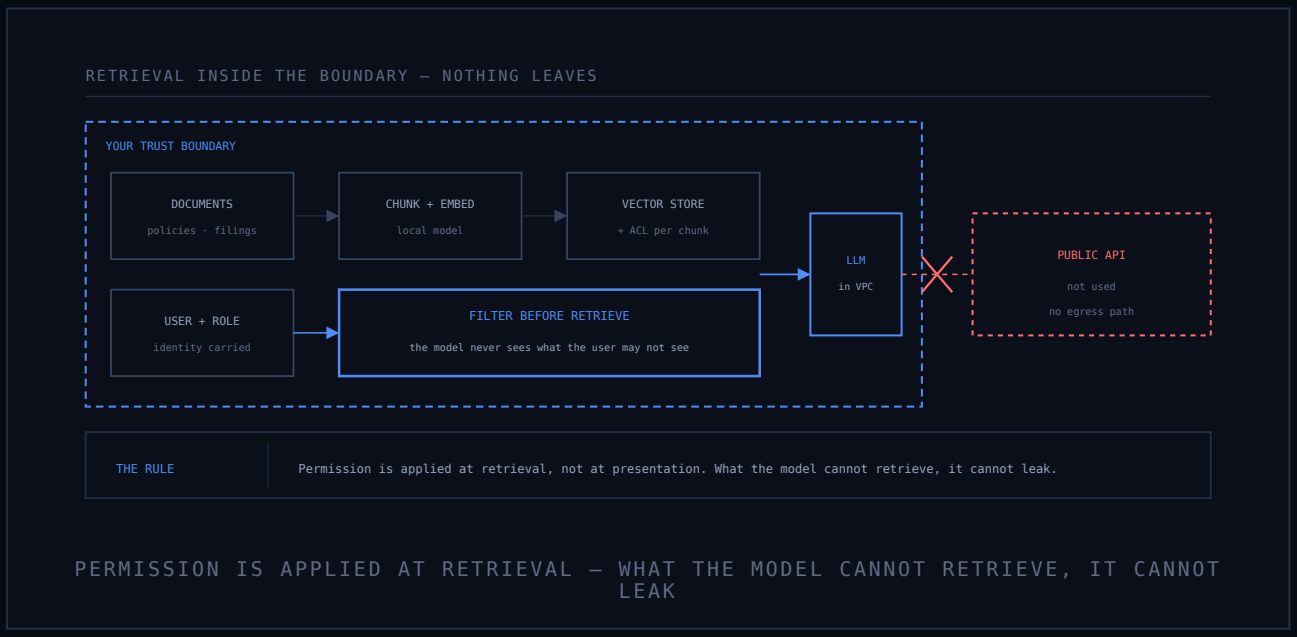
The Model Answered Confidently. Nobody Could Say Where It Got That.

A bank's most useful knowledge sits in its own documents — credit policy, procedure manuals, contracts, prior decisions, the accumulated record of how this institution actually operates. It is exactly the corpus that would make a language model useful, and exactly the corpus that cannot be uploaded to one.

Retrieval-augmented generation is the standard answer, and the standard implementation quietly does the thing you were trying to avoid. Documents are chunked and embedded, frequently by a hosted service. Chunks are retrieved and placed in a prompt, sent to a hosted model. **The architecture that was supposed to keep your data inside has been shipping it out at every stage, in pieces, which feels different and is not.**

We build the other version: retrieval and inference inside the institution's own boundary, permission applied at retrieval rather than to the answer, and every claim traceable to the passage that produced it — with an enforced ability to refuse, because a model that cannot say "I do not know" will say something else instead.

WHAT PRIVATE ACTUALLY MEANS



THE FIVE PROPERTIES

01

Nothing crosses the boundary

Embedding, retrieval and inference run inside the institution's own environment. There is no egress path to a public endpoint, because there is no configuration in which one would be used.

02

Permission at retrieval

Access control is evaluated before the search, not applied to the answer. The model is never given context the user could not have opened themselves.

