# Zenforge

## Security Overview

Architecture, Controls & Data Handling

March 20, 2026

Classification: Public

# Contents

# 1 Introduction

Zenforge is a cloud-native SaaS platform that uses large language models (LLMs) to help teams understand and act on unstructured information. This document describes Zenforge's security architecture, controls, and data handling practices for prospective customers conducting vendor security reviews.

## 1.1 Company Overview

| | |
|---|---|
| **Product** | **Zenforge — stakeholder management platform** |
| **Hosting** | Fly.io, Sydney region (Australia) |
| **Primary Database** | Neon PostgreSQL, Sydney region |
| **SOC 2 Status** | Pursuing SOC 2 Type I |

# 2 Architecture

## 2.1 System Overview

Zenforge is a server-rendered web application with an API-first design. The system consists of:

- **Web application:** Go backend with server-rendered HTML
- **Database:** PostgreSQL (Neon) — single primary data store for all application data
- **Job queue:** PostgreSQL-native job queue for pipeline orchestration
- **LLM integration:** API calls to Anthropic, OpenAI, or Google Vertex AI to extract facts from uploaded text
- **Identity:** Kinde (OIDC provider) for authentication
- **Observability:** OpenTelemetry + Langfuse for LLM trace monitoring

## 2.2 Data Flow

1. User authenticates via OIDC (Kinde) and receives a JWT
2. User submits signal content through the web interface or API
3. Signal text is stored in PostgreSQL (Sydney, AU)
4. A processing pipeline sends signal text to the configured LLM provider to extract facts
5. Extracted structured data (entities, facts, interpretations) is stored in PostgreSQL
6. LLM providers delete API inputs after their retention period (see below)

**Customer data leaves the Zenforge application boundary in two ways:** (1) uploaded text is sent to the configured LLM provider for extraction — the provider processes it, returns the result, and then deletes it according to their retention policy (see below); (2) full prompt and completion content is sent to Langfuse for diagnostics. LLM providers do not use API-submitted data for model training.

## 2.3 LLM-Specific Security Concerns

| | |
|---|---|
| **Does customer data train models?** | **No. All three supported LLM providers (Anthropic, OpenAI, Google) prohibit using API-submitted data for model training under their current terms of service.** |
| **What data is sent to LLM providers?** | Signal text and workspace context (names, settings), wrapped in extraction prompts. No user credentials or authentication tokens are included in LLM API calls. |
| **Provider data retention?** | Anthropic: 7 days. OpenAI: 30 days. Google: up to 90 days for flagged abuse monitoring (exception available on request). See Subprocessor List for details. |
| **Zero data retention available?** | All three providers offer zero-retention agreements, and Zenforge plans to activate these. Default provider retention currently applies: Anthropic 7 days, OpenAI 30 days, Google up to 90 days. |

| | |
|---|---|
| **Data residency?** | Application data stored in Sydney, AU. LLM processing occurs in the US (provider API endpoints). Langfuse trace data stored in the EU (Langfuse Cloud). No customer data is persisted by LLM providers beyond their stated retention windows. |
| **Prompt/completion logging?** | Full prompt and completion content is sent to Langfuse for diagnostics. Langfuse data is covered by the same workspace data erasure process as the primary database — when a workspace is deleted, its Langfuse traces are purged. |

# 3 Authentication & Authorization

## 3.1 Authentication

Zenforge uses OpenID Connect (OIDC) for all user authentication via Kinde. Users authenticate with Google, Microsoft, or GitHub accounts. The architecture supports any OIDC-compliant provider via standard discovery and JWKS validation.

- **Token validation:** JWT signature, issuer, and audience verified on every request using JWKS discovery
- **Signing algorithms:** Industry-standard RSA, ECDSA, and EdDSA algorithms supported
- **Email verification:** Enforced on all provisioning paths (unverified emails are rejected with HTTP 403)
- **Token transport:** Bearer token in the Authorization header, or an HttpOnly cookie for browser sessions
- **Session cookies:** HttpOnly, SameSite=Lax, Secure (on HTTPS)
- **OAuth state parameter:** 16-byte cryptographically random value for CSRF protection
- **User provisioning:** Just-in-time on first login, with concurrent-login race condition handling

## 3.2 Authorization

Every workspace-scoped endpoint verifies the authenticated user's membership in the target workspace. This is enforced in middleware — there is no code path that bypasses this check.

- **Role-based access control (RBAC):** Four hierarchical roles (owner > admin > editor > viewer) with least-privilege defaults
- **Workspace isolation:** All database queries are scoped to the requesting workspace; cross-workspace data access is structurally impossible
- **Owner-only operations:** Destructive operations (e.g., workspace deletion) require the owner role

# 4 Network & Transport Security

## 4.1 HTTPS

All traffic is served over HTTPS. Fly.io's forced-HTTPS configuration redirects all HTTP requests to HTTPS.

