



How to Use Private Data in Generative AI: End-to-End Solution for RAG with CrateDB and LangChain

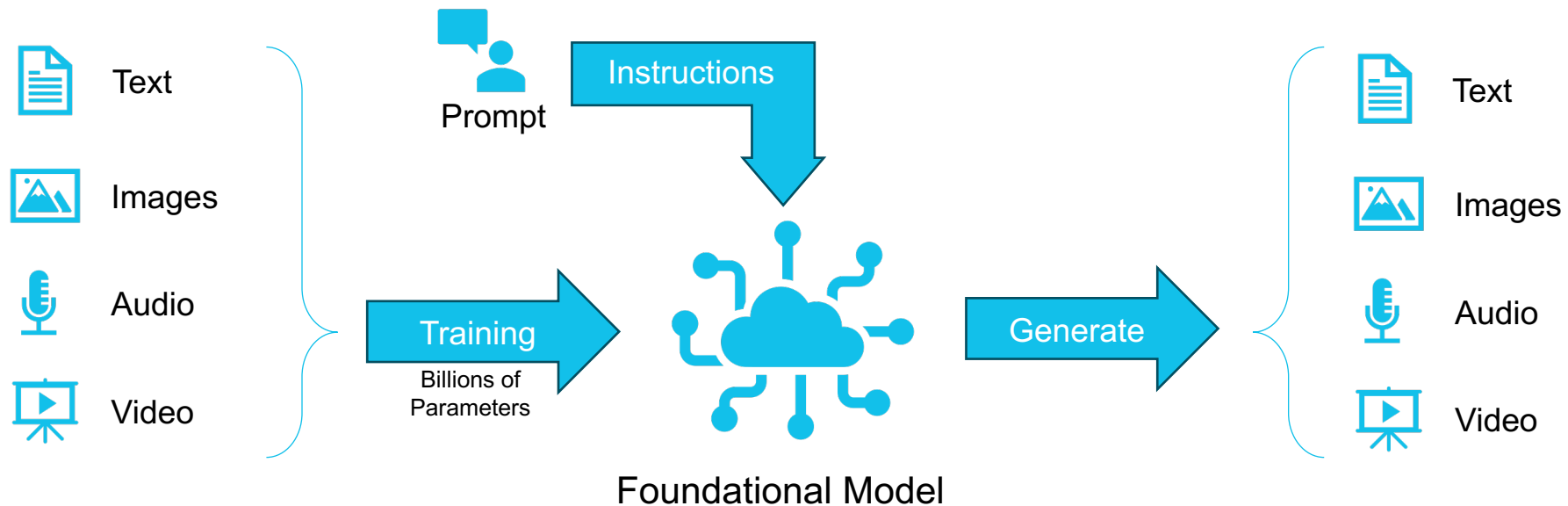
Marija Selakovic, Developer Advocate, CrateDB

Christian Kurze, VP Product, CrateDB

What is Generative AI?

Generative AI is a set of artificial intelligence methodologies that can produce novel content that resembles the training data they were exposed to.

The content could be anything spanning from synthesizing text, generating code, realistic images, music and more.



Huge Potential, but also Challenges of Generative AI



Quality & Reliability: Hallucinations, accuracy, timely input data

Ethical & Societal: Deepfakes, misinformation, bias in AI-generated content require policies and controls

Computational Costs & Environmental Impact: High power required to run large generative AI models

Intellectual Property & Copyright: Generated content resembles human-created work

Managing & Governing AI: Frameworks to manage the development and deployment of generative AI technologies

The Importance of Current, Accurate, Private Data



You

"I saw a promotion for a 20% discount on my next purchase in your store, but it's not applying at checkout. What's the promo code I need to use?"



