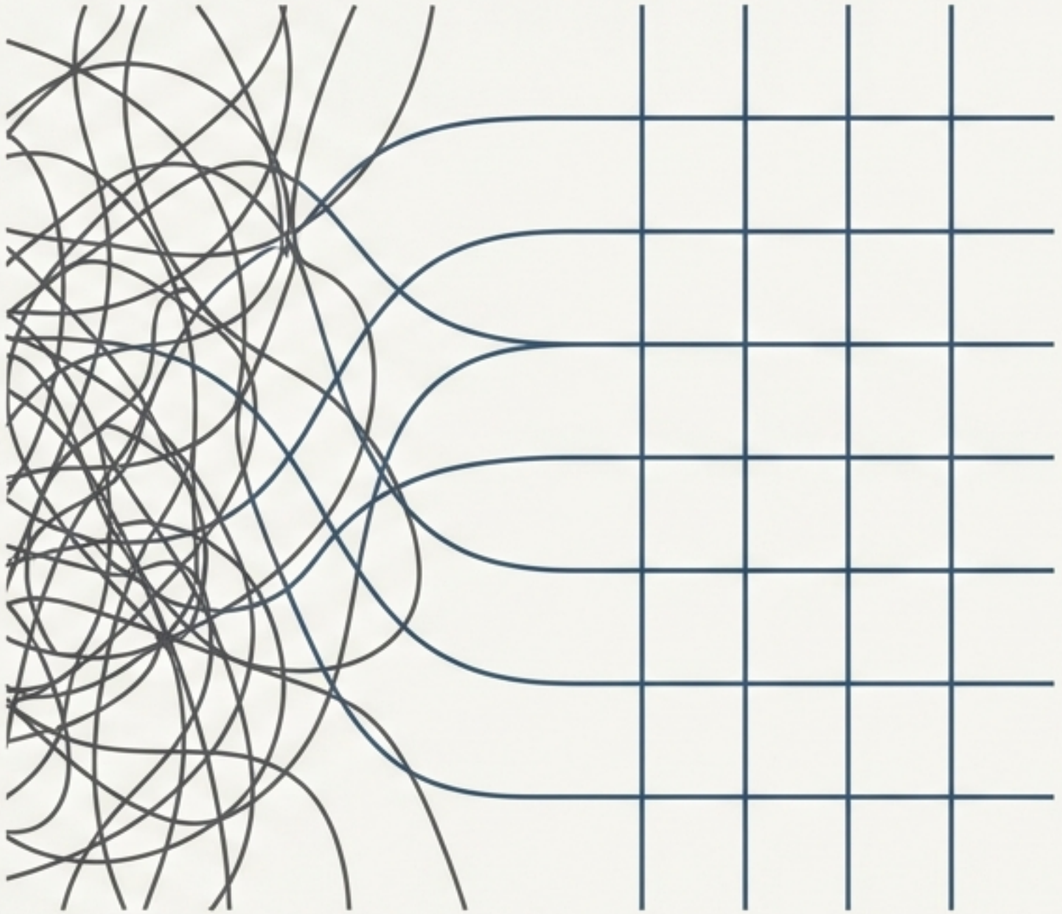
