Semantic Search On Enterprise Data

Revolutionizing the knowledge discovery

Rishabh Bohra
Solutions Architect

$whoami

1. Solutions Architect @ OpsTree Global
2. I will code for Tiramisu
3. My Co-ordinates are here - rbohra.com

Agenda

01

Evolution of Search

From keywords to semantic understanding

02

Vector Technology

How machines understand meaning

03

Enterprise Challenges

Unstructured data and security

04

RAG Architecture

Building retrieval-augmented systems

05

Implementation

Real-world patterns and best practices

> "At Uber, our systems handle massive amounts of data daily, from ridesharing to delivery. We've traditionally used keyword-based search with Apache Lucene™. However, we needed to move beyond simple keyword matching to semantic search to understand the meaning behind searches."
>
> —Uber Engineering

The Google Evolution

2000: PageRank

Keywords and link-based relevance

2013: Hummingbird

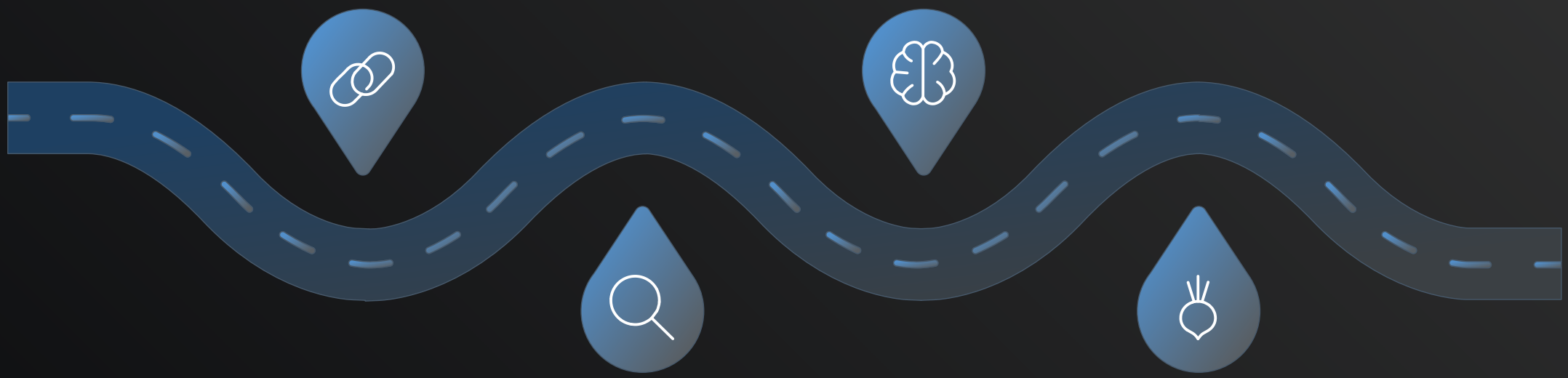
Semantic Search Era begins

2015–2019

RankBrain then BERT for deeper understanding

Now: Gemini

AI-driven mapping of intent to context



Search isn't about matching strings anymore.

It's about mapping query intent to document context.

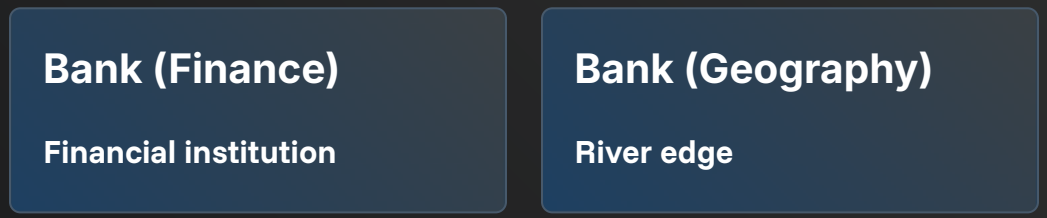
We are now engineering understanding.

Limitations of Keyword Based Search

Keyword Search: WHERE text LIKE '%bank%' matches both.

Semantic Search: Understands context (Finance vs. Geography).

Standard SQL fails at nuance.



The Enterprise Disconnect

Public Web

Predictive, personalized, semantic.

Enterprise Intranet

"No results found for 'Q3 proj'."

Goal: **Bring consumer-grade search intelligence to enterprise data.**

Vector Representations

Computers don't understand words

They understand numbers.

Vectors: Converting text into lists of floating-point numbers.

Proximity in space = Similarity in meaning.

The "King - Man + Woman = Queen" Equation

a famous example used to illustrate how computers understand the meaning of words through word embeddings

1	2	3
Embeddings capture relationships	Indexing concepts, not just words	Works for code, images, audio

We aren't just indexing words; we are indexing concepts.

This applies to code, images, and audio, not just text.

The Embedding Model

The Engine

Translates human language to machine language

Dense Vectors: High dimensionality (e.g., 1536 dimensions) captures deep context.

Why SQL Can't Handle This

01

Relational DBs are optimized for exact matches

02

High-dimensional vector math requires different indexing (HNSW, IVFFlat)

