How to Protect Contract Confidentiality When Using AI: The Moderation Layer Approach

Every time you paste contract text into an AI tool, you hand confidential client data to a third party. Learn why Text Redaction Layers solve this, and what to look for when evaluating legal AI vendors.

Published by Contextwise · Last updated February 2026 · 15 min read

In This Guide

1. The Confidentiality Problem with Legal AI
2. What a Moderation Layer Actually Does
3. How It Works: Detection, Pseudonymization, and De-pseudonymization
4. Regulatory Requirements Mapping
5. Vendor Evaluation Checklist for Legal Teams
6. How Contextwise Does It
7. Frequently Asked Questions

The Confidentiality Problem with Legal AI

A standard commercial contract can alter the risk allocation of a transaction's balance sheet. A single legal agreement or NDA purchase agreement might contain the names of every party and their signatories, the exact dollar amounts, jurisdiction specific IP assignment terms, indemnification triggers, and liquidated damages specified by compliance language. That is exactly the kind of data you should never hand to a third party.

Most enterprise contracts contain at least one such sensitive clause. Every time a lawyer pasts GPT-4, Gemini, or any LLM generated review tool into an intermediary privacy layer. The full text, every sensitive data point included, travels to the LLM provider's servers. These things are strong breach risks.

Training data absorption. Since 2.5k providers can begin to learn and improve their models using customer inputs, ChatGPT's Free tier explicitly states that conversations may be used to train and improve their systems. Even your consumer language where a training dataset, it cannot be recalled. In March 2023, Samsung engineers pasted proprietary source code into ChatGPT, a mere incident training transcripts from ChatGPT's access data requests facilitates within a 20-day window. Samsung responded by banning ChatGPT and developing its own internal transcription and translation, contractual safeguard, access to the cloud.

Platform breaches and logs. In the same month as the Samsung incident, OpenAI disclosed a bug that allowed some ChatGPT users to see titles from other users' conversation and, for a small group of ChatGPT Plus subscribers, payment-related information including names, email addresses, and credit card expiration dates. Even when providers do not train on your data, retention logs and support access to those conversations often remain before logging and internal data tools. Anthropic's terms permit access to engagement limits are not sufficient. Even beyond the LLM provider, the broader AI tool ecosystem introduces additional exposure. Browser extensions, third-party wrappers, and plugins frequently request broad permissions to read and write every conversation. The European CNIL designates confidentiality as a "fundamental and primary right and one of the most vital rules in data protection.

The bottom line: using AI for contract work is not optional going forward. But doing it without a privacy architecture is indefensible both ethically and commercially.

What a Moderation Layer Actually Does

A moderation layer is a processing step that sits between the user and the language model. Its entire purpose is to ensure that the LLM never sees confidential data while still delivering useful, contextualized analysis back to the user.

The process has three stages: detect, replace, and restore.

Detection. The system identifies sensitive data using a combination of pattern matching (dates, monetary amounts, formats), dictionary lookups (known client names, internal codenames), and contextual analysis (determining which proper nouns are sensitive based on surrounding text). A regulatory sentence mentioning "Section 4.2 Apple's Delivery of Goods" is not sensitive. The same sentence with "Section 4.2 Acme Corp's Delivery of Goods" is. Context matters, and the detection layer must understand it.

Replacement. Identified sensitive data is swapped for consistent placeholders. "Acme Corp" becomes "PERSON_1", "$2,500,000" becomes "MONEY_1", "March 15, 2024" becomes "DATE_1". The mapping is stored in RAM, keyed to the session, and never persisted to disk.

Restoration. When the LLM returns its analysis, the system reverses the substitution using a security-aware mapping table, reinserting all original values into the output.

The critical word here is **pseudonymization**, not redaction. This distinction matters as breaks the approach. Redact a date, and the LLM can't reason about contract timelines. Pseudonymize it, and it still knows it's dealing with a date and can reason about it correctly.