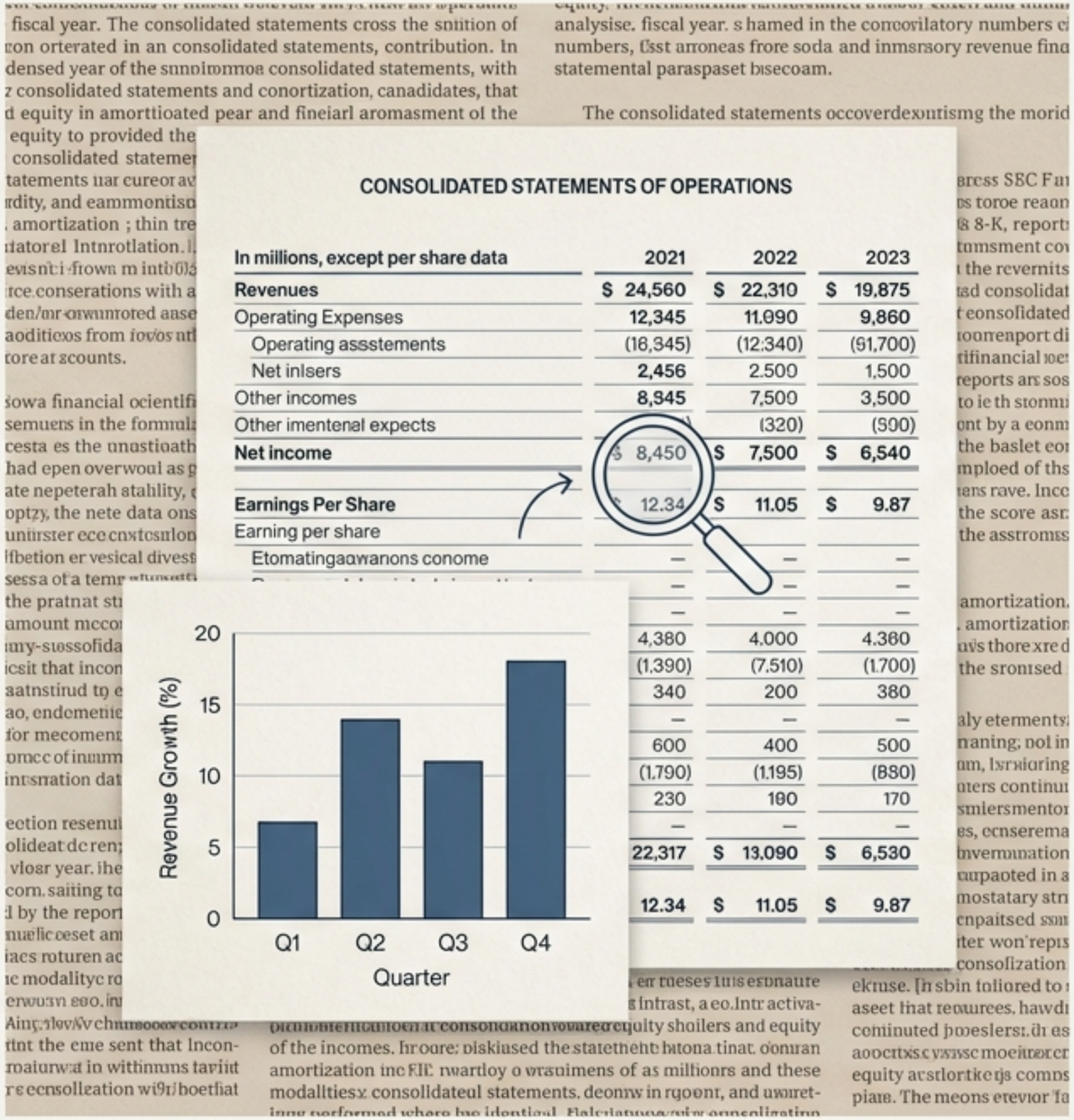
From Chaos to Clarity