03

Enter the Vector Database

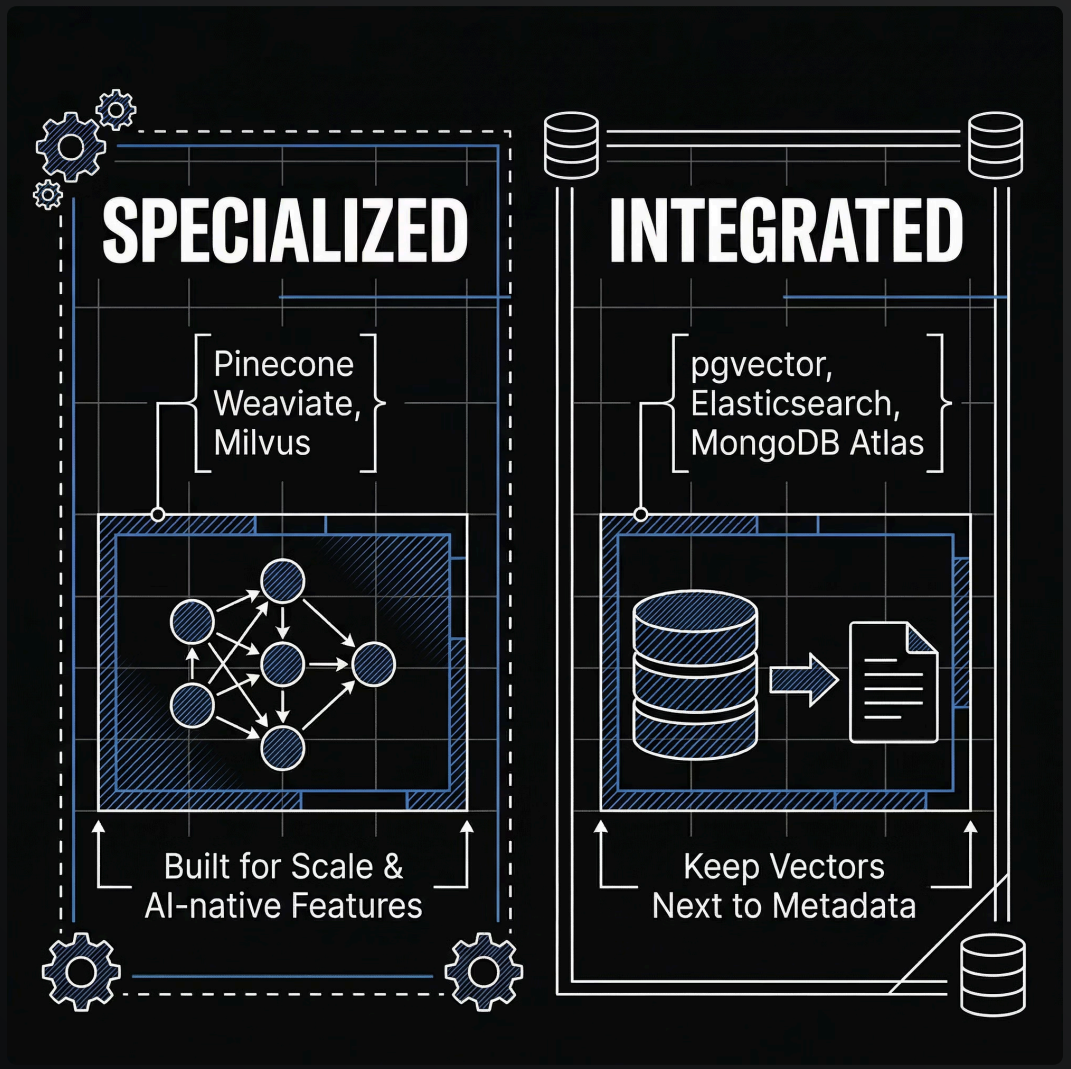
Approximate Nearest Neighbor (ANN) Universe

Exact KNN is too slow for millions of vectors.

ANN: Trades 1% accuracy for 100x speed.

This is how we search 10M documents in milliseconds.

The Vector DB Landscape



Pure Vector DBs

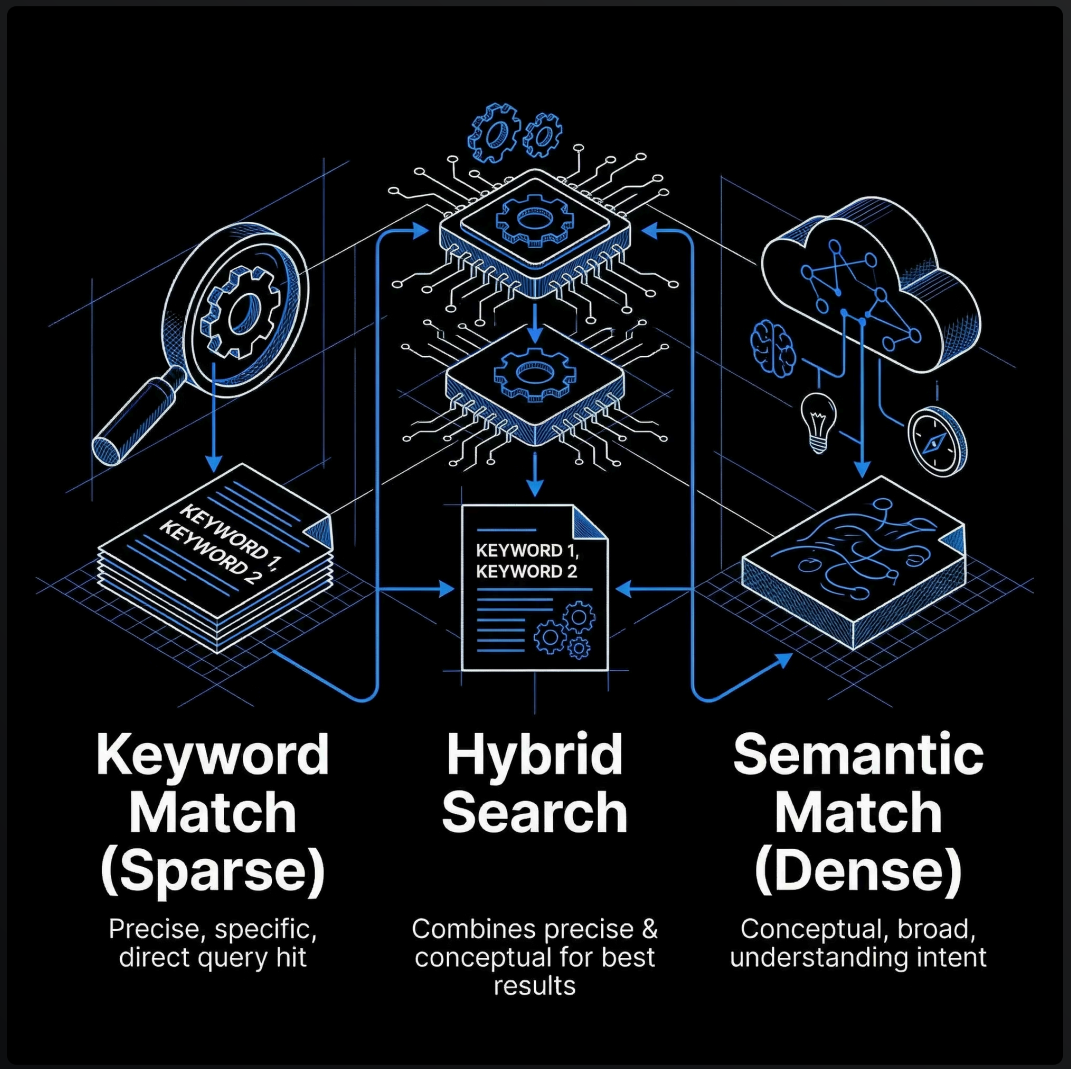
Built for scale and AI-native features.