03

Every claim cited

Answers carry the passages that produced them. A reviewer follows a claim back to the paragraph rather than taking the model's word for it.

04

Refusal is a valid answer

When retrieval returns nothing adequate, the system says so. This must be built deliberately, because the default behaviour is to answer anyway.

05

The query log is sensitive too

What people ask reveals as much as what they read. The audit trail is protected as the data it describes, not as debug output.

+

Structure where structure exists

Regulatory obligations are a graph, not a bag of paragraphs. Similarity search alone loses the relationships that carry the meaning.

WHY THE DEMO DOES NOT DEPLOY

The gap between a working retrieval demonstration and a deployable one is not model quality. It is that the demonstration answers a different question — `can this work` rather than `can this be trusted, by whom, with what, and how would anyone know`.

Start with the leak nobody notices. Chunking a policy estate and sending the chunks to an embedding endpoint means the institution's documents have already left, before a single question is asked. Embeddings are not anonymisation; the content was transmitted to produce them. For a bank whose documents include customer detail, contractual terms and unreleased financial position, that transfer is the incident, and it happened during the pilot.

Then the permission problem, which is worse because it looks solved. The obvious implementation retrieves the relevant passages, generates the answer, and then checks whether the user is allowed to see it. But the model has already read the restricted material, and the answer it produced is shaped by content the user was not cleared for. Filtering the output does not unlearn the context. The information has already crossed.

THE LEAK

Documents leave the boundary to be embedded. The transfer happens once, at ingestion, and it is permanent.

THE FALSE FIX

Filter the answer. But the model already read the restricted context, and the answer is shaped by it.

THE ACTUAL FIX

Filter the retrieval. The model is only ever handed material this user could have opened.

WORKSTREAM 01

Retrieval Inside the Boundary

WHAT WE ACTUALLY DID

The entire pipeline runs inside the institution's environment: document processing, chunking, the embedding model, the vector store, and inference. Not a managed service with a data processing agreement — the institution's own infrastructure, where the network path to a public endpoint does not exist rather than merely being disabled.

Access control is carried through the pipeline rather than bolted to its ends. Every chunk retains the permissions of its source document, the querying user's identity travels with the query, and the vector search is filtered by that identity before it executes. The model receives only material the user could have opened themselves.

THE CHALLENGES WE ENCOUNTERED

- **Chunking destroys permission context.** A document has an owner and an access list. Its chunks, once embedded, are floating vectors in a store with no native concept of who may see what.
- **Filtered vector search degrades.** Applying a permission filter to an approximate nearest neighbour search is not free — naive post-filtering can return almost nothing when a user's permissions are narrow, because the top matches are all filtered away.
- **Permissions change; embeddings persist.** A document reclassified on Tuesday has chunks embedded on Monday, sitting in a store that does not know.
- **Self-hosted models are an operational commitment.** Running inference internally means owning GPU capacity, model updates and capacity planning — a real cost a hosted endpoint hides.
- **Document quality is worse than anyone admits.** The estate is scanned PDFs, tables that lost their structure, documents whose current version is ambiguous, and drafts indistinguishable from approved policy.

HOW WE RESOLVED THEM

- Permission metadata travels with every chunk as first-class state, and is evaluated at retrieval time against the caller's identity rather than assumed at ingestion.
- Filtering is applied within the search rather than after it, so that a user with narrow permissions receives the best results they are entitled to rather than the leftovers of a search run for someone else.
- Reclassification propagates to the index. A permission change on a source document updates its chunks, because an index reflecting last month's classification is a compliance finding waiting to be written.
- Document quality was treated as an engineering problem with a budget rather than a precondition someone else would fix. Where the estate contained two plausible current versions, the system says so rather than silently picking one.

HOW WE PROVED IT WORKED

We tested the permission boundary by attacking it. Queries were constructed to elicit restricted content indirectly — asking around a document rather than for it, requesting summaries and comparisons that would require reading material the user could not open. Retrieval was verified at the vector store rather than at the interface, confirming that restricted chunks were never returned rather than merely never displayed. **Egress was verified by network policy, not by configuration review: the path to a public endpoint does not exist.**

RISK AVOIDED

The institution's document estate transmitted to a third party during a pilot nobody classified as a data transfer — discovered afterwards, when the vendor's retention terms become the subject of a very different conversation.

