - model, token counts, cost, latency, status.
- **Optional trace capture (off by default):** a team may enable trace capture, which stores **redacted, length-capped** prompt, response, tool-call, and error text for *that team's* calls, to debug and explain failures. It is **per-team and disabled unless enabled**; secrets are stripped before write and entries are **automatically deleted after 30 days**.

Connected data sources (ROI measurement). To measure the business impact of your agents, you may connect systems such as your CRM, support desk, data warehouse, product analytics, billing, or project tools - always **read-only** and **scoped to only the metrics you choose**. Jents reads these in two ways, and in neither does it read the **body of a record**.

- **Aggregate reads.** Most reads are a single aggregate - a count, sum, or average over a time window - computed **on the source's side**; the records never leave your system. Where a source can't aggregate server-side, Jents reads only the specific field it needs (a number, a timestamp, a status) to compute the total, and **stores only the resulting number**.
- **Record metadata.** So a chart can be broken down by team, status, or owner, Jents also stores **one metadata row per record** (a pull request, issue, ticket, or deal): its identifier, dates, state, labels, and **the person it is assigned to** - which depending on the system is a display name, a work email, or an internal user ID. Some sources also supply the record's **title** (a Jira summary, a Linear issue title, a deal name) so a chart can label a bar.
- **What is never read or stored:** the contents of a record - descriptions, comments, message and ticket text, code, and your own free-text custom fields. This exclusion is enforced in the connector code itself, not left to configuration.
- Every read runs through a single **audited path** with row-count caps, and is **logged to your audit trail** (what was read, when, how many rows it touched).
- The credential you connect is **encrypted at rest** (AES-256-GCM); connections are **revocable at any time**.
- Jents stores the **measurements** (the numbers over time), your metric configuration, and the record metadata above - **never your records' contents**.

We do not sell your data, and we do not use it to train models.

4. Encryption

- **In transit:** all traffic is encrypted with **TLS (HTTPS)** - application, gateway, and database connections.

- **At rest:** the database is encrypted at rest by our infrastructure provider; the specific secrets we hold are additionally application-encrypted (AES-256-GCM).

5. Authentication & access control

- **Authentication** is handled by **WorkOS**, a dedicated SOC 2-certified identity provider - this supports **enterprise SSO / SAML and SCIM directory sync** for your team.
- **Role-based access:** Owner / Admin / Member. Sensitive actions (managing budgets, connecting vendors, provisioning keys, changing alert destinations, billing records) are **restricted to admins**; regular members cannot perform them.
- Sessions use signed, encrypted cookies. We never handle or store your users' passwords - the identity provider does.

6. Infrastructure & subprocessors

Jents is built on established, **SOC 2-certified** infrastructure. Current subprocessors:

Provider	Purpose
Vercel	Application hosting (SOC 2 Type II)
Supabase	Primary database - PostgreSQL, encrypted at rest (SOC 2)
WorkOS	Authentication / SSO / directory (SOC 2 Type II)
Railway	Metering gateway hosting
Sentry	Error monitoring / diagnostics - secrets stripped before send
PostHog	Product analytics (US) - identifies users by work email
Stripe	Payment processing (PCI DSS Level 1)
OpenAI / Anthropic / Google	LLM inference - via <i>your</i> BYOK keys
AWS / Google Cloud / Microsoft Azure	Model inference and agent inventory - only if you connect your own cloud account
Slack / email	Alert delivery - only if you connect it

Data residency: Jents' application data is hosted in the **AWS US East (N. Virginia) region** (via Supabase). If your organization requires data to remain in a specific region (e.g. EU), we can discuss this for enterprise engagements. A current subprocessor list is available on request, and we give notice of material changes.

7. How a call flows (data path)

1. Your agent calls the **Jents gateway** using a per-agent key you provision.
 2. The gateway forwards the request to your chosen vendor **using your own (BYOK) key**.
 3. The gateway records **cost + token usage** (never your secret keys) and sends that metering data to Jents, attributed to the right agent and company.
 4. You see cost, attribution, budgets, and alerts in the Jents dashboard - scoped to your company only.
-

8. Availability, backups & data ownership

- **Live system status:** real-time service availability is published at <https://stats.uptimerobot.com/HEDlujdMfi>.
 - Managed PostgreSQL on encrypted, cloud-redundant infrastructure, with **automated daily backups** (7-day retention).
 - **Your data is yours.** We retain metering data for the duration of your engagement. On request, or at offboarding, we **delete your organization and everything attached to it** (the delete cascades across every table). That deletion is performed by us rather than self-serve today, so email security@jents.io and we will confirm in writing when it is done. We can accommodate a specific retention window on request.
-

9. Compliance posture (honest, early-stage)

- Jents is **not yet independently SOC 2 / ISO 27001 certified** - we're an early-stage company, and we'd rather be transparent than imply otherwise. A **SOC 2 Type II readiness program is actively underway**.
- **Documented security program.** We maintain a written set of information-security policies - access control, authentication, change management, secure development, vulnerability management, incident response, encryption, logging & monitoring, business continuity, data retention/deletion, and vendor management - that govern how we operate.
- **Self-assessment available on request:** a CSA CAIQ (Consensus Assessments Initiative Questionnaire) documenting our cloud-security controls.
- **Payments:** all card processing is handled entirely by **Stripe (PCI DSS Level 1)**; Jents never sees or stores card data (PCI SAQ-A scope).
- We follow **GDPR- and CCPA-aligned** data-handling principles and will sign a **DPA on request**. Our DPA covers the EU/UK GDPR, US state privacy laws, and the Israeli Privacy Protection Regulations.
- **Published legal documents:** [Terms of Service](#), [Privacy Policy](#), and the [Service Agreement](#) governing use of the Solution.
- We build on **SOC 2-certified infrastructure** (Vercel, Supabase, WorkOS) and apply aligned practices: encryption, least-privilege access, strict tenant isolation, and dependency hygiene.

10. Secure development & monitoring

- **A tenant-isolation guardrail and automated cross-tenant tests run in CI on every change** (see §1)
 - isolation is continuously verified, not assumed.
- **Dependency & secret scanning:** automated dependency-vulnerability alerts and fixes run continuously; secret scanning (gitleaks) and a dependency audit run on every pull request, on merge to production, and weekly.
- **Hardened response headers:** every response carries HSTS, anti-clickjacking, MIME-sniffing protection, and related browser-security headers. Our Content-Security-Policy is deployed in **report-only** mode while we validate the allow-list, so it reports violations rather than blocking them.
- **Audit logging:** administrative and data-access actions are recorded to an audit trail (who did what, when).
- **Error & uptime monitoring** run continuously with alerting to our team (see §8).
- Dependencies are pinned to vetted, security-patched versions (the gateway is held to a current hardened release line).
- Machine-to-machine endpoints (the metering webhook and integrations) authenticate with signed / secret-verified requests and reject unauthorized or malformed input.

11. Deployment options

- **Standard:** multi-tenant SaaS with the isolation guarantees in §1.
- **Enterprise:** for organizations with stricter data-residency or isolation requirements, we're happy to discuss a dedicated / isolated deployment - contact us.

12. Reporting a vulnerability / contact

We welcome responsible disclosure. Please email **security@jents.io** with any security concern; we'll acknowledge promptly and keep you updated. For questionnaires, a DPA, or a walkthrough with your security team, reach out to the same address.

This overview describes Jents' security posture as of the date above and is provided for evaluation. It is not a contractual commitment except where incorporated into a signed agreement. © Jents.