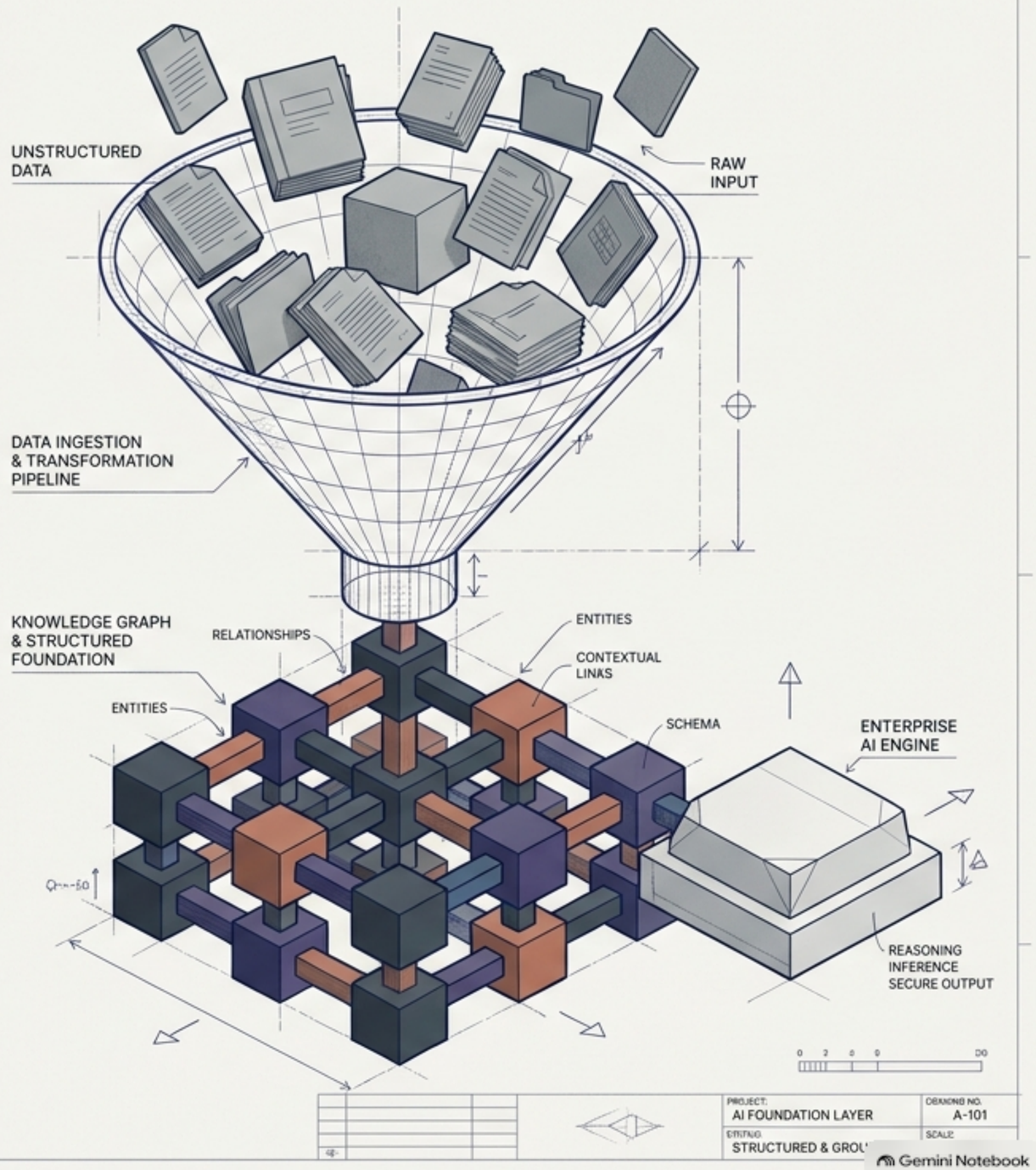
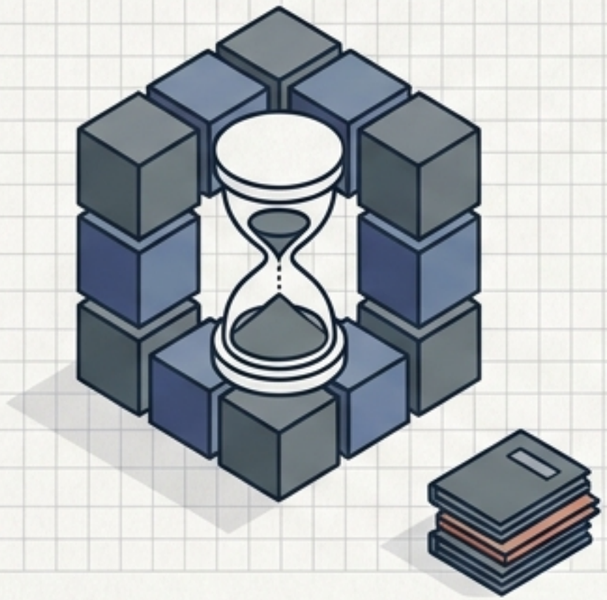
Grounding AI in Enterprise Truth

Large Language Models possess the reasoning power to transform business—but they lack the one thing that that matters most: your proprietary data.



