AI-Powered Due Diligence for Deal Teams — Security Whitepaper

Is AI Safe for M&A Due Diligence?

What to check before uploading confidential deal documents to an AI due diligence tool: training-data use, PII handling, encryption key ownership, tenant isolation, access control, and audit logging.

PUBLISHED BY
Anweshna
AI Due Diligence Platform

VERSION & DATE
v1.0
August 2026

Contents

Executive Summary

Deal documents are among the most sensitive material a company handles — unreleased financials, litigation strategy, customer lists, the fact that a transaction is happening at all. The instinct to be cautious about putting that through any external system, AI or otherwise, is correct.

This whitepaper states plainly what to check before uploading anything to an AI due diligence tool, using Anweshna's own architecture as the working example throughout. It covers training-data use, PII handling, encryption key ownership, tenant isolation below the application layer, role-based access control, and immutable audit logging.

Key assurances: Zero model training on client data (contractual, not configurable) · PII stripped before inference · Per-client HKDF encryption keys · Dual-layer tenant isolation (application + Postgres RLS) · Granular RBAC with Clean Team boundaries · Append-only audit trail with provenance watermarks.

Is Your Data Used to Train the Model?

This is the threshold question, and the answer depends entirely on which interface is used. Consumer chat interfaces frequently reserve the right to use conversation content for training unless a user opts out. Enterprise API terms are structured differently: inputs and outputs are typically excluded from training by contract, not merely by a settings toggle a client has to remember to set.

Anweshna processes documents through Anthropic's enterprise API, under terms that prohibit using client inputs to train models. This is a contractual guarantee, not a configurable preference — there is no setting to accidentally leave on.

What Reaches the Model

The strongest privacy control is not sending sensitive data to the model at all. Before any extracted text reaches the AI, Anweshna strips personally identifiable information — Social Security numbers, card numbers, email addresses, phone numbers — from the document. The risk-relevant content survives; the identifiers that would let a document be tied to a specific individual do not.

This matters independently of the training question, because it also limits exposure in the event of any downstream logging or debugging on the model provider's side, which is outside any client's direct control.

Encryption and Key Ownership

“Encrypted at rest” is table stakes and tells you little on its own — the question that matters is whether every client shares a key or each has its own. A shared key means a single compromise exposes every client’s data; a per-client key limits a compromise to one engagement.

Anweshna derives a unique encryption key for every client using HKDF key derivation, salted with the client’s own identifier, and encrypts extracted document text and analysis findings with Fernet symmetric encryption under that key. A breach of one client’s key material does not expose another’s data, because there is no shared key to compromise.

Isolation Below the Application Layer

Most multi-tenant systems isolate clients through application logic — a query filtered by `WHERE client_id = ...`. That works until a bug in that logic ships, and application-layer filtering is exactly the kind of code that accumulates bugs as a system grows. The question worth asking any vendor is what the second, independent layer is.

> **Anweshna enforces isolation twice, independently.** Application-layer filtering is the first layer. Postgres row-level security is the second: the database itself refuses to return another client’s rows to a restricted, non-superuser application role, regardless of what the application code asks for. A bug in the first layer does not become a cross-client data exposure, because the second layer does not depend on the first being correct.

Who Can See What

Role-based access should map to how a deal team actually works, not to a flat all-or-nothing permission. Anweshna’s roles — Owner, Analyst, Viewer, and Clean Team — control upload, analysis and download rights separately, and deal rooms can be placed in a restricted mode that limits access to a walled clean team, relevant where the buyer is a competitor and commercially sensitive material needs to stay outside the broader deal team’s view.

The Audit Trail

Every access, download and change should be attributable after the fact — not as a compliance afterthought, but because a deal team needs to be able to answer “who touched this document and when” during and after the process. Anweshna logs every such action to an append-only audit trail, and every report download carries a provenance watermark identifying who downloaded it, their role, and when.

Certifications & Attestations

We are actively undergoing formal certification processes to provide independent validation of our security posture.