Architecting a Deep RAG System for Financial Intelligence

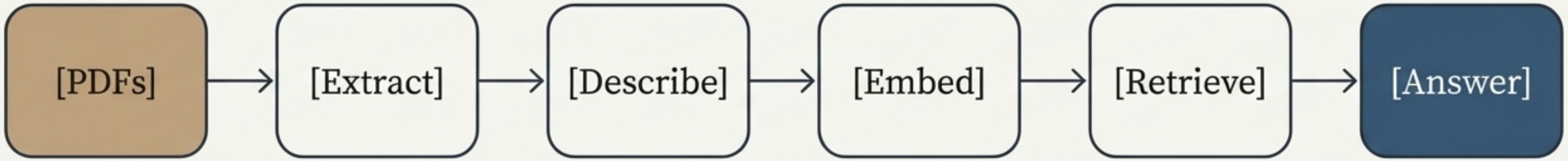


The Challenge: Unlocking Value from Dense Financial Filings

Financial documents like 10-K, 10-Q, and 8-K reports are rich with critical data. However, this information is locked within a complex mix of unstructured text, dense tables, and intricate charts. Extracting precise, actionable insights at scale requires a system that can understand and process all of these modalities seamlessly.



Our Blueprint for Transformation



Everything is Text,
Everything is Searchable.

Our architecture is built on a powerful principle: convert all content types—text, tables, and visuals—into a unified text-based format. This allows us to create a single, powerful, and searchable library of information.

Step 1: Deconstructing the Source Document

We begin by parsing each PDF using the *Docling* converter. The goal is to intelligently break down the monolithic document into its fundamental components while preserving crucial context.

Markdown*: Extracts the full document text, preserving page breaks for context.

Tables*: Captures tables as structured markdown, including two paragraphs of surrounding text for context and the original page number.

Images*: Isolates charts and diagrams larger than 500x500 pixels for dedicated analysis.

Key Feature: Page-level tracking is maintained for all extracted content, ensuring every piece of data can be traced back to its source.

Step 2: Translating Visuals into a Universal Language

To make images searchable alongside text, we translate their visual language into descriptive text. Each extracted image is processed by a Vision AI model to generate a rich, detailed description.

Model Used

Gemini 2.0 Flash Vision

AI Prompt Focus

- Chart/graph data trends and axis labels
- Chart/graph data trends and axis labels
- Table structures and key data points
- Summaries of embedded text content
- Descriptions of the visual layout

Why do this?

To create a Unified Embedding Space. By converting visual content to text, we enable a single, text-only embedding model to index every piece of information in our system.

Step 3: Building a Universal, Searchable Library

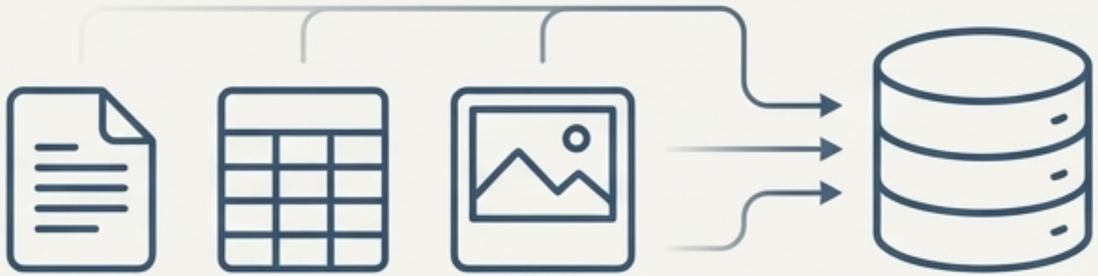
With all content converted to text, we ingest it into a single Qdrant vector collection. We use a hybrid indexing strategy to combine the best of semantic understanding and keyword precision.

Components Table

Component	Technology	Purpose
Vector Store	Qdrant	Storage with `RetrievalMode.HYBRID` enabled
Dense Embeddings	Gemini Embedding 001	Semantic understanding (768 dimensions)
Sparse Embeddings	FastEmbed BM25	Keyword matching