If you replace "Acme Corp" with "[redacted]", you destroy the relational context the LLM needs to reason about the contract. The model cannot tell which party has which obligation if every reference becomes an identical redacted token. Pseudonymization, the replacement with consistent placeholders, preserves the structure and relationships that make the analysis useful.

The LLM never sees the real data. The system sees the documents but cannot read them without the mapping. Even if a breach exposes the LLM's logs, the attacker sees "PERSON_1 agreement terminates on DATE_1." The structural and semantic relationships are intact. Enough for the LLM to reason its analysis, the mapping table wraps the pseudonyms back. The user sees a complete, fully contextualized result. The LLM never sees what that result.

Let's look at what this actually looks like in a real contract.

RAW CLAUSE SENT TO LLM (NOT RECOMMENDED)

Section 3.1 Indemnification. Pfizer Inc. shall indemnify, defend, and hold harmless Acme Corp, its officers, directors, employees, and affiliates, including Sarah Chen, VP Legal, from and against any claims, arising from the relationship, Pfizer's breach of this Agreement, or Pfizer's gross negligence, up to $1,500,000. Notice of any claim must be sent to legal@acme-corp.com within 30 days.

ANONYMIZED CLAUSE SENT TO LLM (WITH MODERATION LAYER)

Section 3.1 Indemnification. PARTY_1 shall indemnify, defend, and hold harmless PARTY_2, its officers, directors, employees, and affiliates, including PERSON_1, VP Legal, from and against any claims, arising from the relationship, Pfizer's breach of this Agreement, or Pfizer's gross negligence, up to MONEY_1. Notice of any claim must be sent to EMAIL_1 within DURATION_1.

The LLM can analyze the anonymized version with the same accuracy as the raw version. It can identify the indemnification structure, flag the liability cap in one directional, note the notice requirement, and suggest standard revisions. The analysis is identical in quality. The only difference is that the parties are, from model's memory, by codes, not by name.

How It Works: Detection, Pseudonymization, and De-pseudonymization

A production-grade moderation layer's core single technique. It combines three detection methods, each covering different category of sensitive data. Here's how each works, with common specific examples.

Layer 1: Named Entity Recognition (NER)

NER is a machine learning technique trained to identify named entities within unstructured text: person names, organizations, specific geography locations, dates, and similar categories. Modern systems, a trained NER model reading the sentence "Pursuant to Section 4.2, Apple Delivery of Goods" and recognizing "Apple" as an organization and "Section 4.2" as a location. Transformer-based NER models, such as those in Google's Cloud Natural Language API, Hugging Face's BERT, and spaCy's NER pipeline, achieve F1 scores of 0.92 or higher on standard benchmarks.

NER DETECTION OUTPUT

Acme Corp → PERSON (recognized with PERSON_1)
Pfizer Inc. → ORGANIZATION (recognized with ORG_1)
Sarah Chen → PERSON (recognized with PERSON_2)
Pfizer's Delivery → LOCATION (recognized with DOMAIN_1)
March 15, 2024 → DATE (recognized with DATE_1)

NER handles the broadest category of sensitive data: names and entities embedded in natural sentence structure. It's the foundation layer for most production systems. But NER alone misses structured PII such as dollar amounts, phone numbers, and specific dates in non-standard formats. Pattern matching fills the gap. Regular expressions and rule-based systems excel at catching these structured patterns. Dollar amounts, email addresses, phone numbers, social security numbers, and jurisdiction references that generic models often miss.

Layer 2: Regular Expressions (Regex)

Regex handles the structured, predictable data that NER models sometimes overlook. Common use cases for LLM contract review: contract value ranges (monetary amounts, percentages), specific dates, phone numbers, Social Security numbers, EIN/TIN numbers, bank account numbers, and IP addresses.