The Three Flaws of Out-of-the-Box LLMs

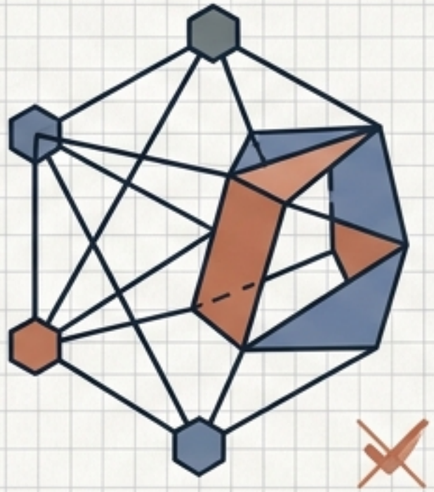


Frozen Knowledge

Models are frozen at their last training cutoff. They cannot access real-time changes in policies, markets, or inventory.

Limited Context Windows

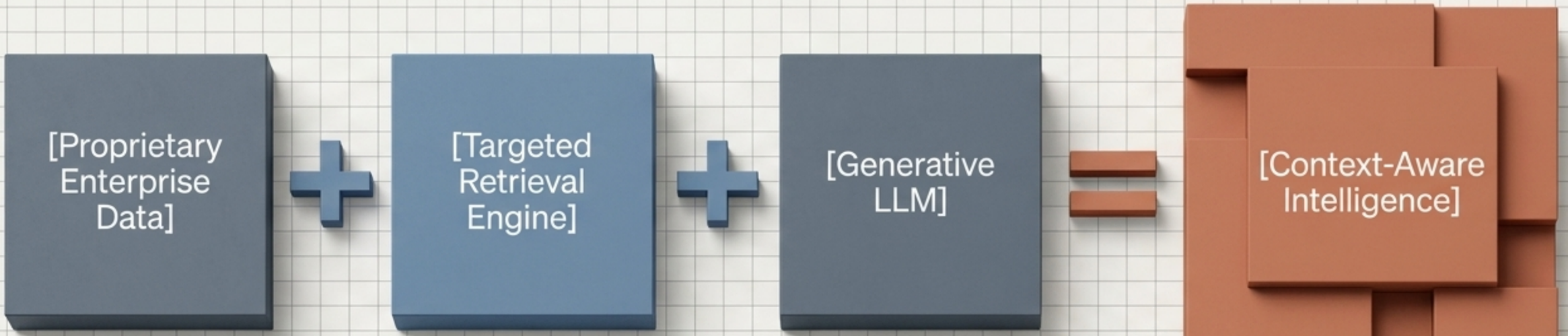
You cannot simply paste a 10,000-page enterprise knowledge base into a single prompt.



Hallucinations

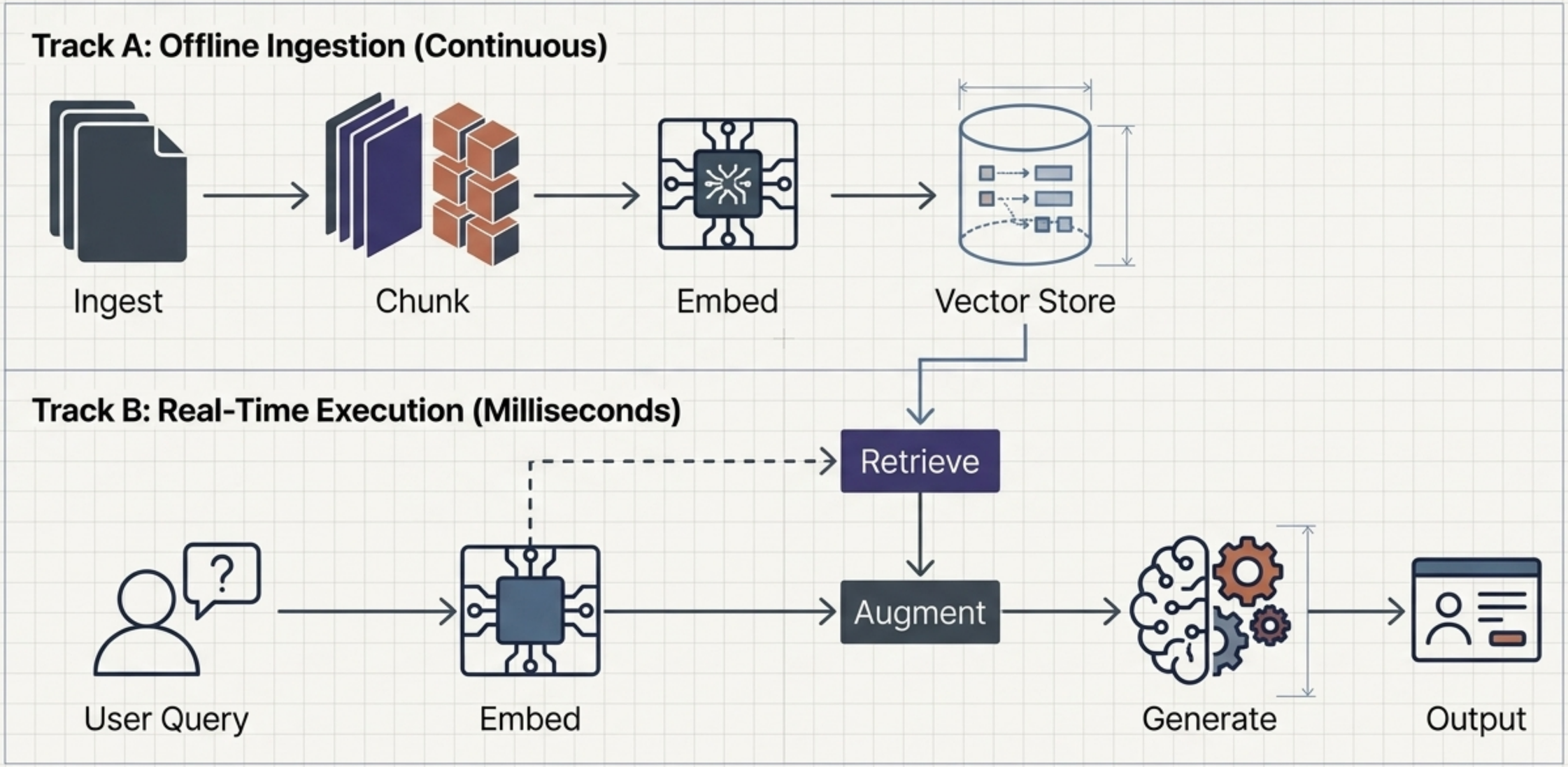
When an LLM lacks specific information, it predicts what sounds plausible, leading to confidently presented, factually incorrect outputs.

