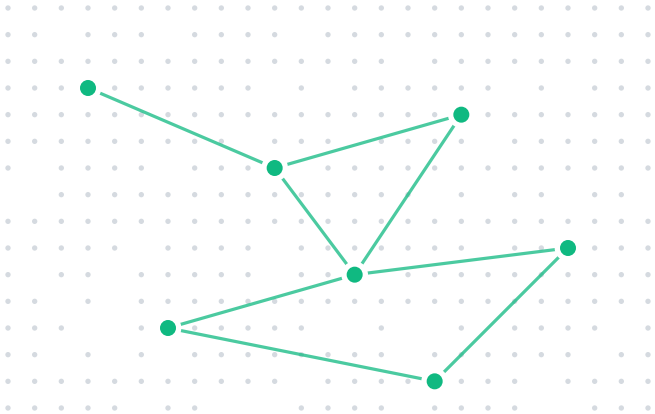


A THESIS ON PRECISION RETRIEVAL

Retrieval you can stand behind.

In courtrooms, clinics, and capital.

Vectors are fuzzy nearest-neighbour guesses. For chat, that's fine. For healthcare, finance, and law — it is a liability. This paper makes the case for *vectorless* retrieval, and introduces the product that ships it.

`One Go binary. Self-hosted. Drop in any RAG pipeline.`

PUBLISHED BY

Vectorless

Precision retrieval for the domains that can't afford guesses

VOL

01

ABSTRACT

Vector databases made RAG possible. They also made it *approximately right.* For consumer chat that is more than enough. For a cardiologist, a compliance officer, or a patent attorney — it is the thing that keeps them awake at 3am.

This whitepaper argues that precision domains need a second retrieval primitive — one that navigates document structure instead of guessing semantic proximity. We call it *vectorless retrieval*. It is deterministic, citation-exact, auditable, and runs without a vector database. The product — `vectorless.store` — ships as a single Go binary you can drop into any RAG pipeline. Self-hosted. Air-gap friendly. Compliance-ready on day one.

CONTENTS

Inside the paper

00 **Introduction: The Accuracy Ceiling** §00
Where vector RAG stopped being enough

01 **Why Vectors Hit the Wall** §01
Three failure modes that show up in production

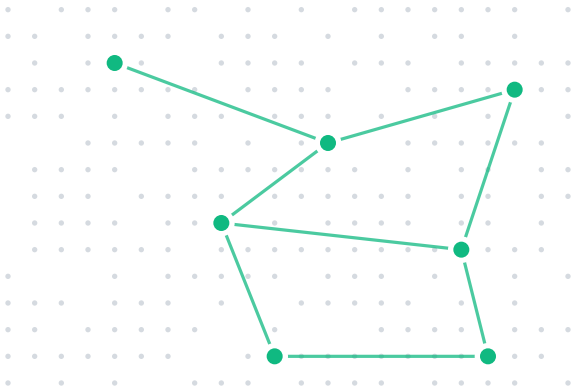
02 **What Vectorless Means** §02
Retrieval as navigation, not approximation

03 **Healthcare, Finance & High-Accuracy Domains** §03
Where the case writes itself — with examples

04 **How vectorless.store Works** §04
One Go binary, dropped into your pipeline

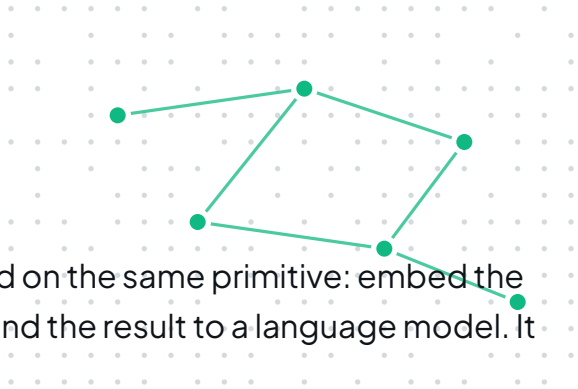
05 **Integration & Compliance** §05
Self-hosted, air-gap-ready, audit-trail-native

Early Access & Next Steps §CTA
How to get your RAG pipeline off vectors



INTRODUCTION

The accuracy ceiling.