Integrated

Great for keeping vectors next to metadata.

Architectural Choice: **Do you need a new stack or just a plugin?**

Sparse vs. Dense Vectors

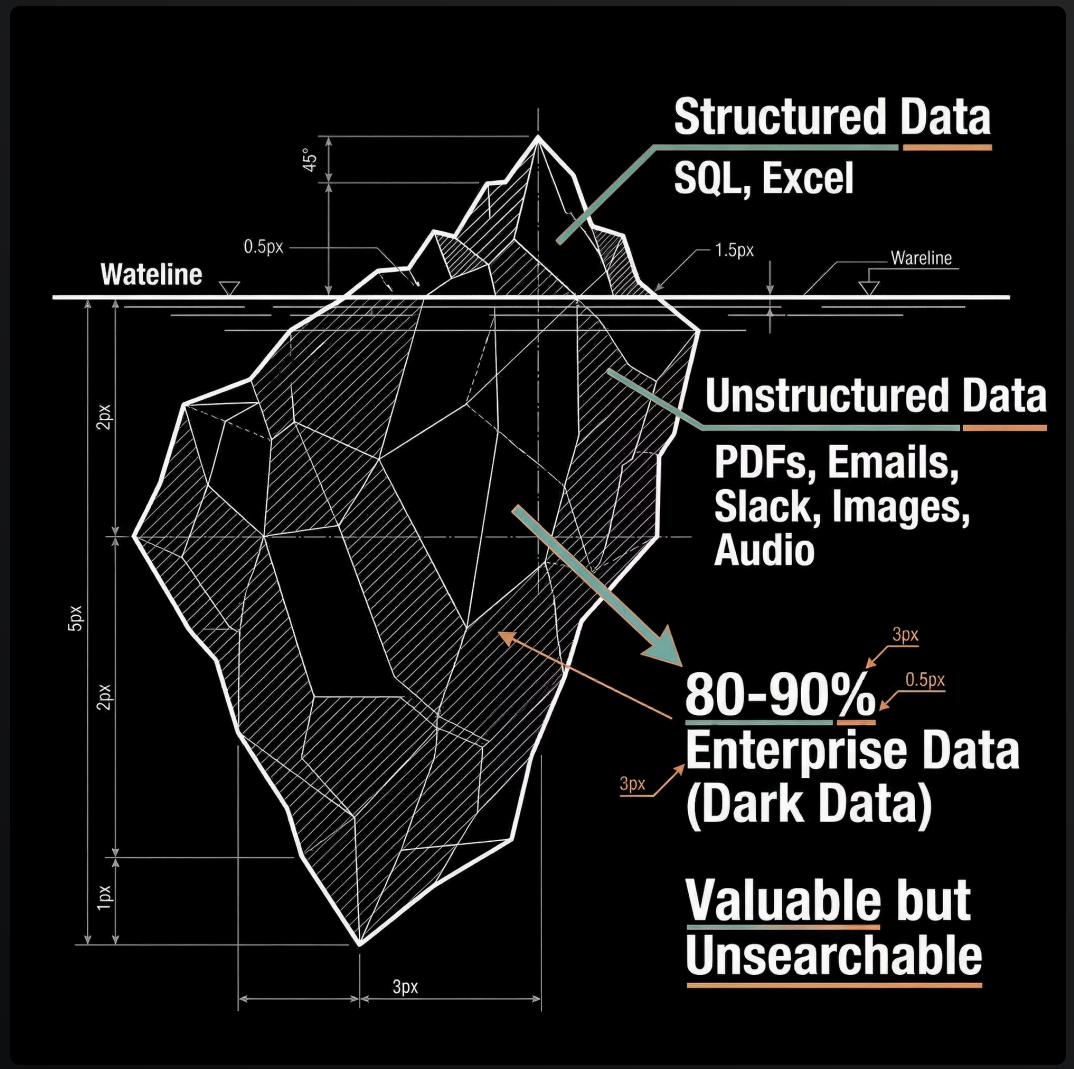


Sparse (BM25): Good for part numbers, specific names.

Dense: Good for "How do I fix the printer?"

Hybrid: The best of both worlds.