The Retrieval-Augmented Generation (RAG) Equation

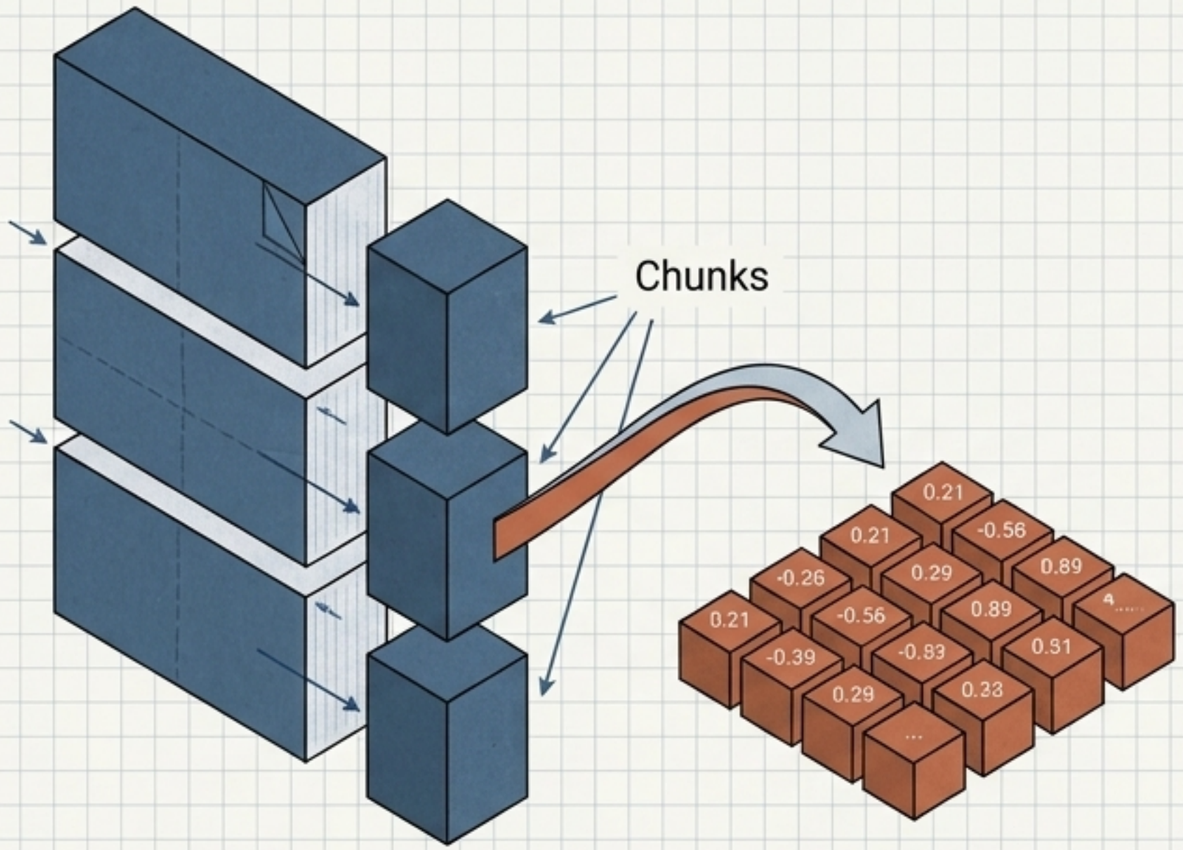


RAG is a hybrid framework that bridges the gap. It intercepts a user's prompt, searches a private database for the exact factual context needed, and forces the LLM to generate an answer grounded only in that retrieved truth—eliminating hallucination and utilizing real-time data.