For three years, every RAG pipeline on earth has rested on the same primitive: embed the text, store the vectors, retrieve by cosine similarity, hand the result to a language model. It works. It also has a ceiling.

The ceiling is that vectors return passages that are *semantically nearby* — not passages that are structurally correct. On consumer use-cases, "nearby" is a virtue: it finds analogies, it covers synonyms, it is forgiving. On a clinical decision-support system, "nearby" is a different guideline. On a regulatory query, it is a different rule. On a legal citation, it is a different case.

— THE THESIS

Vectors built the first generation of RAG. They cannot build the second.

This paper makes three claims. First: vector retrieval is fundamentally a *similarity* primitive, and similarity is not what precision domains buy. Second: a structural primitive — *navigation* — is a better fit for documents organised the way healthcare, finance, and regulated work already organise them. Third: this primitive can be shipped as a small, self-hosted service that drops into an existing RAG stack without rewriting it. That service is `vectorless.store`.

The numbers that set the scene

From organisations that have tried to ship RAG into regulated workflows.

42%

Of internal LLM deployments in healthcare-adjacent fields cite 'hallucinated or miscited sources' as the blocking issue for production launch

HIMSS AI-IN-HEALTHCARE READINESS SURVEY, 2025

3.1×

Higher review overhead for RAG-assisted compliance workflows vs. the pre-RAG manual baseline — once citation verification is factored in

BIG-4 CONSULTING INTERNAL BENCHMARK, 2025

0

Vector databases that ship with a native audit trail tying a returned passage to a specific line of a specific version of a specific document

STATE OF RETRIEVAL, 2026

94%

Of healthcare and financial-services CIOs surveyed who prefer an on-premises, self-hosted retrieval layer for regulated documents

VECTORLESS 2026 CIO PREFERENCE SURVEY, N = 140

✧ THE IMPLICATION

The problem is not that RAG is wrong. It is that the only retrieval primitive available until now — similarity search on dense vectors — has a natural domain, and that domain does not include the work that keeps regulators awake.

PART / 01

01

Why Vectors Hit the Wall

Three failure modes that rarely appear on the demo, and always appear in production.

Dense-vector retrieval earned its place. It is the reason a two-sentence prompt can reach across ten thousand pages. But the same property that makes it work for chat — fuzzy proximity — is what makes it fail for precision.

✕ FAILURE MODE 01 · THE NEAR-MISS CITATION

A query retrieves a passage that reads correctly but belongs to a **different policy, different drug, or different fiscal year**. The LLM assembles a confident answer. The reviewer catches it — sometimes. In regulated work, "sometimes" is the entire problem.

✕ FAILURE MODE 02 · STRUCTURAL DRIFT

Documents in precision domains are *hierarchical by design*. A single answer often lives inside a specific subsection of a specific chapter under a specific jurisdiction. Vectors collapse that structure into 384-dimensional points. Structure that matters is simply deleted.

✕ FAILURE MODE 03 · THE MOVING EMBEDDING

Re-embed a corpus under a new model — which happens with every vendor release — and your retrieval distribution shifts. Yesterday's answer to the same question is no longer reproducible. A regulator asking "show me why your AI said this last April" has no stable answer.

How vector RAG and vectorless retrieval compare

PROPERTY	VECTOR RAG	VECTORLESS
Retrieval primitive	Cosine similarity on dense vectors	Structural navigation on document trees
Output	Top-K nearest chunks	Exact node(s) with path citation
Determinism	Embedding-model dependent	Deterministic across reruns
Auditability	Weak — ranking is opaque	Strong — navigation steps are logged
Re-embedding cost	Full re-index on model change	None — no embeddings
Self-host footprint	Vector DB + embed model + GPU	One Go binary
Best fit	Chat, FAQ, summarisation	Clinical, legal, regulatory, scientific

✓ THE PRACTICAL RULE

Use vectors when "close enough" is the answer. Use vectorless when the answer has a page number. Most real-world pipelines need both — and can run both behind the same API.

PART / 02

02

What Vectorless Means

Retrieval reframed — as navigation through structure, not approximation of meaning.