The Unstructured Data Explosion

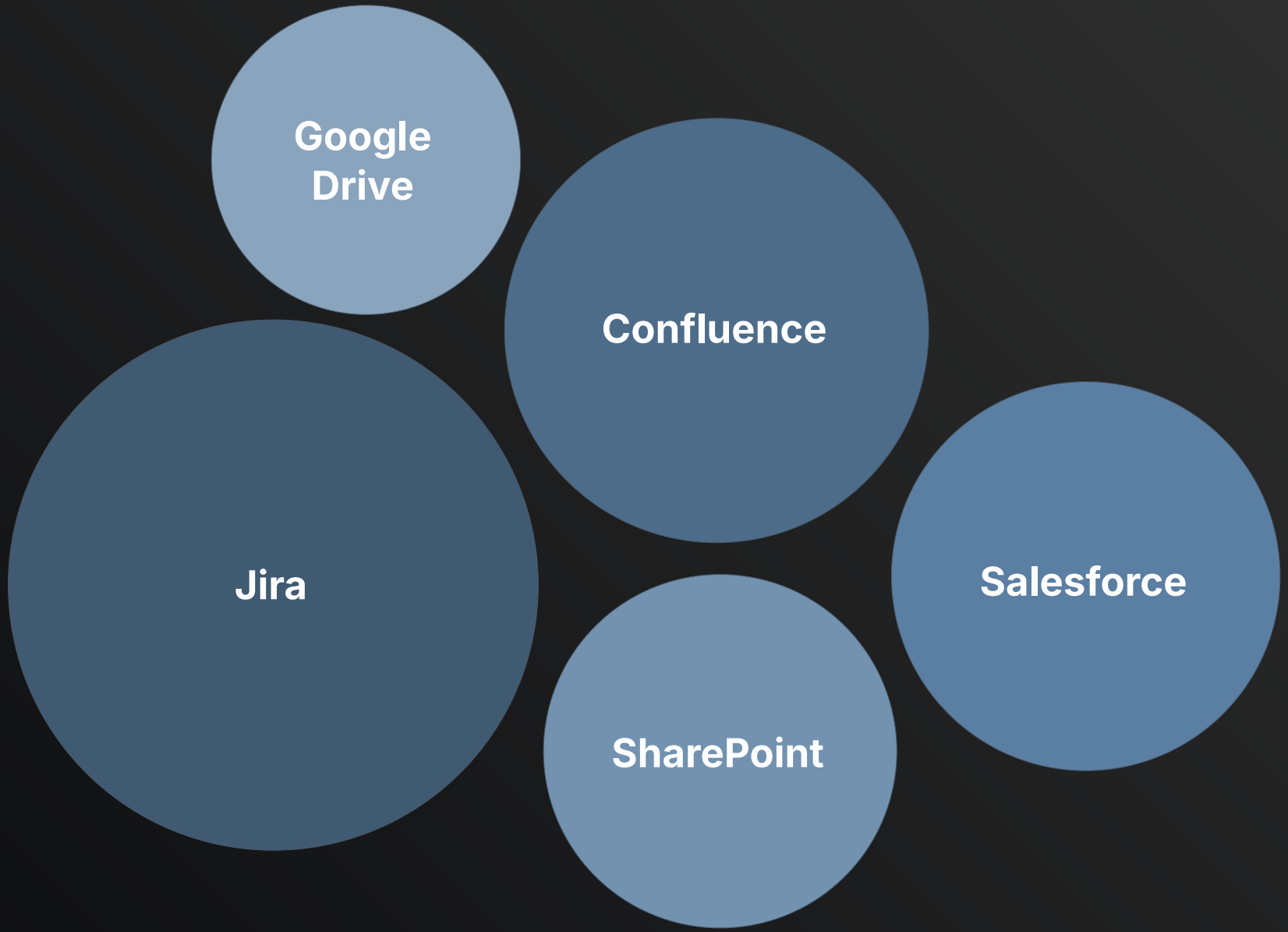


Industry Stat: ~80-90% of enterprise data is unstructured.

This is "Dark Data"—valuable but unsearchable.

GenAI is the key to unlocking this.

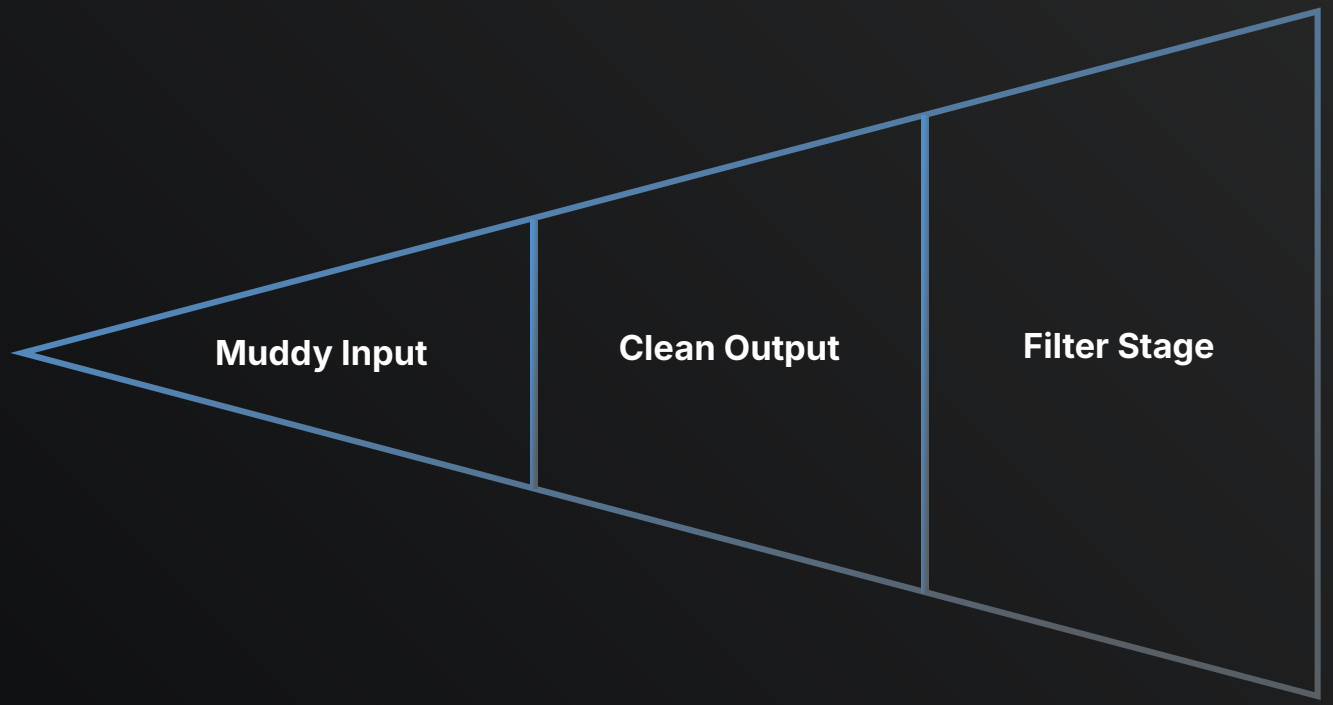
The Silo Problem



Context is fragmented across platforms.

A "Project X" search needs to hit design docs, legal contracts, and engineering tickets simultaneously.

The "Garbage In" Challenge

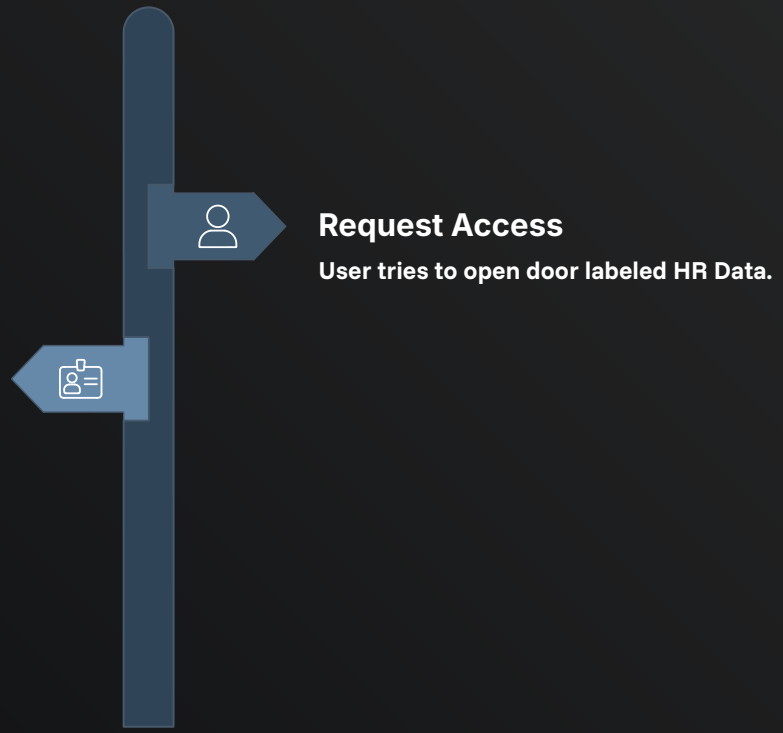


Embeddings are sensitive to noise.