The Dual-Track RAG Pipeline



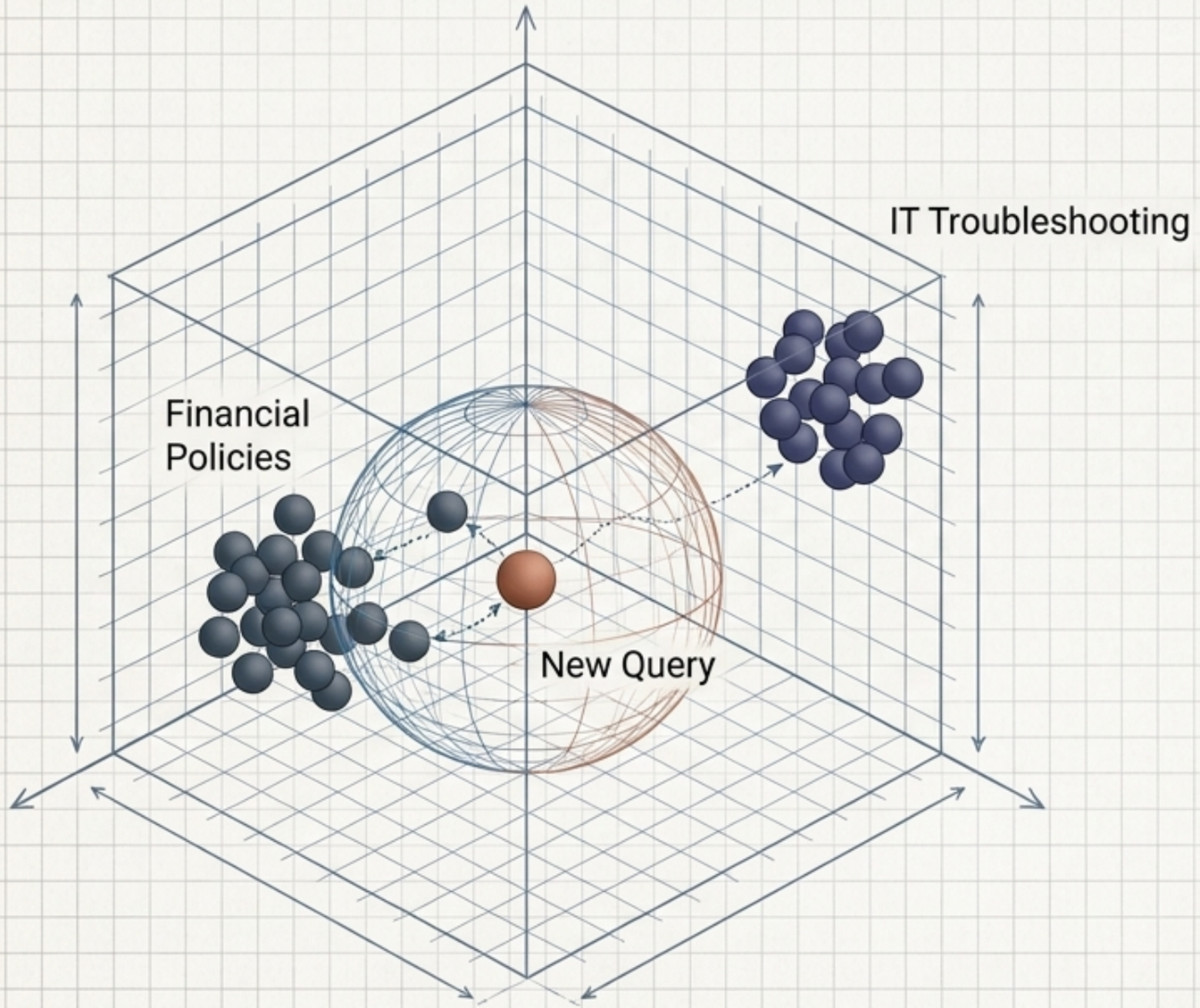
Data Preparation: Chunking and Embeddings



The Goldilocks Rule of Chunking

Too Large:	Includes irrelevant noise that confuses the retrieval model and wastes the LLM's context window.
Too Small:	Strips away the semantic meaning required for the LLM to understand the context.
Goal:	Preserve meaningful context while optimizing for exact retrieval.

The Vector Database: Searching by Meaning, Not Keywords



The Structure

Vector databases (like Chroma DB or FAISS) store data as high-dimensional coordinates.

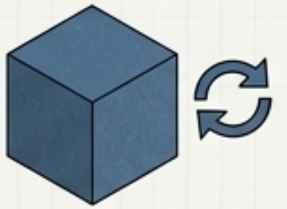
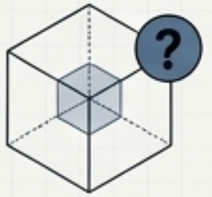
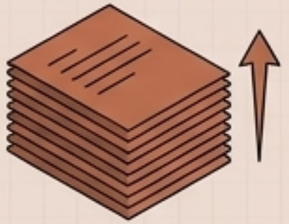
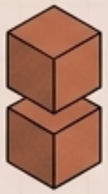
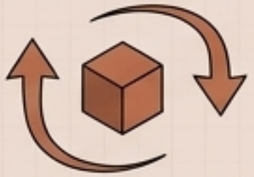
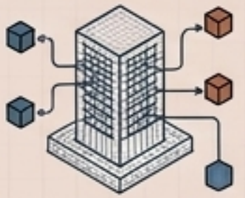
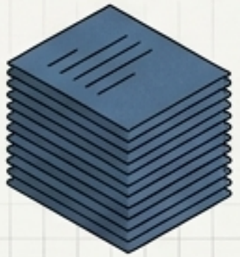
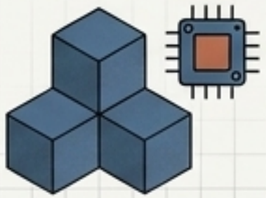
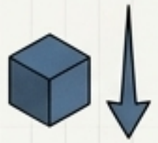
The Math

They use algorithms like **HNSW** and calculate **Dot Product** or **Cosine Similarity** to find the closest matches to a user's query.