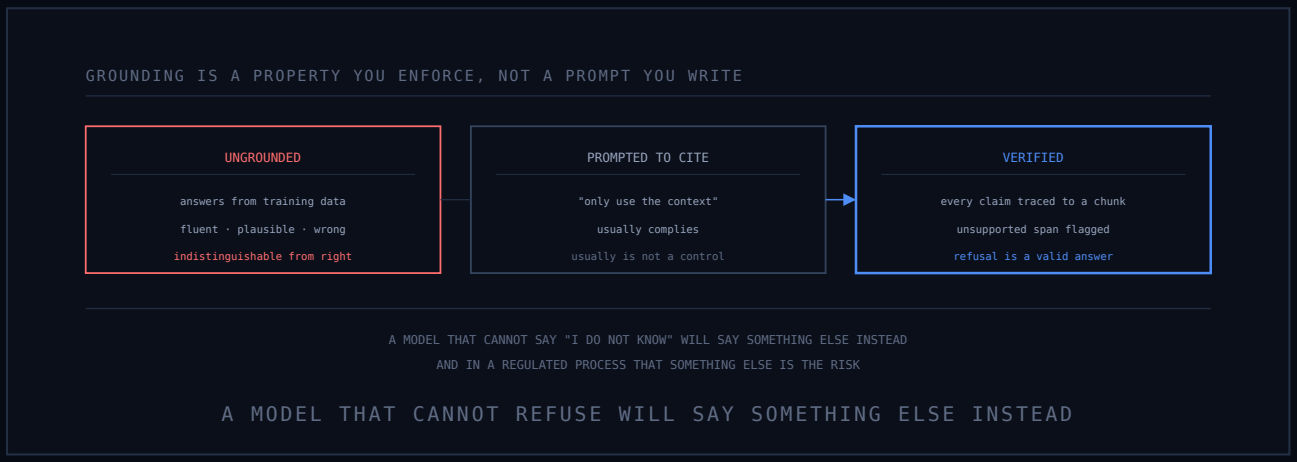
WHAT A US BANK SHOULD TAKE FROM THIS

Ask where your documents were when they were embedded. Not where the answers are generated — where the content was when the vectors were produced. That transfer happened at ingestion, and if it crossed a boundary, it has already crossed.

WORKSTREAM 02

Grounding, Citation and Refusal

A model asked a question it cannot answer from the retrieved context will answer it anyway, from training data, in the same confident register as a well-grounded response. Nothing in the output distinguishes the two.



WHAT WE ACTUALLY DID

Every claim in an answer is tied to the passage that supports it, and the citation is inspectable rather than decorative — a reviewer follows it to the paragraph in the institution's own document. Where retrieval returns nothing adequate, the system refuses and says why, rather than producing a plausible answer from general knowledge.

Refusal had to be engineered deliberately. The default behaviour of a language model handed thin context is to fill the gap gracefully, and that graceful fill is precisely the failure mode a regulated process cannot absorb.

THE CHALLENGES WE ENCOUNTERED

- **"Only use the provided context" is an instruction, not a control.** It works most of the time. Most of the time is not a standard that survives a model risk review.
- **Plausible-but-wrong is worse than obviously wrong.** An answer that is confidently incorrect and internally coherent will be acted upon, and the reviewer who might have caught it had no signal that anything needed checking.
- **Similarity is not relevance.** The passage most semantically similar to a question is frequently not the passage that answers it, particularly where the institution's vocabulary diverges from the regulator's.
- **Structure is lost in a flat index.** Regulatory material is a web — this control satisfies that obligation, which requires this evidence. Similarity search over chunks cannot represent the relationships, and confidently returns something adjacent instead.
- **Refusal is unpopular.** A system that says "I do not know" gets described as unhelpful, and the pressure to tune the threshold down arrives within a week of launch.