- **SOC 2 Type I:** Readiness assessment in progress. Target completion: Q1 2027.
- **ISO 27001 & ISO 42001:** Information Security and AI Management Systems implementation in progress.
- **Penetration Testing:** We mandate regular, third-party penetration testing of our application and infrastructure. Contact us for our latest letter of attestation.

Data Residency & Subprocessors

We process data exclusively in top-tier cloud regions with strict service level agreements.

- **Data Residency:** Database hosted on Supabase (Asia-Pacific). Application hosted on FastAPIcloud. EU/UK isolation is available for Enterprise plans.
- **Breach Notification SLA:** We commit to notifying affected customers within 48 hours of discovering a confirmed security breach, as enshrined in our standard DPA.

Authorised Subprocessors

Subprocessor	Service	Location
Anthropic	LLM Inference (Zero Data Retention)	USA
Supabase	Database & Auth (Row-Level Security)	Asia-Pacific
FastAPIcloud	Application Hosting	USA
Paddle	Payment Processing	UK / USA

Implemented Controls Mapping

We align our security practices with the 15 core principles required for institutional due diligence vendors:

- **Data Minimization:** PII scrubbed pre-inference.
- **Tenant Isolation:** Row-Level Security enforced at the database layer.
- **Encryption:** HKDF per-client key derivation; AES-256 at rest, TLS 1.3 in transit.
- **Access Control:** Granular RBAC including exclusive “Clean Team” boundaries.
- **Model Sovereignty:** Contractual zero-training guarantees via Anthropic API.
- **Provenance Tracking:** Cryptographic document watermarking and append-only audit logs.

Seven Questions to Ask Any AI Due Diligence Vendor

1. **Is my data used to train your models, or the underlying model provider's?** Ask for the contractual basis, not just a verbal assurance.
2. **Are personal identifiers removed before text reaches the model?**
3. **Is data encrypted at rest under a key unique to my account, or a key shared across all clients?**
4. **Is tenant isolation enforced below the application layer?** If the answer is only "we filter by client ID in our queries," that is one layer, not two.
5. **What is the retention and deletion policy, and is deletion actually irreversible?**
6. **Is every access logged immutably, and can I get that log?**
7. **Can access be restricted by role within my own team**, including a walled configuration for competitively sensitive deals?

A vendor with confident, specific answers to all seven has thought about this properly. A vendor whose answer to any of them is "we will look into that" has not, regardless of how the product otherwise looks.

Frequently Asked Questions

Is it safe to upload confidential M&A documents to an AI due diligence tool?

It depends on the specific system, not on AI as a category. Check whether documents are used to train models (contractually, not just by setting), whether personal identifiers are stripped before text reaches the model, whether encryption uses a key unique to your account, whether tenant isolation is enforced below the application layer, and whether access is logged immutably. A vendor with clear, specific answers to all of these has built the system properly.

Are documents used to train AI models when using Anweshna?

No. Anweshna processes documents through Anthropic's enterprise API, under terms that contractually prohibit using client inputs to train models. This is a contractual guarantee built into the processing agreement, not a setting that has to be configured or remembered.

How does per-client encryption actually protect data?

Anweshna derives a unique encryption key for every client using HKDF key derivation, and encrypts that client's extracted document text and findings with Fernet symmetric encryption under that key. Because there is no shared key across clients, a compromise of one client's key material does not expose any other client's data.

What does row-level security add beyond normal application permissions?

Most systems isolate tenants only through application code — a query filtered by client ID. That protection fails if a bug in that code ships. Row-level security enforces isolation a second time, independently, inside the database itself: a restricted database role cannot retrieve another client's rows regardless of what the application code requests, so a single application-layer bug does not become a cross-client data exposure.

Anweshna scores documents for risk signals. It does not replace legal review, independent verification, or expert due diligence. Thresholds cited here are general industry and supervisory reference points, not legal advice — verify current requirements with the relevant regulator.

Contact: anweshna.com **Terms:** anweshna.com/terms.html **Privacy:** anweshna.com/privacy.html