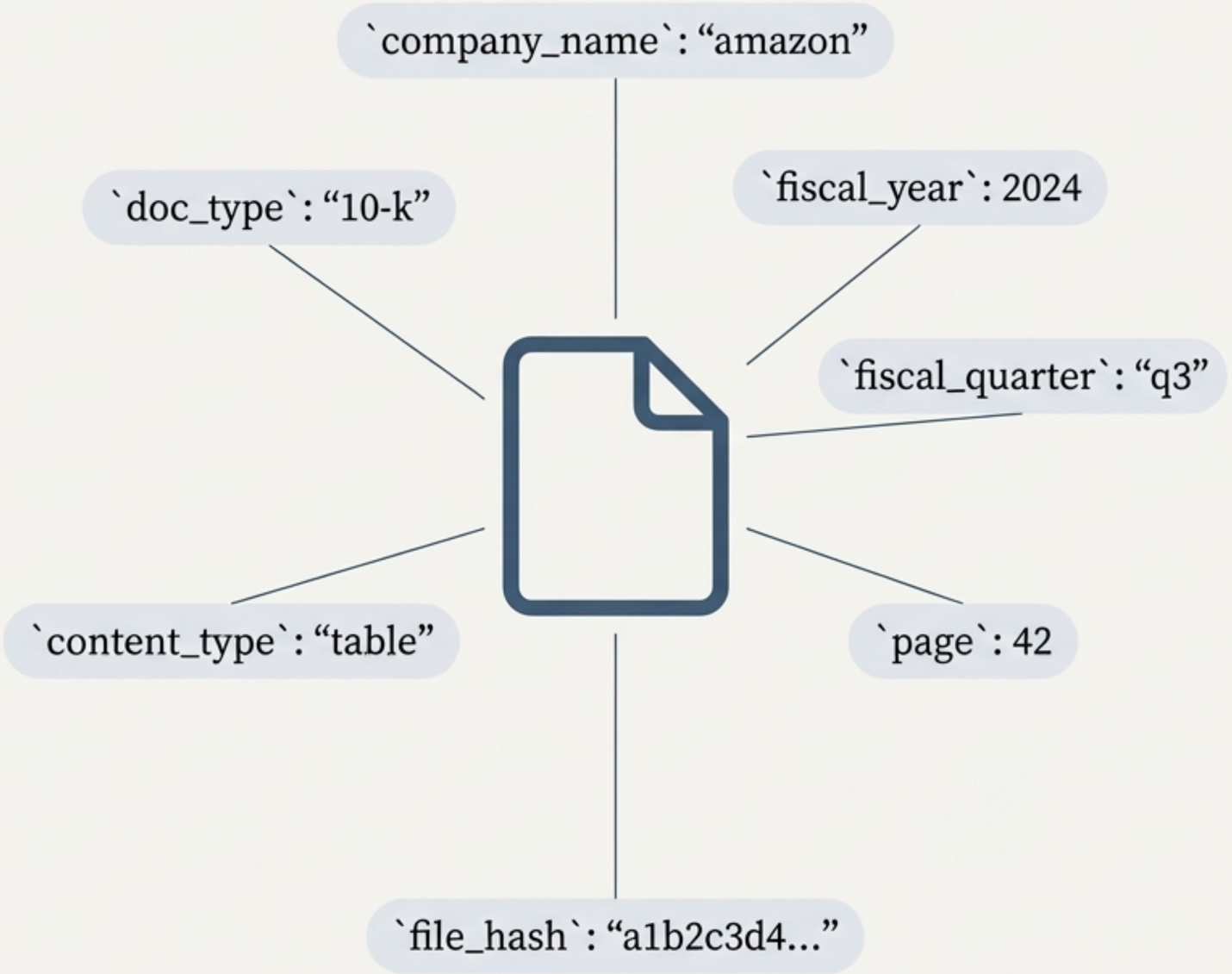
Chunking Strategy

- **Text:** Split by page breaks (`` markers).
- **Tables & Images:** Ingested as individual documents (no splitting) to preserve their complete context.