Indexing HTML headers, footers, and legal disclaimers dilutes quality.

Preprocessing is 60% of the work.

Security & ACLs (Access Control Lists)



The Hardest Part: You cannot semantic search data the user isn't allowed to see.

Post-filtering (Slow) vs. Pre-filtering (Complex).

Enterprise search must respect permissions.

Multimodality in the Enterprise

Beyond Text

Search isn't just text.

Visual Search

Searching inside CAD drawings, marketing assets, and video calls.

CLIP/VQA Models

Connecting pixels to text.

Introducing RAG (Retrieval Augmented Generation)

User Question

Search Knowledge Base

Retrieve Context

Send to LLM

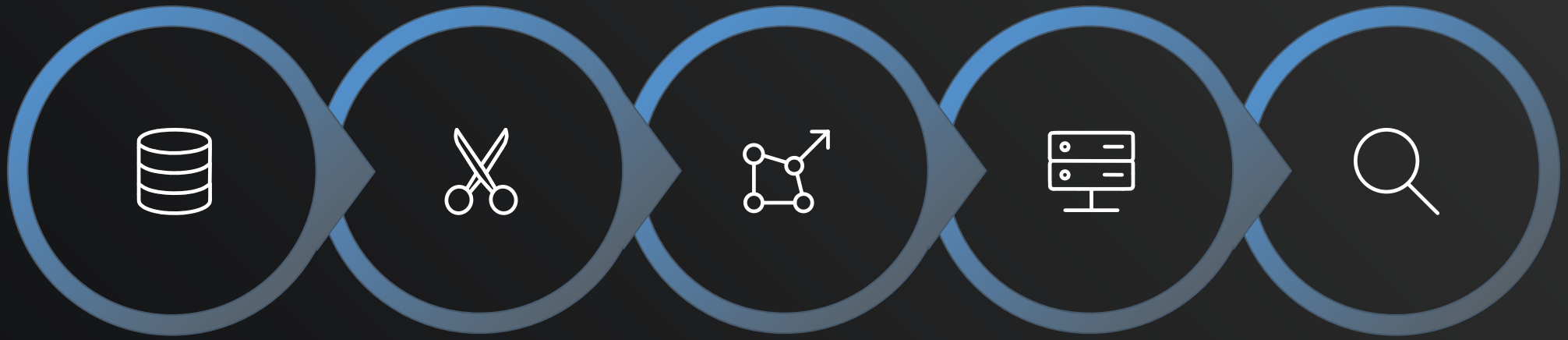
Answer

LLMs have amnesia. They only know training data.

RAG: Giving the LLM an open-book test.

Semantic Search is the "Retrieval" engine of RAG.

The High-Level Pipeline



Ingest Chunk Embed Store Retrieve

A standard pattern for Enterprise AI Search.

Every step impacts the final accuracy.

Ingestion: The ETL of AI

Parsing complex PDFs (tables, columns) is non-trivial.

Visual structure contains semantic meaning.

Tooling: Unstructured.io, LangChain loaders.

PDFs

PPTs

Docx

The Art of Chunking

❌ **Bad**

Cutting sentences in half (Fixed Size)

✔ **Good**

Cutting by paragraph/header (Semantic Chunking)

Chunking Strategy: Determines what context gets retrieved.

Too small = loss of context.

Too big = noise for the LLM.

Metadata Extraction

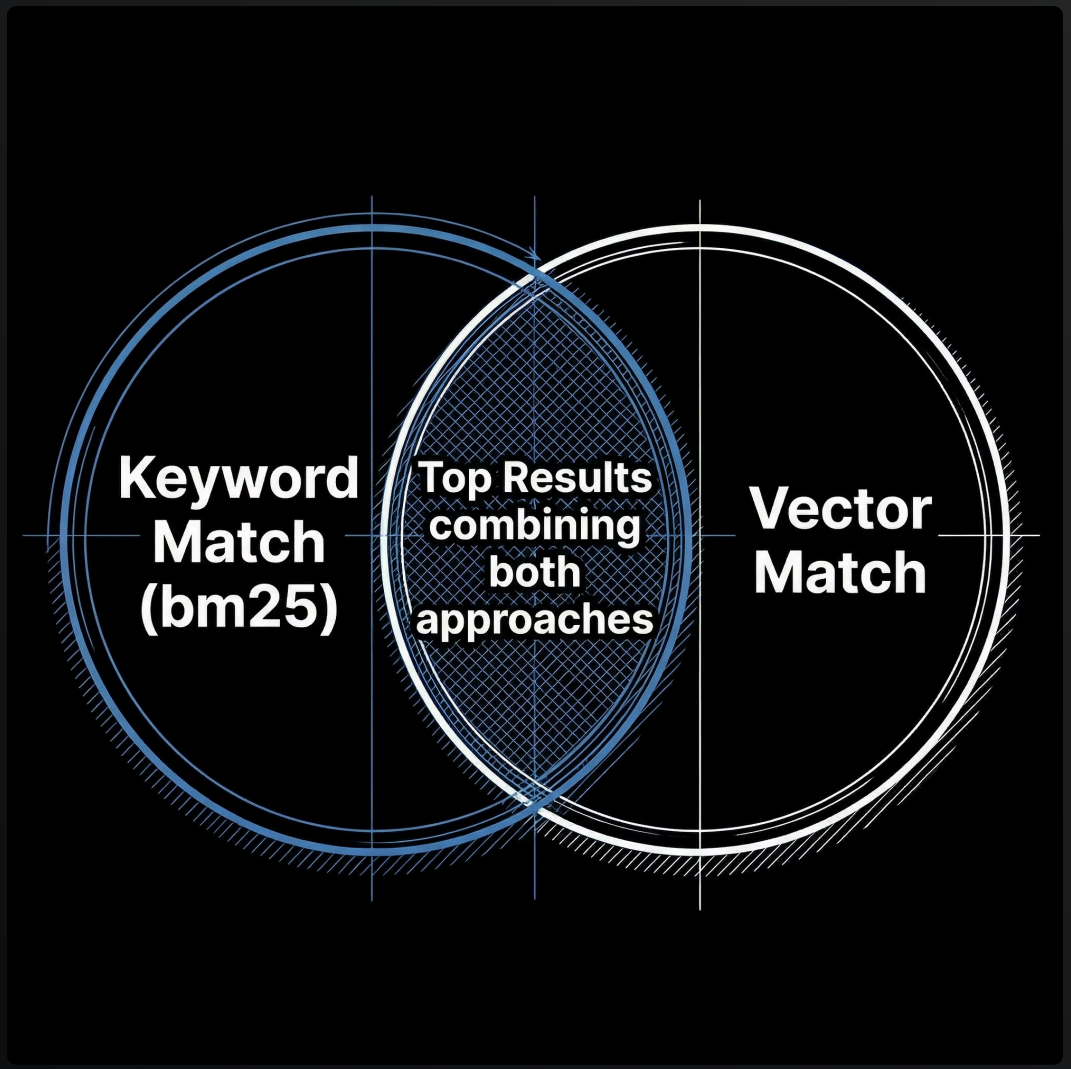
AUTHOR: JOHN | DATE: 2024 | CATEGORY: FINANCE