## 4.2 Security Headers

All responses include the following security headers:

| Header | Value |
|---|---|
| `Strict-Transport-Security` | `max-age=31536000; includeSubDomains` |
| `X-Frame-Options` | `DENY` |
| `X-Content-Type-Options` | `nosniff` |
| `Referrer-Policy` | `strict-origin-when-cross-origin` |
| `Permissions-Policy` | `geolocation=(), microphone=(), camera=()` |
| `Content-Security-Policy` | `default-src 'self'; script-src 'self' 'nonce-...' 'unsafe-eval' https://cdn.jsdelivr.net; style-src 'self' 'unsafe-inline'; connect-src 'self' https://cdn.jsdelivr.net; frame-ancestors 'none'` |

**Inline script nonces.** Inline scripts use per-request cryptographic nonces rather than `unsafe-inline`. A fresh 128-bit random nonce is generated for each HTTP response and injected into both the CSP header and the corresponding `<script>` tags. Scripts without the correct nonce are blocked by the browser.

**`unsafe-eval` allowance.** The Datastar front-end framework requires the `unsafe-eval` Content Security Policy header to evaluate its data-binding expressions via `Function()`. This is a framework constraint with no alternative mode available (see Datastar security reference). The risk is mitigated: all expressions are defined at compile time in server-side templates, user-supplied content is never interpolated into executable attributes (templ auto-escapes into text nodes and value attributes only), and the per-request nonce requirement independently blocks injection of new scripts.

Authenticated responses additionally include no-cache, no-store, and must-revalidate cache directives to prevent caching of sensitive data.

## 4.3 Rate Limiting

API requests are rate-limited to **200 requests per minute per IP address** using a sliding window algorithm.

# 5 Application Security

- **Input validation:** All API requests and responses validated against the OpenAPI specification
- **No SQL injection:** All database queries are pre-defined and parameterized — user input cannot alter query structure
- **Server-rendered:** Server-rendered pages with minimal client-side JavaScript, reducing client-side attack surface
- **Static analysis:** Strict linter configuration with zero issues tolerated
- **Vulnerability scanning:** Go vulnerability database scanner (govulncheck) runs on every build, detecting known vulnerabilities in dependencies and standard library
- **CI/CD gating:** All changes pass code generation, format check, lint, and test before deployment. Only the main branch can deploy.
- **Dependency management:** Pinned dependency versions
- **No secrets in source:** All secrets provided via environment variables

# 6 Infrastructure Security

## 6.1 Container Hardening

- **Minimal base image:** Alpine Linux (minimal attack surface)
- **Multi-stage build:** Build tools are not included in the runtime image
- **Non-root execution:** Application runs as an unprivileged user
- **Static binary:** No C library dependencies

## 6.2 Deployment

| | |
|---|---|
| **Platform** | **Fly.io** |
| **Region** | Sydney, Australia |
| **HTTPS** | Forced (all HTTP redirected to HTTPS) |
| **Health monitoring** | Automated health checks with restart on failure |

# 7 Data Security

## 7.1 Data at Rest

All application data is stored in Neon PostgreSQL (Sydney, AU). Neon provides:

- Encryption at rest (managed by Neon)
- Point-in-time recovery (currently 6 hours; configurable up to 30 days)
- Automated backups

## 7.2 Data in Transit

All data in transit is encrypted via TLS:

- Client ↔ Fly.io: TLS terminated at Fly.io edge
- Fly.io ↔ Application: WireGuard-encrypted internal network (managed by Fly.io)
- Application ↔ Neon: TLS-encrypted PostgreSQL connection
- Application ↔ LLM providers: HTTPS API calls

## 7.3 Data Erasure

Zenforge supports complete workspace data deletion:

- All workspace data is removed atomically from the primary database in a single transaction
- Associated LLM traces are purged from Langfuse
- In-flight processing jobs are cancelled
- LLM provider data auto-deletes per their retention policies (no application-side action required)
- Deletion events are audit-logged

## 7.4 Data Classification

| Category | Examples | Storage |
|---|---|---|
| User identity | Email, name, OIDC subject | PostgreSQL + Kinde |
| Workspace metadata | Workspace name, settings | PostgreSQL |
| Signal content | Customer-submitted text | PostgreSQL |
| Extracted intelligence | Entities, facts, interpretations | PostgreSQL |
| LLM traces | Prompts, completions, timing, token counts | Langfuse (EU) |
| Application logs | Request logs, error logs | Fly.io (ephemeral) |

# 8 Observability & Logging

- **LLM tracing:** Full prompt and completion content is sent to Langfuse (EU) for diagnostics and quality monitoring. Traces are included in workspace data erasure.
- **Request logging:** Structured logging with URL truncation to prevent customer content leaking into application logs
- **PII protection:** User identity details (email, OIDC subject) are not included in application logs

# 9 Incident Response

A formal incident response plan is being developed as part of the SOC 2 readiness program. It will cover detection, triage, containment, customer notification, and post-incident review.