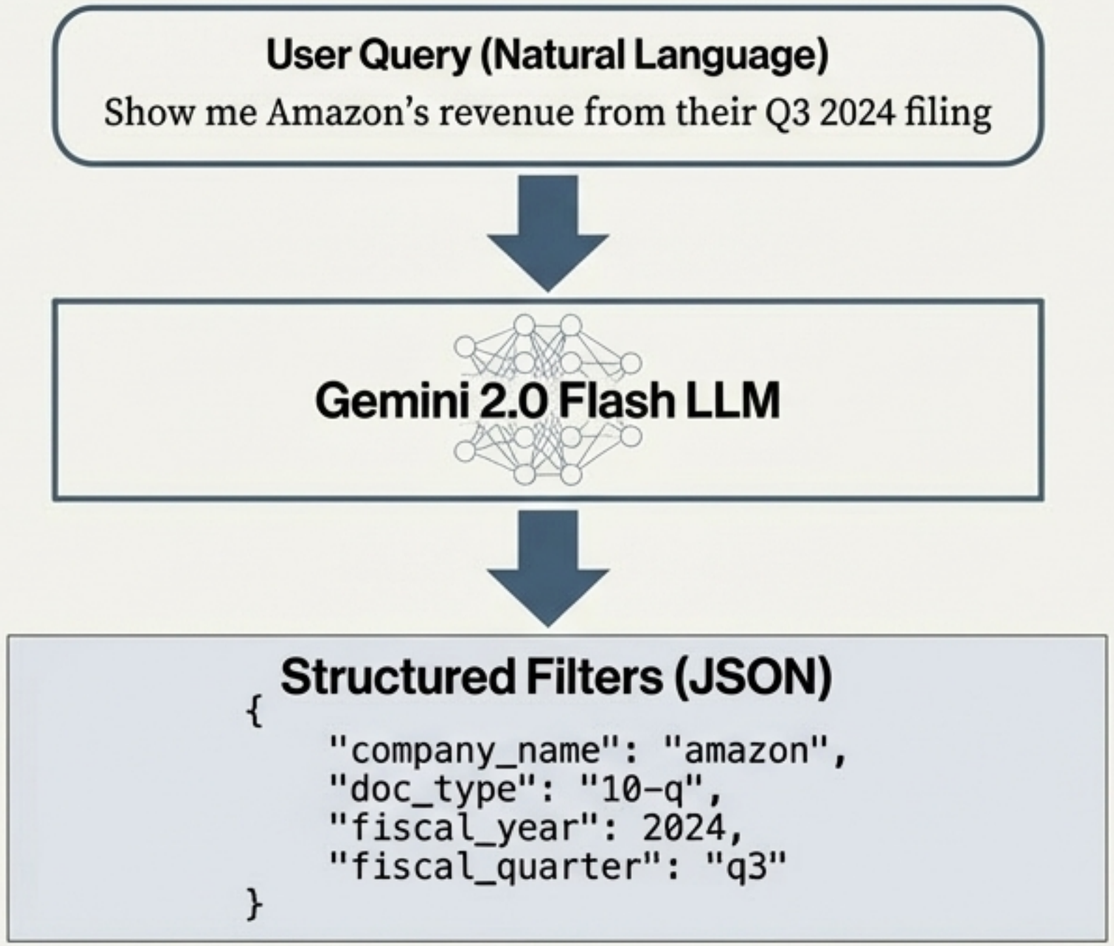
The Card Catalog: Powering Precision with Metadata

Every document chunk is enriched with a comprehensive set of metadata extracted from its filename and content type. This structured data is the key to narrowing the search space *before* performing spare semantic search.



Step 4: The Intelligent Retrieval Engine

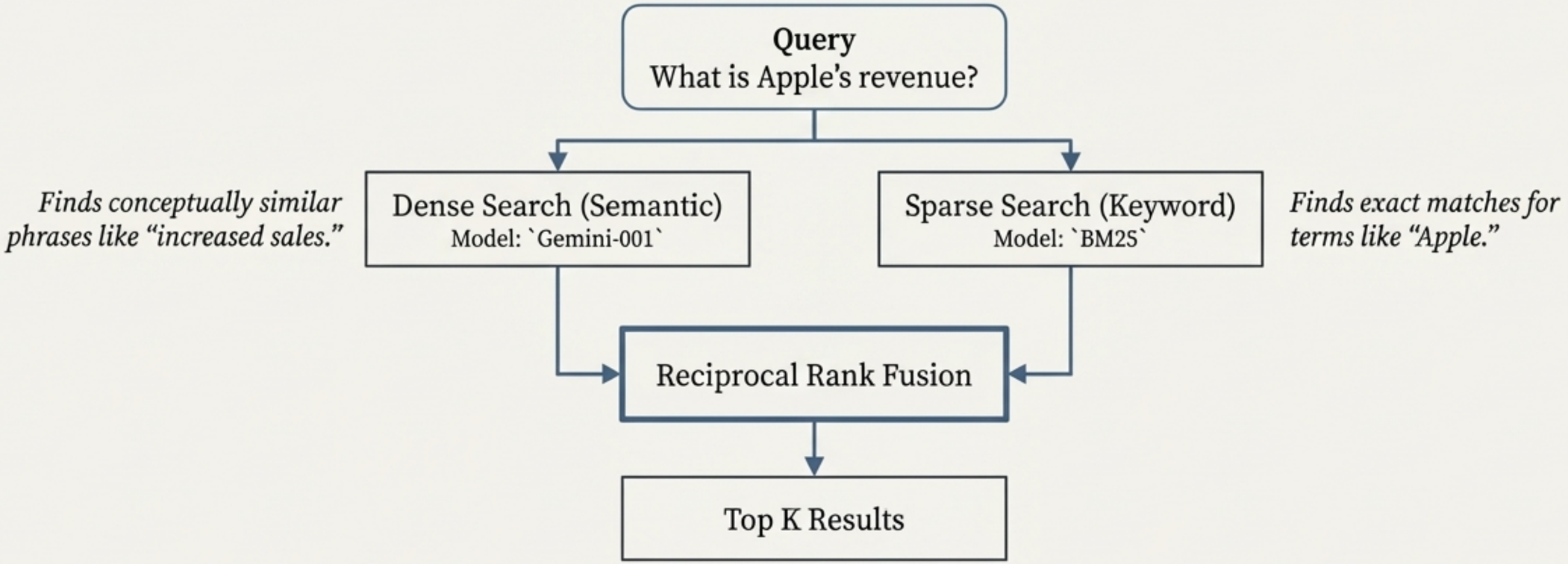
Part I: Translating Intent with LLM-Powered Filtering