HOW WE RESOLVED THEM

- Grounding is verified rather than requested. Generated claims are checked against the retrieved passages, and spans without support are flagged rather than shipped — the instruction is a starting point, the verification is the control.
- A knowledge graph of the regulatory structure complements the vector store. The graph constrains retrieval to what is structurally relevant to the obligation at hand; similarity supplies the institution's own language within that constraint.
- Refusal thresholds were set with the risk function rather than with the users, precisely because the users will lobby to lower them. The threshold is a control parameter, not a usability setting.
- Every answer carries its confidence and its sources, and low-confidence answers are presented as such rather than in the same register as well-grounded ones. The interface is required to convey the uncertainty that the model's fluency conceals.

HOW WE PROVED IT WORKED

Grounding was tested adversarially, with questions whose correct answers exist only in the institution's own documents and contradict the general regulatory position. A model answering from training data gets these confidently wrong, which makes them the only questions worth asking. Refusal was tested by querying for material genuinely absent from the estate, confirming that the system declines rather than constructs. Citations were verified by following them: **where a citation did not support the claim it was attached to, the answer was wrong regardless of how well it read.**

RISK AVOIDED

An articulate, well-cited, entirely unfounded answer entering a regulatory process — and being indistinguishable, at the moment it is acted upon, from the answers that were correct.

WHAT A US BANK SHOULD TAKE FROM THIS

Ask what your system does when the answer is not in the corpus. If it always produces something, you have not deployed a retrieval system — you have deployed a language model with a document estate attached and a plausible reason to trust it.

WHERE THIS BELONGS

The use cases are not exotic. They are the ones where a bank's people spend their days reading documents to answer questions the documents already answer — and where being wrong has a regulator attached.

WHERE IT FITS	WHY THERE SPECIFICALLY	MAPS TO
Policy and procedure	Staff asking what the institution's own policy requires, answered from the current version with the paragraph attached.	INTERNAL CONTROL
Regulatory change	What does this new obligation touch? A graph of controls and obligations answers structurally, not by keyword.	COMPLIANCE
Third-party review	Contract and vendor documentation queried at volume, inside the boundary, with permission intact.	INTERAGENCY TPRM 2023
Model documentation	Model risk teams reading their own inventory — with citation, so the answer is checkable.	SR 11-7
Audit and examination	Finding the evidence that supports a claim, with a trail showing where it came from.	INTERNAL AUDIT
Sovereign environments	Where the public endpoint is not an option at any price, self-hosted is the only architecture available.	DATA RESIDENCY

Private means private

Not a data processing agreement. Not a promise of non-retention. The documents do not leave, because the network path does not exist; the model sees only what the user could open, because permission is applied before retrieval; and every claim carries the passage that produced it, because an answer nobody can check is not an answer a regulated institution can use.

THE HONEST LIMITATIONS

- **Self-hosting is a real operational commitment.** GPU capacity, model lifecycle and capacity planning become yours. For most workloads that is the wrong trade; for a regulated document estate it is usually the only one available.
- **Self-hosted models lag the frontier.** The best available model is generally not the one you can run inside your own boundary. On grounded retrieval over your own corpus the gap matters far less than on open-ended reasoning — but it is real, and anyone who tells you otherwise is selling.
- **Retrieval quality is bounded by document quality.** A system over an estate with ambiguous versions and lost table structure inherits every one of those problems. This work is unglamorous and it is most of the effort.
- **Refusal has a cost.** Users will describe an honest system as less helpful than a confident one. That reaction is the system working, and it needs defending at the executive level before launch rather than after.

WHERE THIS GOES NEXT

> The retrieval demonstration is easy and everyone has seen one. The version that survives a security review, a model risk review and an examination is a different system that happens to look the same from the outside.
>
> If you have a pilot in flight, the question worth asking today is where your documents were when they were embedded — because that answer is already fixed.

TECH4BIZ SOLUTIONS | CODENUK | AUDITTRAX LABS

tech4bizsolutions.com codenuk.com auditraxlabs.com

contact@tech4biz.io