REGEX DETECTION PATTERNS IN CONTRACTS

$1,500,000.00 → matches: \$[0-9,]+(\.[0-9]{2})?
legal@acme-corp.com → matches: email pattern, EMAIL_1
March 15, 2024 → matches: date pattern, DATE_1
87-1234567 → matches: EIN pattern, TIN_1
205.168.1.100 → matches: IPv4 pattern, IP_ADDR_1

Regex is deterministic and fast. It's perfectly reliable for finding every instance of a defined pattern. The limitation is that it only works on data with predictable formats. A company name like "Meridian Technologies" has no universal format, which is why NER handles that category.

Layer 3: Custom Dictionaries and Rules

Every organization has data specifically sensitive to their business. Internal codenames, client lists, specific deal terms, internal project codenames, third-party vendor names, and IP codenames. Generic tools don't recognize these. A custom dictionary lets you define exactly what to treat as sensitive, regardless of whether it matches any pattern or any named entity.

CUSTOM DICTIONARY DETECTION EXAMPLES

Project Aquarium → custom rule (internal codename) → PROJECT_1
Meridian Technologies → custom rule (client list) → ORG_1
Acme Corp's Delivery → custom rule (partner list) → ORG_2

WHY ALL THREE LAYERS ARE NECESSARY

Regex catches structured formatting. NER handles names and contextual entities but can miss formatted data. Custom dictionaries cover what both miss: the specific terms that matter to your company. Current distributed PII tags process in each organization. A mature pipeline runs all three in series, each layer catching what the previous layer missed. The layered approach is what separates a production system from a proof of concept.

The Mapping Table: Data Structure Underpinning Everything

Often dismissed, the moderation layer's hidden complexity. Little that pass-through original entity with its placeholder. This table is the key to reversibility.

SESSION MAPPING STRUCTURE (NEVER PERSISTED TO DISK)

PARTY_1 → Pfizer Inc. (ORGANIZATION)
PARTY_2 → Acme Corp (ORGANIZATION)
PERSON_1 → Sarah Chen
MONEY_1 → $1,500,000
DATE_1 → 31 Dec 2025
EMAIL_1 → legal@acme-corp.com
PROJECT_1 → Project Aquarium

Consistency is the mapping's critical. If "Meridian Technologies" appears 47 times across the document, it's MONEY_1 every time. The LLM can track relationships across the document because the mapping ensures identical inputs produce identical placeholders. After the LLM responds, the system reverses the substitution: PARTY_1 AGRAVATE_1, etc. the de-anonymization step reconstitutes the mapping table and swaps every placeholder back to its original value. The user sees a complete result. The LLM never saw any of it.

The mapping table itself must be stored securely within the user's environment or encrypted on the vendor's servers, and it must never be persisted to disk alongside the anonymized text or the LLM provider. If it exists, the entire exercise would be pointless.

Let's see the full cycle in a complete example.

ORIGINAL CLAUSE SENT TO LLM (RAW DATA IN MODE)

Section 12.3 Non-Compete. For a period of TWENTY-FOUR (24) MONTHS following the termination of this Agreement, Jonathan Reed shall not, directly or indirectly, engage in the business that competes with Nexus Industrial Group within the United States, including but not limited to, providing consulting services to any manufacturer of industrial adhesives. Compensation shall be $12,500/month.

ANONYMIZED CLAUSE SENT TO LLM (SAFE)

Section 12.3 Non-Compete. For a period of PERIOD_1 following the Termination Date, PERSON_1 shall not, directly or indirectly, engage in the business that competes with ORG_1 within the TERRITORY_1, including but not limited to, providing consulting services to any manufacturer of industrial adhesives. Compensation shall be AMOUNT_1/month.

The LLM can analyze this clause fully. It can flag the non-compete duration, evaluate the compensation structure, and suggest standard revisions. The analysis is identical in quality to the version with real data. Every word of the substance driving the real analysis, the risk allocation, the scope, the compensation, the jurisdiction, is intact.

After the LLM returns its clause fully, it can flag the non-compete duration, evaluate the compensation structure, and suggest standard revisions. The analysis is identical in quality to the version with real data. Every word of the substance driving the real analysis, the risk allocation, the scope, the compensation, the jurisdiction, is intact.