Key Concept: The first step in our retrieval pipeline isn't search; it's understanding. An LLM parses the user's query to extract structured metadata filters, dramatically reducing the search space to only the most relevant documents.

The Intelligent Retrieval Engine

Part II: Fusing Meaning and Keywords with Hybrid Search



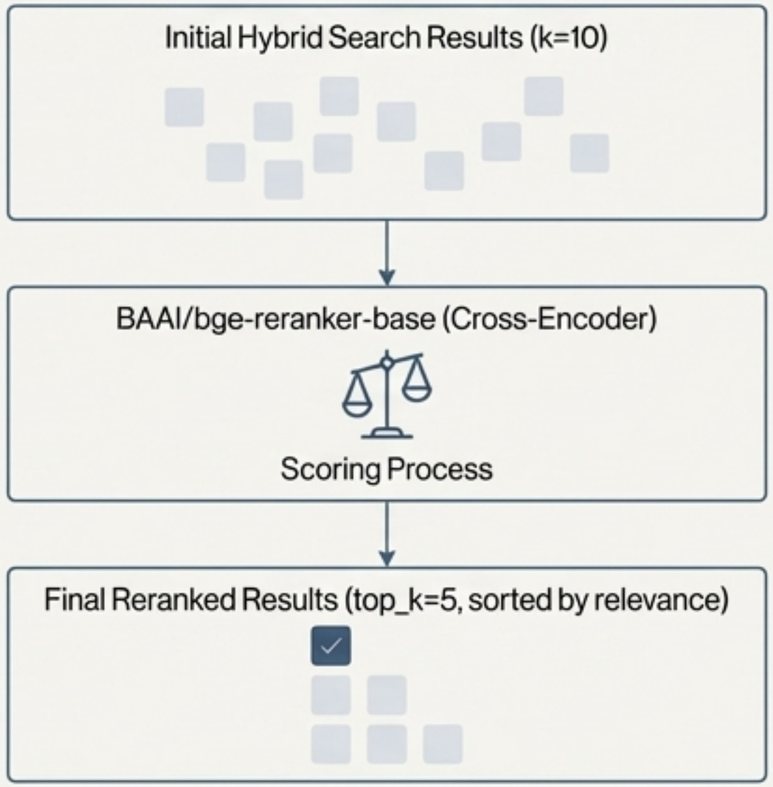
****Why Hybrid?****

Combining dense search for semantic understanding with sparse search for keyword precision provides the most comprehensive and relevant initial candidate set.

The Intelligent Retrieval Engine

Part III: Ensuring Precision with Cross-Encoder Reranking

After the initial hybrid search, a cross-encoder model performs a deeper, more computationally intensive analysis. It directly compares the user's query against each candidate document to calculate a final, precise relevance score.



Purpose: This deep interaction between query and document is critical for eliminating 'near miss' results and promoting the most contextually relevant information to the top.

Our Architectural Pillars: The Rationale Behind the Design

Unified Text Embeddings

Decision:
Use a single text embedding model (Gemini-001) for all content.

Benefits:
Simplified architecture, unified search, cost-effective.

