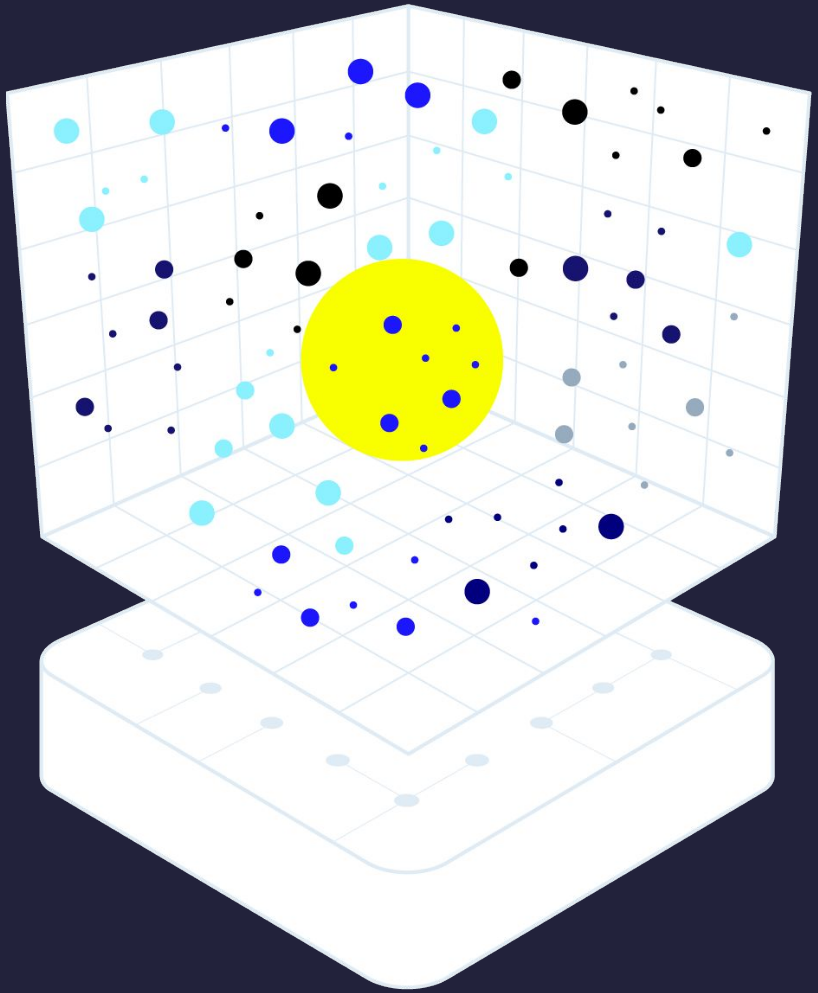


Pinecone

Retrieval Augmented Generation workshop

Roie Schwaber-Cohen, Staff Developer Advocate (Pinecone)

Quick Refresher

Contextualized Meaning

Ground LLMs

- LLMs don’t know anything about **our** data.
- Consider LLMs as a **Natural Language Interface** or a reasoning engine instead of the source of truth.
- We query our knowledge based on the user’s prompt to retrieve content ***we*** consider **relevant**.
- We inject the relevant content into the **context window** of the LLM as the basis for future responses.

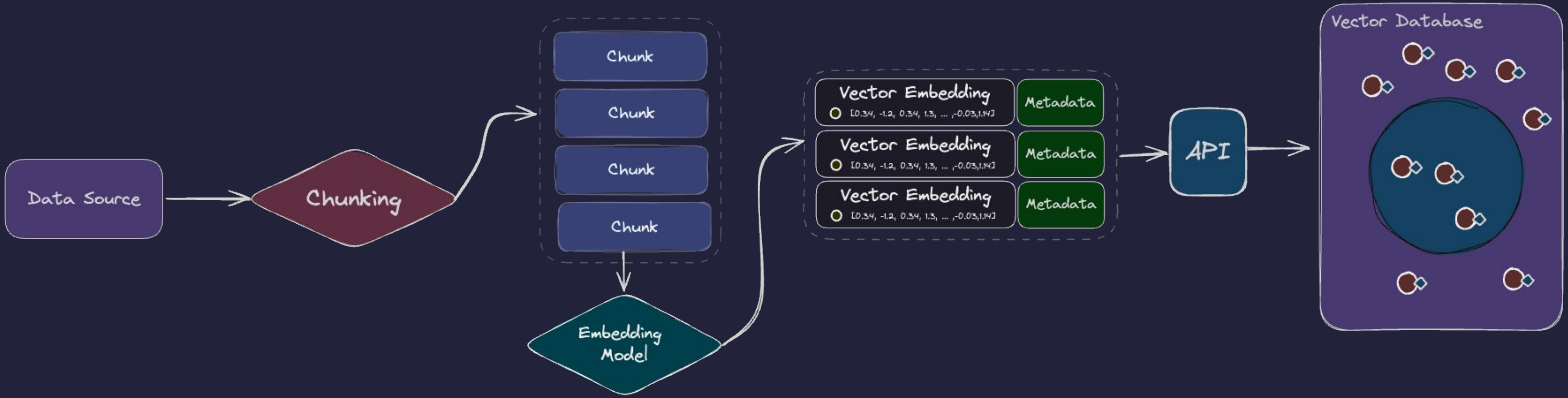
Retrieval Augmented Generated (RAG)