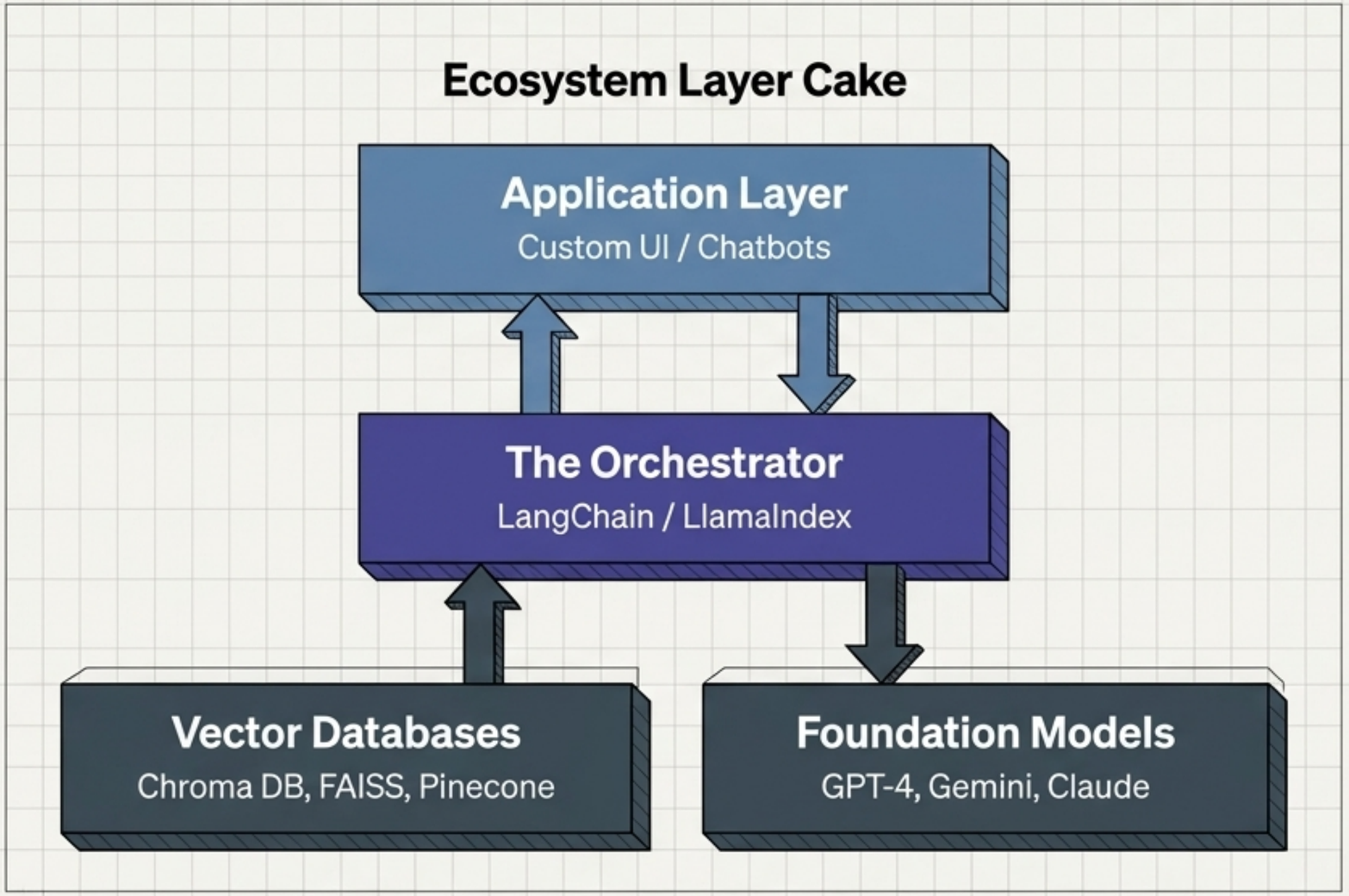
The Result

If a user searches for **'dog'**, the system retrieves documents about **'puppies'** and **'canines'** because their vectors are mathematically clustered together.

The AI Strategy Diagnostic Matrix

Method	Dataset Size	Initial Dev Cost	Ongoing Operating Cost	Best Use Case
Structured Prompts	Small (~300 pages max) 	Low 	High (repeated data input) 	Prototyping 
Sweet Spot: **Retrieval-Augmented Generation (RAG)**	Massive / Scalable 	Medium 	Low / Optimized 	Enterprise Knowledge Bases 
Fine-Tuning	Massive 	Very High (requires GPUs) 	Lowest 	Deep behavioral customization 