# 10 MVSP Self-Assessment

Self-assessment against the Minimum Viable Secure Product checklist (mvsp.dev), an industry baseline from Google, Okta, Salesforce, and Slack.

### 10.0.1 Business Controls

| ID | Control | Status | Notes |
|---|---|---|---|
| 1.1 | Vulnerability disclosure | Planned | security@zenforge.ai contact established. Formal VDP to be published. |
| 1.2 | Customer testing | Planned | Will enable customer-initiated security testing on request. |
| 1.3 | Self-assessment | Pass | This document. CAIQ v4 self-assessment also in progress. |
| 1.4 | External testing | Planned | Penetration test to be contracted with a third-party firm. |
| 1.5 | Training | Planned | Will implement role-specific security training as the team scales. |
| 1.6 | Compliance | In progress | SOC 2 Type I via Openlane. CAIQ v4 for CSA STAR Level 1. |
| 1.7 | Incident handling | Planned | Formal incident response plan in development as part of SOC 2 readiness. |
| 1.8 | Data handling | Pass | Neon provides encryption at rest. No unencrypted production data on local media. |

### 10.0.2 Application Design Controls

| ID | Control | Status | Notes |
|---|---|---|---|
| 2.1 | Single Sign-On | Pass | OIDC via Kinde (Google, Microsoft, GitHub). No additional cost. No password-based auth. |
| 2.2 | HTTPS-only | Pass | Forced HTTPS, HSTS with 1-year max-age, Secure cookie flag. |
| 2.3 | Security headers | Pass | CSP, X-Frame-Options: DENY, no-cache on authenticated responses. See §4.2. |
| 2.4 | Password policy | N/A | No passwords. Authentication is entirely SSO via OIDC. |

| ID | Control | Status | Notes |
|---|---|---|---|
| 2.5 | Security libraries | Pass | Go standard library + maintained frameworks. All database queries parameterized via code generation. |
| 2.6 | Dependency patching | Pass | Go modules with pinned versions. govulncheck scans for known vulnerabilities on every build. |
| 2.7 | Logging | Partial | Auth events and mutations logged with user ID and timestamp. 30-day retention via Fly.io Grafana. Audit logging for deletions. CRUD-level audit trail not yet comprehensive. |
| 2.8 | Encryption | Pass | TLS for all data in transit. Encryption at rest via Neon. See §7. |

### 10.0.3 Application Implementation Controls

| ID | Control | Status | Notes |
|---|---|---|---|
| 3.1 | List of data | Pass | Data classification table maintained. See §7.5. |
| 3.2 | Data flow diagram | Pass | Data flow documented in §2.2. Subprocessor list maintained separately. |
| 3.3 | Vulnerability prevention | Pass | No SQL injection (sqlc), CSRF protection (OAuth state + SameSite), XSS mitigation (server-rendered, security headers), authorization on every endpoint. |
| 3.4 | Time to fix vulnerabilities | Planned | No formal SLA yet. Will establish 90-day remediation target. |
| 3.5 | Build and release | Pass | Git version control, CI/CD pipeline, secrets stored separately from source code. |

### 10.0.4 Operational Controls

| ID | Control | Status | Notes |
|---|---|---|---|
| 4.1 | Physical access | N/A | Managed infrastructure (Fly.io, Neon). No physical datacenters operated. |
| 4.2 | Logical access | Partial | MFA delegated to OIDC provider (Kinde). Workspace membership enforced. Formal access review process not yet established. |

| ID | Control | Status | Notes |
|---|---|---|---|
| 4.3 | Sub-processors | Pass | Subprocessor list maintained with data categories, retention, and DPA status. See companion document. |
| 4.4 | Backup and DR | Partial | Neon provides point-in-time recovery (currently 6 hours; configurable up to 30 days). Application is stateless (redeployable from CI/CD). Formal DR plan not yet documented or tested. |

**Summary:** 13 Pass, 3 Partial, 6 Planned, 1 In Progress, 2 N/A. Remaining items are formal processes (VDP, incident response, DR testing, access review) being addressed as part of SOC 2 readiness.

# 11 Compliance Roadmap

| | |
|---|---|
| **CAIQ v4 Self-Assessment** | **In progress. Will be published to CSA STAR Registry (Level 1).** |
| **SOC 2 Type I** | In progress. |
| **Penetration Test** | Planned. Will engage a credible third-party firm. |

# 12 Contact

For security inquiries, vulnerability reports, or to request additional documentation:

- **Email:** security@zenforge.ai