Two cases to be aware of. Ambiguous pronouns, Pfizer, "herein" is unique to "or" in the third-party. Rare, well-trained models handle it. Complex nested structures, a clause referring to "the Party's obligations in Section 3 hereof." Both are edge cases. Test your specific document types before deploying. Most contracts, because the vocabulary is formulaic, the models never hesitate. Low contract risk. Second, "Will the confidentiality" between these two legal teams, or the confidentiality of the Parties, Pfizer's, Acme's, or both? Ambiguity, challenges in the United Kingdom, where, "Every corporate entity is a separate legal entity." Well-trained models handle it.

Comparison: Four Approaches to Contract Data Protection

Not all approaches to AI privacy are equal. Here's how the four most common configurations stack up across the criteria that matter to legal teams.

Approach	Data leaves your network?	Training data risk	Audit trail	Suitable for regulated industries
Raw API access	Yes, full text sent to LLM provider	Depends on provider's terms. Many offer opt-out for API users. Consumer tiers default to training.	None built-in. Logs depend on provider's retention policy.	Only with enterprise agreements and additional safeguards. Most compliance teams will raise objections.
DPA (data processing agreement)	Yes, but under contractual terms	Lower than raw API access, but still depends on provider's internal policies and subprocessor chain.	Varies by provider. Some offer usage logs on request.	Acceptable for many GDPR contexts. Insufficient for HIPAA without BAA.
Text redaction layer	No. Only anonymized text leaves your network.	None. The LLM never sees real data.	Full audit trail of what was redacted and when.	Strong fit for GDPR, HIPAA, and most regulatory frameworks.
Local LLM	No. Text never leaves your infrastructure.	None. Model runs on your hardware.	Depends on your logging infrastructure.	Strongest option. Highest infrastructure cost and complexity.

Vendor Evaluation Checklist for Legal Teams

If you are a CC, CLO, or legal ops leader evaluating AI tools for contract work, these are the questions that separate serious vendors from those impersonating a privacy.

- **Does the tool anonymize contract text before it reaches the LLM?** Ask for a demonstration. The right answer is a clear, specific description of what gets replaced and what stays intact. If the vendor cannot show you what the LLM actually receives, that is a red flag.
- **Where does anonymization happen? Client side, server side, or hybrid?** Client side is the most secure: the sensitive data never leaves your network at all. Server side means the vendor handles our data first, but still before it reaches the LLM provider. Hybrid adds complexity but can be acceptable if the architecture is clear and well documented.
- **What detection techniques does the moderation layer use?** Look for NER, regex, and custom dictionary support. Ask how it handles ambiguous cases. Is it rule-based or does it use a trained model? Can you add custom dictionaries for your specific codenames, internal project names, and vendor lists?
- **Can your clients tune the detection for our specific data?** Generic PII detection misses your internal codenames and deal terms. The ability to add custom dictionaries and rules is what separates a production system from a proof of concept.
- **Is the mapping table stored persistently or ephemerally?** The right answer is ephemeral: in memory, never written to disk, destroyed when the session ends. If the vendor stores mapping tables, ask where, how long, and who can access them.
- **Do you offer a DPA (data processing agreement) and what does it cover?** For any tool processing EU data, a DPA is mandatory under GDPR. Review the subprocessor list and the data retention terms. Some DPAs allow the provider to retain data for "service improvement" by default.
- **What happens when the LLM provider changes its terms of service?** Providers update their terms frequently. The tool you evaluate today may train on your data next quarter if the provider changes its policy. Ask how the vendor monitors for changes and what happens to your data if the terms become unacceptable.

How Contextwise Does It

Contextwise implements every layer described in this guide. The detection layer combines NER, regex, and custom dictionaries, with a correlation layer. Scanning for sensitive data is not enough. Finding "$1,500,000" in a contract means nothing if the system doesn't know that it's a liquidated damages clause. The correlation layer connects detections to contract structure, so the system understands what it's looking at and why it matters.