The Orchestration Layer: Tying It All Together

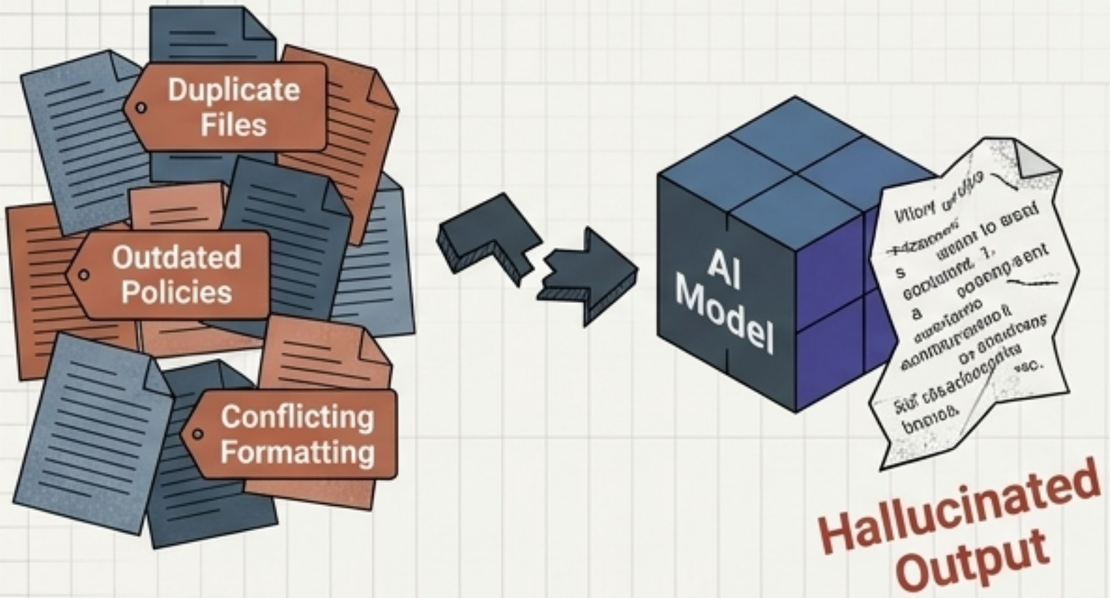


Frameworks like LangChain provide the vital control logic. They act as the middleware that manages the prompt templates, chains the retrieval events to the vector store, and formats the augmented data for the generative model.

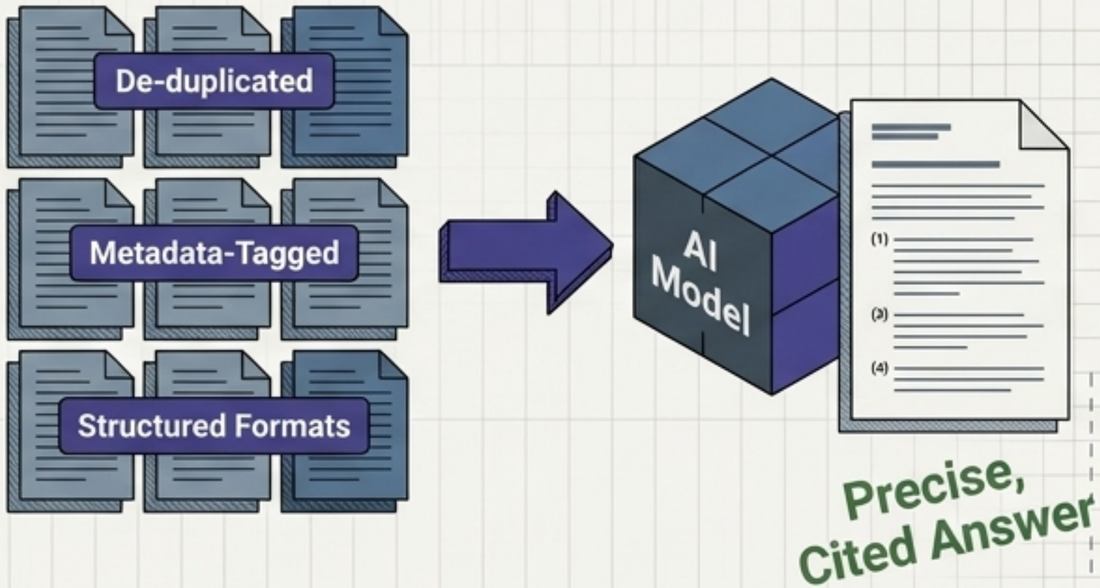
The Data Hygiene Imperative

"80% of RAG efforts are spent trying to get data foundations in order."

Garbage In



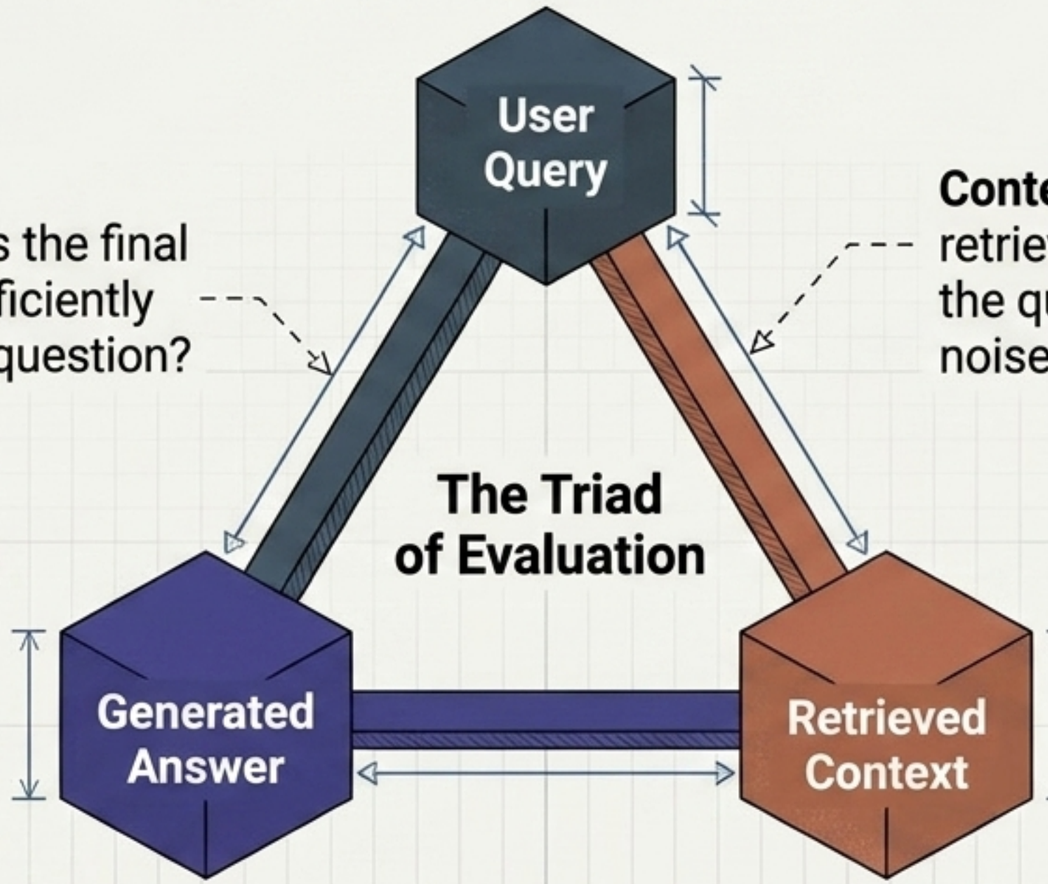
Quality In



A sophisticated retrieval engine cannot fix contradictory, outdated, or messy source documents.