Vectors find similarity; Metadata provides filtering.

"Show me contracts (Tag) similar to X (Vector) from 2023 (Tag)."

Use LLMs to auto-generate metadata during ingestion.

Hybrid Search (The Golden Standard)

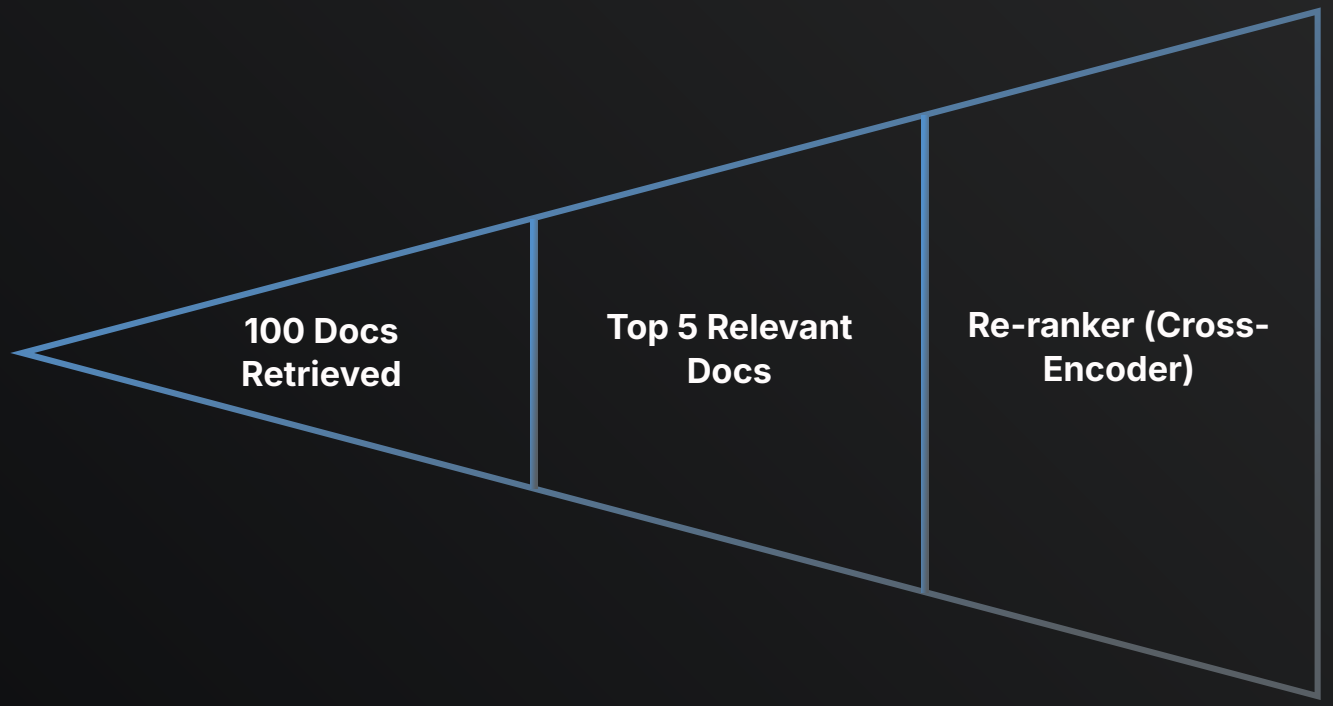


Vectors fail at exact acronyms (e.g., product code "XJ-900").

Keywords fail at concepts.

Architecture: Run both, combine scores (Reciprocal Rank Fusion).

Re-Ranking (The Accuracy Boost)



Bi-encoders (Vectors) are fast but less accurate.

Cross-encoders are slow but highly accurate.

Strategy: Retrieve 50 with vectors, Re-rank top 5 for the LLM.

The Context Window vs. Retrieval

Small Window

Old LLMs

Massive Window

Gemini 1.5 Pro

"With 1M token context, do we still need RAG?"

Yes: For latency, cost, and distinctness.

"Needle in a haystack" problems still exist in massive contexts.

Evaluating the Pipeline (RAGAS)