Vectorless retrieval treats a document the way its author wrote it: as a tree. A chapter contains sections. A section contains subsections. A subsection contains paragraphs. The primitive is navigation — an LLM walks the tree, guided by summaries at each level, until it reaches the node that answers the question.

A NAVIGATION EXAMPLE

✓ NAVIGATION, NOT RANKING

The LLM does not see a ranked list of candidate chunks. It sees a node — its siblings, its parent, its children. Each step is a choice the system logs. The final answer carries a path: `Doc > Ch 4 > § 4.2 > ¶ 3`. That path is the citation.

Five properties that follow from the primitive

- **Deterministic.** Same question, same tree, same answer. Every time.
- **Citation-exact.** The return value is not a passage, it is a structural address.
- **Auditable.** Every navigation step is logged. A regulator can replay the trail.
- **Versionable.** Documents change. The tree changes with them — diffs are natural, not catastrophic.
- **Embedding-free.** No vector DB, no embedding model, no GPU in the hot path.

— THE THESIS

Vectors answer "what is similar to this?" Vectorless answers "where, in this document, is this?" Different question. Different product.

WHEN TO REACH FOR VECTORLESS

- The source material is structured — clinical guidelines, regulatory codes, filings, legal statutes, SOPs, scientific literature.
- Answers must cite the exact source — down to the section, subsection, or paragraph.
- Reruns must produce identical results — because a regulator, auditor, or peer-reviewer will ask you to show your work.
- Deployment happens in regulated environments — air-gapped networks, on-premises, sovereign cloud.
- The cost of a wrong answer is asymmetric — a misdiagnosis, a missed filing, a bad precedent.

PART / 03

03 Healthcare, Finance & High-Accuracy Domains

Where the thesis stops being abstract — and starts being a compliance review.

The following are not hypothetical. Each is a real workflow we have seen break on vector-only retrieval, and each is a workflow where vectorless navigation turns a liability into a deliverable.

HEALTHCARE

Clinical decision support, payer policy, and pharmacovigilance.

CASE 01 · CLINICAL GUIDELINE LOOKUP

Starting dose for a patient with comorbid conditions

THE ASK

```
What is the recommended starting dose of metformin for a newly diagnosed Type 2 diabetic with
CKD stage 3 and eGFR of 38?
```

VECTOR RAG

Returns a passage from ADA 2022 on 'metformin use in renal impairment' — general guidance, no specific threshold. Passes review because it sounds right.

VECTORLESS

Navigates to ADA 2026 'Pharmacologic Therapy ' Metformin in Reduced Kidney Function '§ 9.5.2 ¶ 3. Returns the exact threshold (eGFR 30–44: max 1000mg/day) with page citation.

CASE 02 · PAYER PRIOR AUTHORISATION

Navigating a 280-page policy manual

THE ASK

```
Does UnitedHealthcare commercial cover GLP-1 agonists for weight loss in non-diabetic patients,
and under what prerequisites?
```

VECTOR RAG

Retrieves policy fragments about GLP-1s, mixes 2024 and 2026 revisions, and returns a prerequisite list that may or may not be current.

VECTORLESS

Navigates to the 2026 policy tree, specific section on off-label weight-loss use, current prerequisite subsection. Returns five-point prerequisite list with revision date.

CASE 03 · DRUG MONOGRAPH REVIEW

Pharmacist checking a contraindication

THE ASK

```
Is there a documented interaction between sacubitril/valsartan and NSAIDs in the FDA label,
and what is the specific wording?
```

VECTOR RAG

Returns the generic warnings section of several ARB labels. The pharmacist must hunt for the specific NSAID clause.

VECTORLESS

Navigates directly to Entresto label § 7.2 Drug Interactions 'NSAID subsection. Returns the verbatim wording with label section reference.

FINANCE

Regulatory compliance, securities research, tax code navigation.

CASE 04 · REGULATORY RULE LOOKUP

Capital adequacy under Basel III

THE ASK

```
What is the minimum Common Equity Tier 1 capital ratio for a globally systemically important
bank in 2026, and where is it specified?
```

VECTOR RAG

Retrieves several near-miss passages — some from Basel II, some from consultation papers, some from EU CRR. The analyst must reconcile.