Evaluating RAG Quality: The RAGAS Framework

The Triad of Evaluation



Answer Relevancy: Does the final answer directly and efficiently resolve the user's initial question?

Context Relevancy: Does the retrieved data directly address the query without redundant noise?

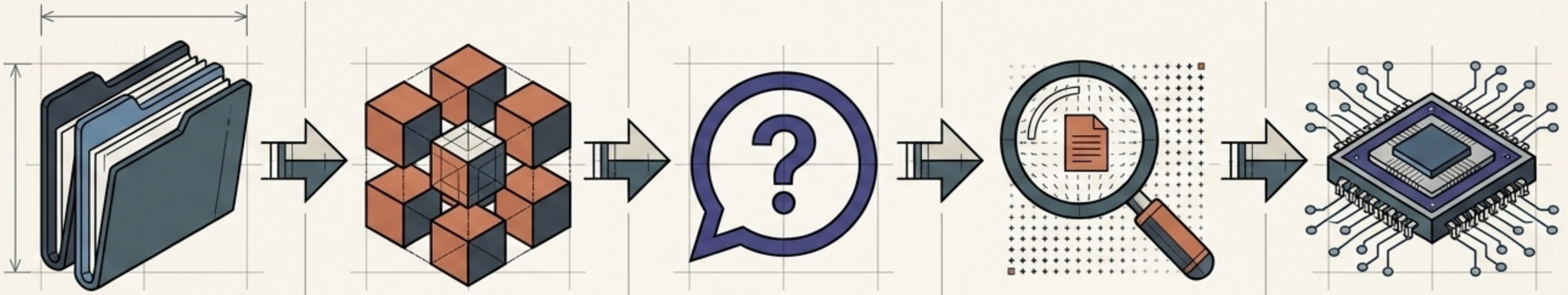
Faithfulness: Is the final answer factually consistent with the retrieved context (zero hallucinations)?

The Triad of Evaluation

System Pitfalls & Optimization Strategies

Symptom	Root Cause	Solution
System returns completely irrelevant answers.	Embedding mismatch (using different models for documents and queries).	Unify embedding models and ensure re-embedding after document updates.
AI misses crucial details in long documents.	Poor chunking strategy (chunks are too large, diluting semantic meaning).	Reduce chunk size and apply overlap to preserve context boundaries.
Accurate data is retrieved but ignored by the LLM.	"Lost in the Middle" phenomenon (LLMs prioritize the beginning and end of prompts).	Implement re-ranking algorithms to place the highest-scoring chunks at the top of the context window.

Real-World Architecture: The Maizey AI Blueprint



1. Canvas Course Data
(Syllabi, readings, lectures)

2. Vectorization
(Creating the knowledge base)

3. Student Query
("What is the late policy?")

4. Retrieval Engine
(Finding the exact syllabus clause)

5. GPT-4 API
(Drafting the response)

Maizey AI perfectly mimics the thought process of a human Teaching Assistant—relying on a specific, bounded memory of course materials to provide highly relevant, hallucination-free answers to students.

Enterprise Applications Across Domains

Financial Advisory

Data Source: Market reports, SEC filings, regulatory changes.

Use Case: AI assistant retrieves the latest compliance rules to advise on retirement account changes.

Healthcare

Data Source: Patient histories, clinical guidelines, research papers.

Use Case: AI provides doctors with concise, evidence-based answers to specific clinical questions.