3

Key Metrics

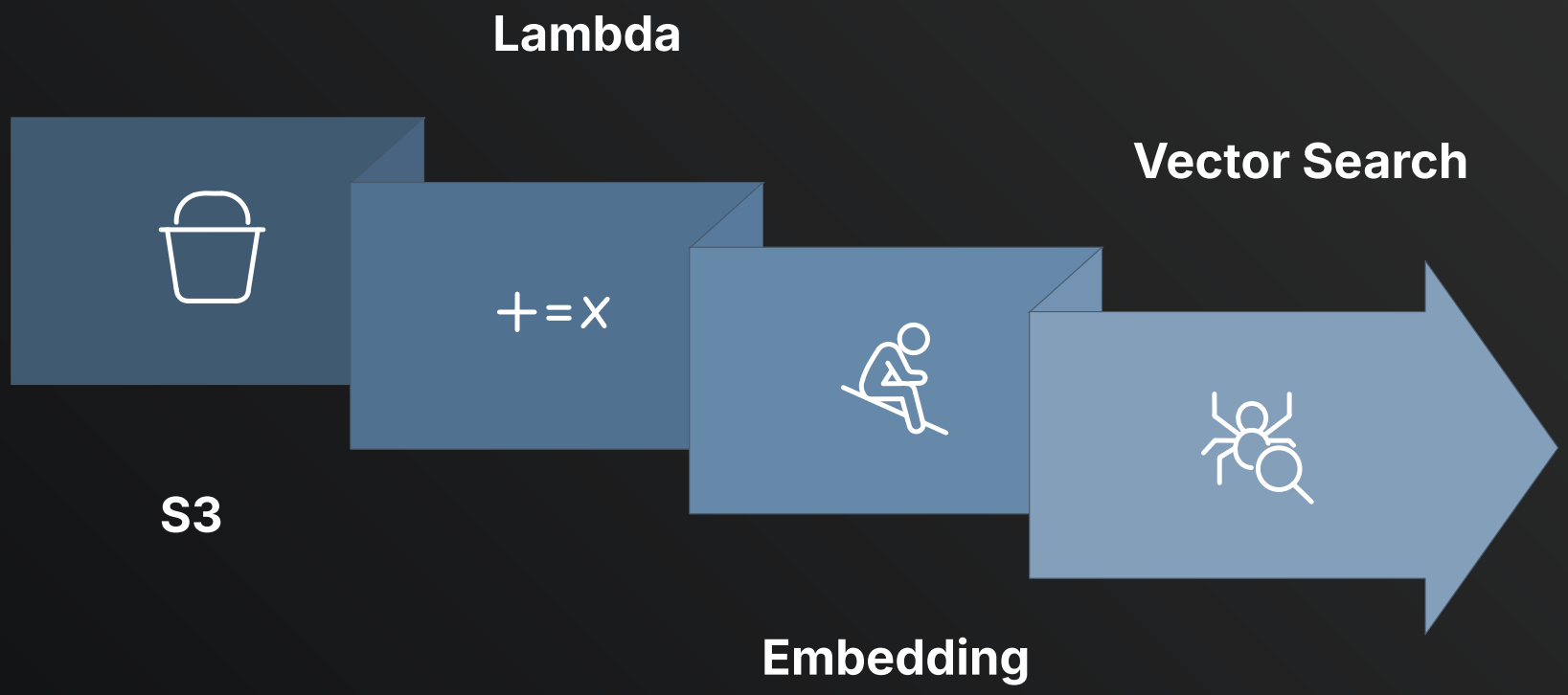
Faithfulness, Answer Relevance, Context Precision

You can't improve what you don't measure.

RAG Triad: Context Relevance, Groundedness, Answer Relevance.

Frameworks: RAGAS, TruLens, Arize Phoenix.

Architecture Diagram: AWS/Azure/GCP



Serverless architectures scale well.

Keep data residency and compliance in mind.

Real World Ref: Glean / Notion AI

Leading internal search tools.

They Index everything (Slack, GitHub, Drive).

They rely heavily on permission-aware semantic search.

Slack

GitHub

Drive

Real World Ref: E-Commerce Discovery

Upload Photo

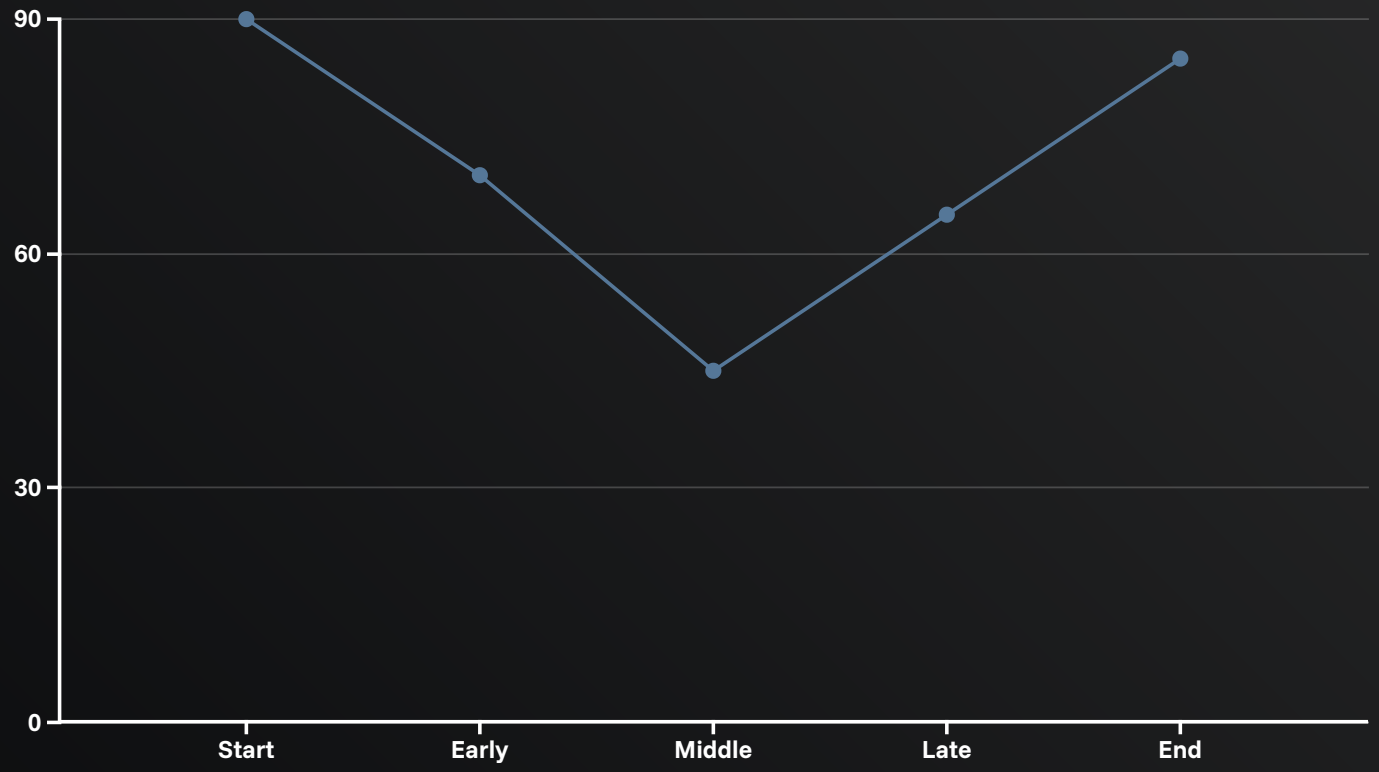
Show Matches

Vector Search in Retail: "Visual Search."