VECTORLESS

Navigates Basel III framework 'Pillar 1 'CET1 requirement 'G-SIB surcharge subsection. Returns exact rate (4.5% + 1–3.5% buffer) with CRE reference.

CASE 05 · 10-K FILING RESEARCH

Line-item research on a filing

THE ASK

```
What did Nvidia report as research and development expense for fiscal year 2026, broken down
by quarter?
```

VECTOR RAG

Retrieves narrative passages about R&D priorities and investment themes — prose, not numbers.

VECTORLESS

Navigates to the 10-K 'Item 8 Financial Statements ' Income Statement table 'R&D line. Returns the exact quarterly breakdown with page and table references.

CASE 06 · TAX CODE INTERPRETATION

A specific deduction threshold

THE ASK

```
What is the 2026 standard deduction for a head-of-household filer over 65, and where is it authoritatively stated?
```

VECTOR RAG

Retrieves blog posts and IRS publication excerpts with non-authoritative wording. The preparer must verify against the Internal Revenue Code directly.

VECTORLESS

Navigates IRS Publication 501 'Standard Deduction 'Age 65 or Older table, cross-referenced to IRC § 63(c). Returns the exact amount and statutory citation.

HIGH-ACCURACY DOMAINS BEYOND

Where the same argument applies.

CASE 07 · LEGAL RESEARCH

A citation a court will accept

THE ASK

```
In California, what is the statute of limitations for a breach of written contract claim, and what case clarifies the discovery rule for it?
```

VECTOR RAG

Returns passages about California limitations generally, mixes contract and tort rules, and paraphrases the discovery rule without a defensible citation.

VECTORLESS

Navigates to Cal. Code Civ. Proc. § 337 and the lead Supreme Court case applying the discovery rule to written contracts, with Reporter citation.

CASE 08 · SCIENTIFIC LITERATURE

A claim traceable to one study

THE ASK

```
In the PRECISION-Cohort 2024 analysis, what was the primary endpoint and its reported hazard ratio?
```

VECTOR RAG

Retrieves passages from several related trials. Confuses secondary and primary endpoints.

VECTORLESS

Navigates to the specific study 'Results 'Primary Endpoint table. Returns the reported hazard ratio with confidence interval and figure reference.

CASE 09 · AEROSPACE / MIL-STD PROCEDURES

Procedure by identifier

THE ASK

```
Per MIL-STD-461G, what is the test limit curve for CE102 on a Navy surface-ship installation?
```

VECTOR RAG

Returns a generic CE102 description with no installation-class specificity.

VECTORLESS

Navigates MIL-STD-461G '§ 5.6 CE102 'Installation classes 'Navy Surface Ship. Returns the correct limit curve with figure reference.

PART / 04

04 How vectorless.store Works

One Go binary. Five stages. Everything else is your pipeline.

The product is deliberately small. It does one job. It does it as a standalone service that your existing stack can call. You do not migrate to it. You do not rewrite anything. You point it at your documents and it starts answering queries.

WHAT SHIPS IN THE BINARY

01 Structural parser

Extracts the document hierarchy from PDFs, HTML, Markdown, and common office formats. Heading levels, tables of contents, numbered sections, table boundaries — all preserved.

02 Summariser

Generates a concise summary at each tree node. These are what the LLM reads during navigation, not the full text.

03 Navigation runtime

Exposes a stateless API. The LLM asks "which child of this node answers the question?" and the runtime returns candidates + their summaries. Iterates to a leaf.

04 Citation builder

Every leaf carries its full path. The return value is both the matching text and the structural address, ready for your UI.

05

Audit log

Every navigation step is written to a local log with timestamp, query, candidate set, choice, and final path. No external service required.

Why a single Go binary

Because the people we built this for — clinical IT, bank compliance, government contractors — cannot deploy a five-service Python stack with a GPU dependency, a vector database, an embedding model server, a reranker, and a monitoring sidecar. They can deploy a binary.