Industrial Operations

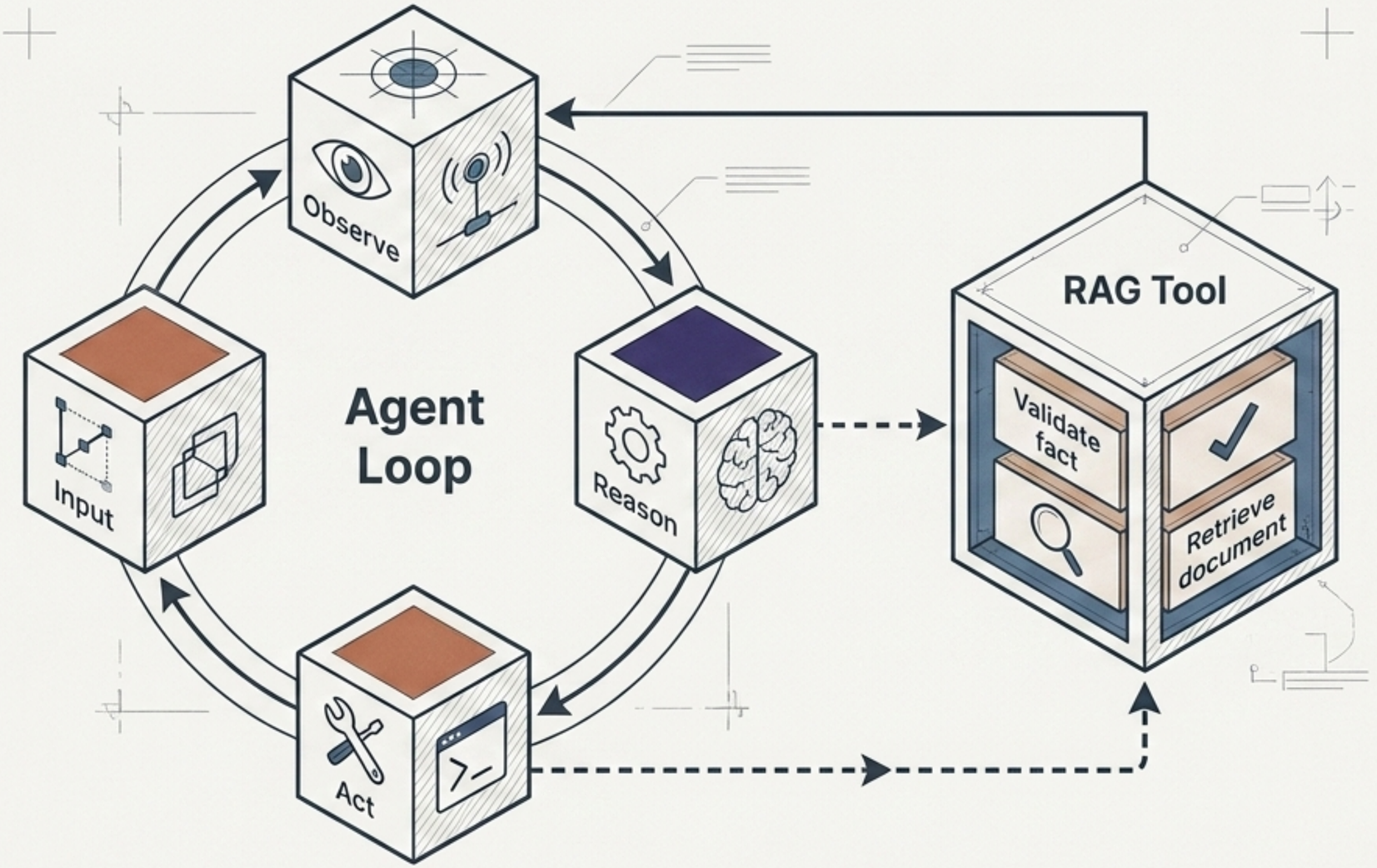
Data Source: Equipment manuals, maintenance logs, safety protocols.

Use Case: AI helps plant operators analyze operational data and suggest rapid maintenance actions.

The Next Evolution: Agentic RAG

RAG is evolving from reactive to proactive.

In an Agentic architecture (using the ReAct framework), the AI doesn't just answer prompts. It plans multi-step tasks, dynamically routes queries, and uses RAG as a 'skill' to fetch and validate data autonomously until a complex goal is achieved.

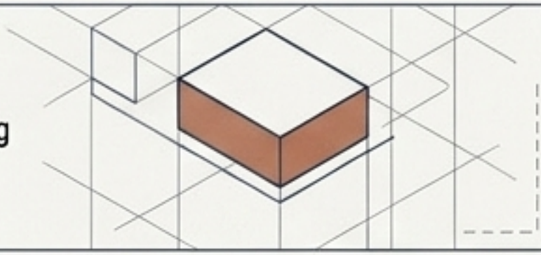


The RAG Implementation Blueprint

1. Data Foundation

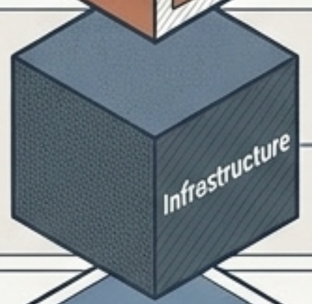
Audit data quality, establish metadata tagging, and ensure continuous deduplication.

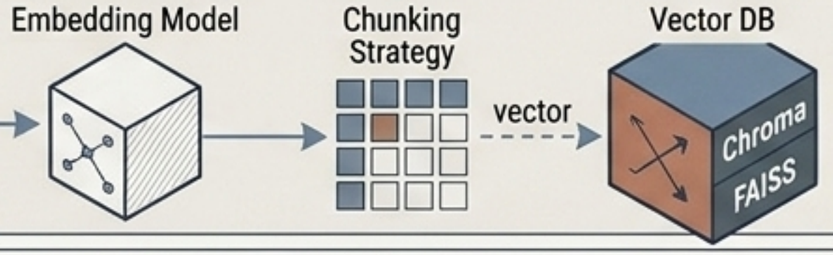
- Quality Audit
- Metadata Tagging
- Deduplication



2. Infrastructure Setup

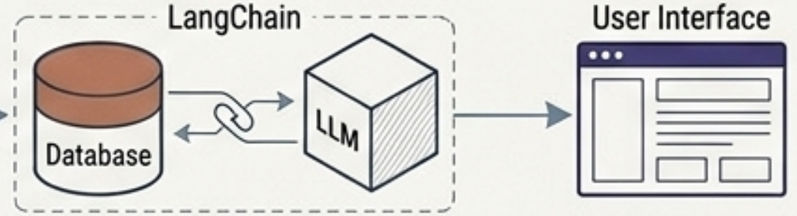
Select an embedding model, optimize chunking strategy, and deploy a scalable Vector Database (e.g., Chroma, FAISS).





3. Orchestration & UI

Utilize frameworks like LangChain to connect databases to LLMs and build a seamless user interface.



4. Continuous Evaluation

Implement the RAGAS framework to rigorously monitor Faithfulness and Context Relevancy before full production rollout.