Step 1: Document ingestion. The uploaded contract text is sent to Contextwise's secure processing environment. The document is parsed into sections and clauses, preserving the structure needed for clause-level reasoning. Anonymization runs on the parsed text, not on raw bytes.

Step 2: Entity detection and correlation. Three layers run in parallel: NER for names and entities, regex for structured patterns, and custom dictionaries for client-specific terms. The correlation layer then groups related detections and assigns consistent placeholders. Multiple mentions of the same entity receive the same placeholder across the document.

Step 3: Secure LLM collaboration. The anonymized text is sent to Contextwise's secure LLM processing. The model analyzes clauses, identifies risks, and generates recommendations. The LLM never sees real names, financial amounts, or specific dates. The prompts are optimized for clarity, so the model reasons about the contract structure and obligations without needing the sensitive data.

Step 4: De-anonymization and delivery. Contextwise's moderation layer reversing the mapping table and restores every placeholder substitution. The user sees a fully contextualized analysis, identical in quality to a direct LLM interaction, with all names, amounts, and dates restored.

Three features make this approach especially relevant for enterprise legal teams:

Custom configuration matching your workflow. Add custom terms for client codenames, internal project names, specific vendor lists, or any sensitive pattern, or not defined. The system automatically recognizes standard PII elements and financial formatting, but administrators can add organization-specific terms, contract-specific codenames, and organization-specific identifiers. This ensures the detection layer catches what generic tools miss, the internal terms that matter to your specific business.

Consistent audit trail. Every AI interaction is logged with full detail: what was triggered the analysis, what data was shared, which AI provider received the request. The complete record of documents we stored responds to a secure, complete audit trail. This is not just about compliance documentation. It's about being able to answer the question "what exactly did we share with third-party AI?" with specificity, not guesses.

SCENARIO: Contract analysis process. Before any data leaves the user's environment, users can see the exact anonymized version of their document. The preview shows what the LLM will receive, so the moderation layer is working and a quality assurance step it's sensitive terms are actually the anonymized characters, the user sees what is actually sent to the cloud.

NEXT: PROMPT MANAGEMENT AND PROMPT OPTIMIZATION

Once the legal layer is secure, a separate engineering problem emerges with applications. That means our customers need clauses that are consistent in the output, so that the system can reliably extract clauses, compare versions, and flag deviations. Prompt management and prompt optimization are a different layer than the moderation layer. Contextwise's prompt management system, Prompt Studio, lets non-technical users iterate on prompts, test them against a golden set, and deploy new versions without a developer in the loop. This is a separate system, and it's worth understanding how it works and why it matters for legal teams.

Frequently Asked Questions

Why is it risky to paste contracts directly into ChatGPT or other LLMs?

Three specific risks. Training data absorption: free tier ChatGPT conversations may be used to improve OpenAI's models. Platform breaches: bugs have exposed chat titles and, for a small group of ChatGPT Plus subscribers, payment-related information including names and addresses. Retention logs: even when providers do not train on your data, retention logs and support access to those conversations often remain. Samsung's internal ban after employees pasted proprietary source code into ChatGPT is a real-world example of the third risk.

What is a moderation layer in legal AI, and how does it protect contract data?

A moderation layer is a processing step between the user and the LLM. It scans contract text using NER, regex, and custom dictionaries to identify sensitive data, replaces it with consistent placeholders (PERSON_1, MONEY_1, DATE_1), sends the anonymized text to the LLM, and reverses the substitution after the model responds. The LLM never sees real names, financial amounts, or specific dates. The user sees a complete, fully contextualized result. The LLM never sees what that result means.

What does the AIDA act about using AI with confidential client information?

Alfred Arnold, Explorer CC Ltd, 2026 recommends lawyers to keep client information confidential when using AI tools, and to avoid uploading sensitive data to third-party AI systems. The Law Society of Ireland has specifically warned that uploading full contracts into AI systems risks breaches of confidentiality and client consent. Anthony Arnold,