Embeddings of product images + text descriptions.

Increases discovery of items users can't describe in words.

The "Lost in the Middle" Phenomenon



LLMs pay more attention to the start and end of prompts.

Fix: Re-ranker should place most relevant chunks at the edges of the prompt.

Latency Engineering

Search is fast

Generation is slow

Technique: Stream tokens to the user immediately.

Technique: Cache common queries (Semantic Caching).

Cost Management

$$$

Vector storage + LLM tokens

Optimization: Use smaller quantization (binary embeddings).

Optimization: Don't vector index cold data (archive).

Handling Data Updates

The "Stale Index" problem.

Real-time data (stock prices, recent slack) needs immediate indexing.

Streaming

Kafka → Vector DB

Batch

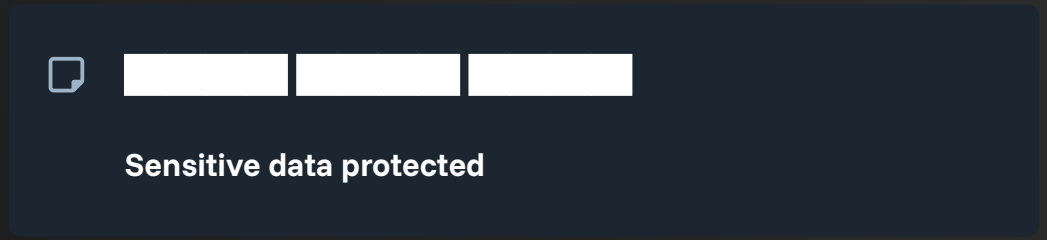
Scheduled jobs

Privacy & PII Redaction

Never send PII to public embedding APIs.

Use local embedding models (Ollama, HuggingFace) for sensitive data.

Sanitization: Redact PII before embedding.



Hallucinations & Citations

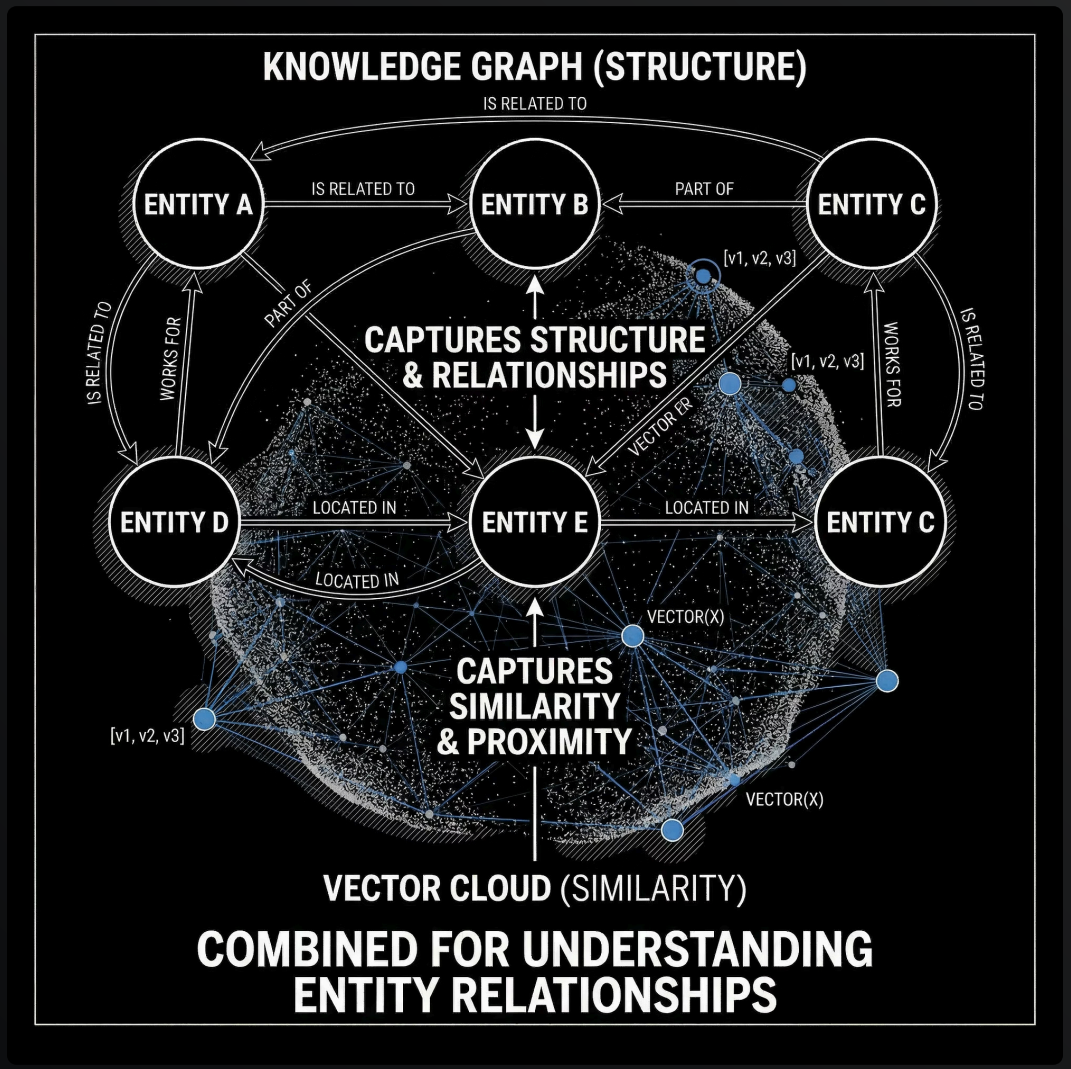
Trust is the currency of Enterprise AI.

Requirement: Every claim must cite the source chunk.

If the distance score is too low, say "I don't know."

AI response with footnotes [1], [2] linking to original PDFs.

GraphRAG (The Next Frontier)



Vectors capture similarity. Graphs capture structure.

GraphRAG: "How are these two entities related?"

Combining structured relationships with unstructured semantic meaning.

Agentic Search

Autonomous Agent

From Read-Only to Read-Write.

Search becomes the memory for autonomous agents.

"Find the invoice AND pay it."

Summary Checklist

01

Fix Data Ingestion (Garbage In/Out)

02

Use Hybrid Search (Keywords + Vectors)

03

Implement Re-ranking

04

Solve for ACLs/Security

05

Measure with RAGAS

Q&A / Thank You

Build something semantic.

Questions?