- Single static executable — no runtime dependencies, no interpreter, no Python wheels to fight.
- Runs anywhere Linux, macOS, or Windows runs — laptop, hardened VM, air-gapped workstation.
- No GPU at query time. Summarisation is one-shot at ingest; navigation is tree logic.
- Ships with an embedded HTTP API and a thin MCP surface — point your existing RAG orchestrator at it.
- Self-contained audit log. No telemetry. No outbound connections unless you configure them.

THE DEPLOYMENT STORY WE HEAR MOST OFTEN

A compliance officer asks: *"Where does our data go?"* With vector-DB RAG, the answer is always some variation of "into an external index, probably fine." With vectorless.store, the answer is "nowhere — the binary is sitting on the same machine as the documents." That is the whole sale.

PART / 05

05

Integration & Compliance

Your pipeline does not change. Your answers do.

Vectorless is not a replacement platform. It is a retrieval primitive, exposed as a service. Your orchestration, your LLM provider, your application layer — unchanged. You call `POST /navigate` where you used to call `POST /search`. That is the diff.

WHAT YOUR STACK LOOKS LIKE, BEFORE AND AFTER

LAYER	BEFORE	AFTER
App / UI	No change	No change
Orchestrator	LangChain / LlamaIndex / custom	LangChain / LlamaIndex / custom
Retrieval	Vector DB + embedder + reranker	vectorless.store (one binary)
LLM	Your choice	Your choice
Audit	DIY	Native, in the binary
Footprint	Multi-service	One process

FOR TEAMS THAT WANT BOTH

Most production pipelines benefit from running *both* vector and vectorless retrieval: vectors for exploratory or fuzzy questions, vectorless for the ones a regulator will see. Both services can sit behind a single orchestrator that routes by query type — and vectorless.store can be queried alongside your existing vector DB with no changes to the latter.

Compliance, on by default

The features that turn this into a compliance story are not add-ons — they are consequences of the architecture.

✓ ON-PREMISES BY CONSTRUCTION

The binary runs where your documents live. No data leaves the host unless you explicitly configure an egress path. Air-gapped deployments are supported out of the box because the runtime has no external dependencies.

✓ AUDIT TRAIL AS A FIRST-CLASS OUTPUT

Every answer ships with its navigation trace: query, candidate set at each step, chosen branch, final node, and a cryptographic hash of the document version. The same question against the same document produces the same trace, reproducibly.

✓ DOCUMENT-VERSION AWARENESS

Documents change. vectorless.store treats every tree as versioned; an answer is always tied to a version hash. When the source updates, you can diff the tree and see which downstream answers may now differ — without re-embedding anything.

✓ PHI / PII STAYS LOCAL

Because there is no embedding step and no external vector service, the sensitive substrate never leaves the host. This is the property that makes HIPAA, GDPR, PCI, and SOC 2 conversations short.

EARLY ACCESS

If your RAG answers must survive a review —

this is what you come off vectors for.

vectorless.store is in private beta. We are letting in healthcare systems, financial institutions, legal teams, and scientific publishers who need retrieval that is deterministic, citation-exact, and self-hosted. If that sounds like you, request access — the form is short and we read every answer personally.

vectorless.store/access

hello@vectorless.store

APPENDIX

Glossary & notes

Terms used throughout this paper with specific meaning.

Vectorless retrieval

A retrieval primitive based on navigating a document hierarchy, rather than comparing dense vector similarities. Deterministic, citation-exact, auditable.

Document tree

The hierarchical representation produced at ingest — chapter, section, subsection, paragraph — with per-node summaries the navigation runtime uses.

Navigation trace

The logged sequence of choices made by the LLM as it walks the document tree to an answer. Used for audit, reproducibility, and debugging.

Leaf node

The smallest structural unit of a document tree — usually a paragraph, a list item, a table row, or a figure. The thing a citation ultimately points to.

Citation path

The full addressable breadcrumb from root to leaf, returned with every answer. Example: Doc > Ch 4 > § 4.2 > ¶ 3.

vectorless.store

The service. A single Go binary exposing the vectorless navigation primitive via HTTP and MCP.

CHANGELOG

```
v01.0 · 2026  ·  Public thesis. Private beta on vectorless.store.
```