Alternative Rejected:
Multimodal embeddings were avoided due to API issues and added complexity.

Hybrid Search (Dense + Sparse)

Decision:
Combine semantic and keyword search.

Benefits:
Captures both conceptual meaning ('revenue growth' → 'increased sales') and specific terms ('Q3 2024'). The result is more comprehensive retrieval.

Metadata-Driven Filtering

Decision:
Use an LLM to extract structured filters from the query.

Benefits:
Dramatically reduces the search space before vector search begins, increasing speed and relevance. It's more precise than embedding metadata directly in text.

The Toolkit: Our Technology Stack

Component	Technology	Purpose
PDF Extraction	Docling	Convert PDFs to structured content
Vision	Gemini 2.0 Flash	Generate image descriptions
Embeddings	Gemini Embedding 001	Dense semantic vectors (768D)
Sparse	FastEmbed BM25	Keyword matching
Vector DB	Qdrant	Hybrid search storage
Framework	LangChain	Abstraction layer for clean code
Reranker	BAAI/bge-reranker-base	Cross-encoder reranking
LLM	Gemini 2.0 Flash	Filter extraction & Q&A

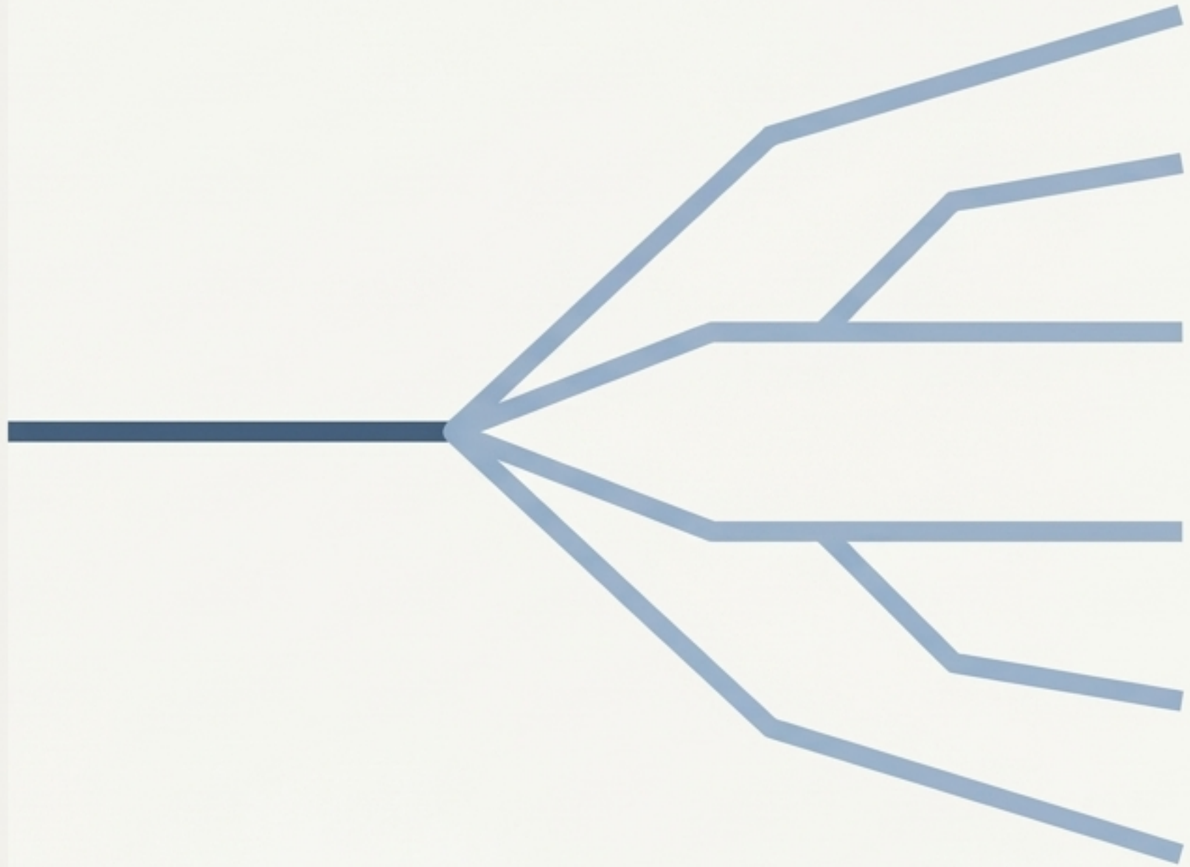
Key Learning Outcomes & Production-Ready Patterns