RAG with a vector database

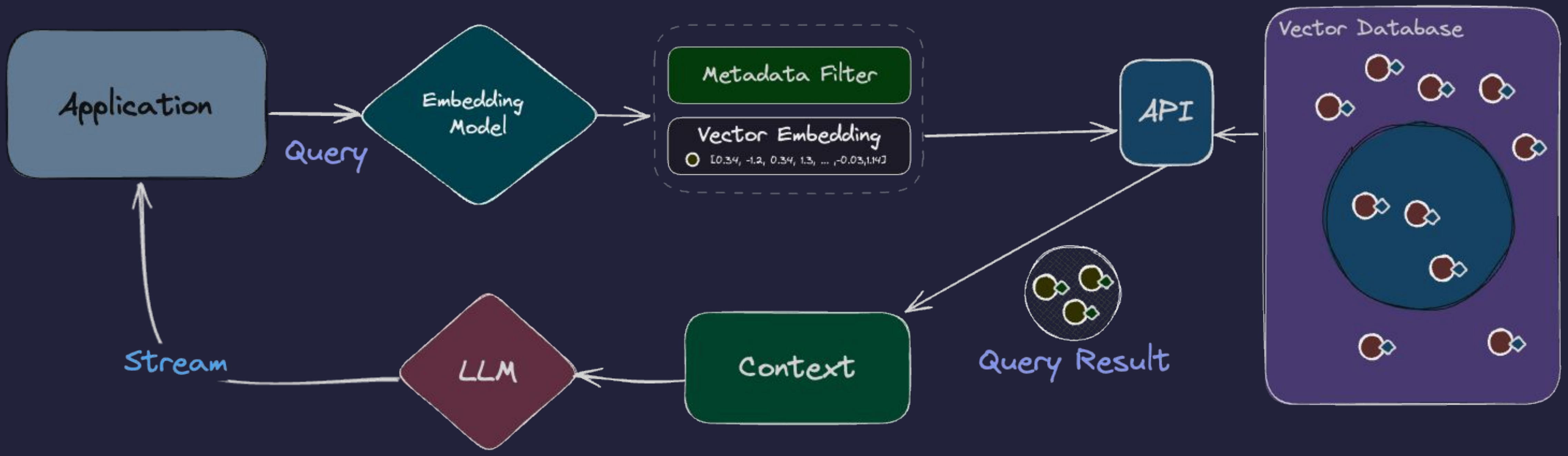
- The user prompt is likely to be semantically ambiguous.
- We can use embeddings models to extract the semantic meaning from the user prompt, and to match it to data we care about.
- We embed our knowledge base which then allows us to query inject semantically relevant content into the context window.
- We can teach the LLM to say “I don’t know”: Using the similarity score, we can filter out responses that don’t pass a given threshold.
- We can leverage metadata to improve relevance and performance

Architecture

RAG Architecture – Ingestion



RAG Architecture - Application



Things to consider

Chunking strategies

- What are you indexing?
- What embedding model are you using?
- Relation to query and retrieved data
- “Content-aware” vs programmatic
- Applying “traditional” NLU strategies like topic modeling, NER etc.

Using metadata

- Filtering:
- Context Enrichment: Adds extra layers of information for more nuanced responses.
- Ranking Boost: Uses metadata like credibility for better document selection.
- Domain Filtering: Allows targeted retrieval based on subject tags.
- Temporal Relevance: Utilizes timestamps for timely results.

Monitoring and evaluation

- Use systems such as TrueLens, Galileo, LangSmith and LlamaIndex to monitor the performance of your application
- Latency: Measure time from query to response.
- F1-Score: Evaluate accuracy on QA datasets.
- User Feedback: Compute user satisfaction index.