- **A Complete End-to-End Pipeline:** A reproducible pattern for transforming raw PDFs into queryable answers.
- **Practical Multimodal Handling:** A robust method for unifying text, tables, and images under a single search index.
- **Advanced Retrieval is a Multi-Stage Process:** The combination of filtering, hybrid search, and reranking delivers state-of-the-art performance.
- **Clean Code via Abstraction:** Using frameworks like LangChain simplifies interaction with complex components like vector databases.
- **Scalability by Design:** The architecture supports incremental ingestion and uses deduplication to handle large volumes of documents efficiently.

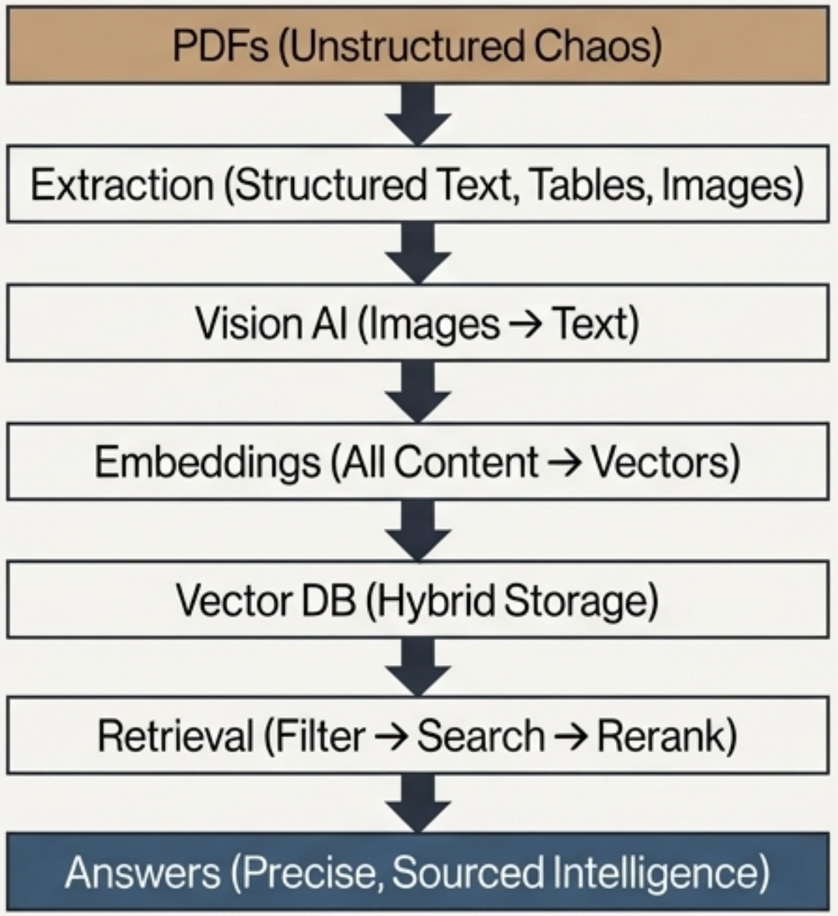
The Horizon: Future Architectural Enhancements

This architecture provides a strong foundation. Future iterations could incorporate several advanced techniques to further improve precision and capability.

1. **Parent-Child Retrieval:** Link smaller chunks back to their larger parent documents for better context in the final answer.
2. **Multi-Query Retrieval:** Generate several variations of a user's query to broaden the search and catch different phrasings.
3. **Graph-Based RAG:** Model entities and relationships within the documents to answer more complex, multi-hop questions.
4. **Contextual Compression:** Use an LLM to filter out irrelevant sentences from retrieved chunks before they are sent to the final model.



From Raw Data to Refined Answers: The Complete Journey



By treating **everything as text** and making **everything searchable**,
we create a simple yet powerful RAG system that works reliably at scale.