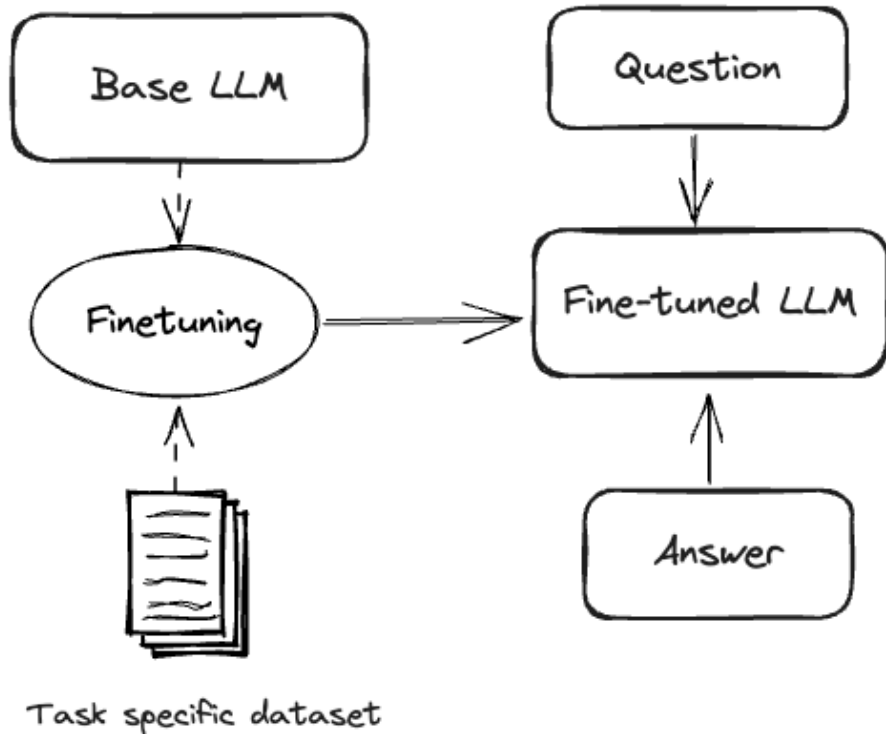
ChatGPT

I'm sorry for the confusion, but as an AI developed by OpenAI, I don't have access to specific promotional codes or the ability to assist with transactions for any specific stores or services. For issues related to promotions or discounts at a specific store, I recommend contacting the customer service department of that store directly. They would be the best resource to help you with the promo code and any issues you're experiencing at checkout.



- **Current & Accurate:** most recent information must be available for meaningful answers
- **Private data:** internal, confidential, sensitive, subject to privacy regulations
- Utilizing with LLMs:
 - Improves accuracy (less hallucinations)
 - Enhanced personalization (better user experience)
 - Richer data insights (documentation, support tickets, legal documents)

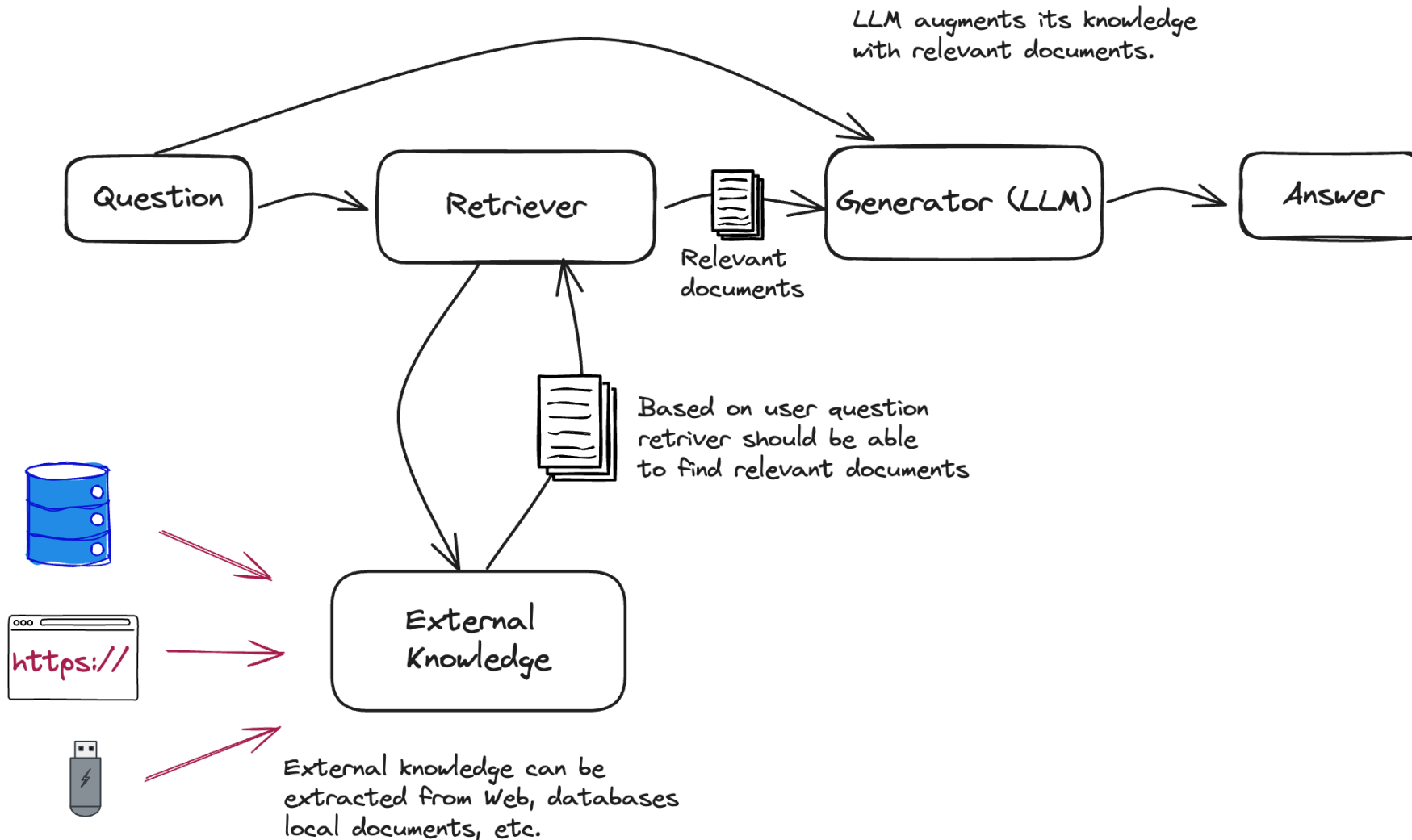
The AI Dilemma: Fine-tuning vs RAG



- **Advantages:**
 - Updates knowledge with domain-specific data
 - More cost-effective than full model retraining
- **Challenges:**
 - Still a need for frequent data update
 - Static knowledge (overfitting risk)
 - May still produce hallucinations
 - Resource intensive



Retrieval Augmented Generation (RAG)



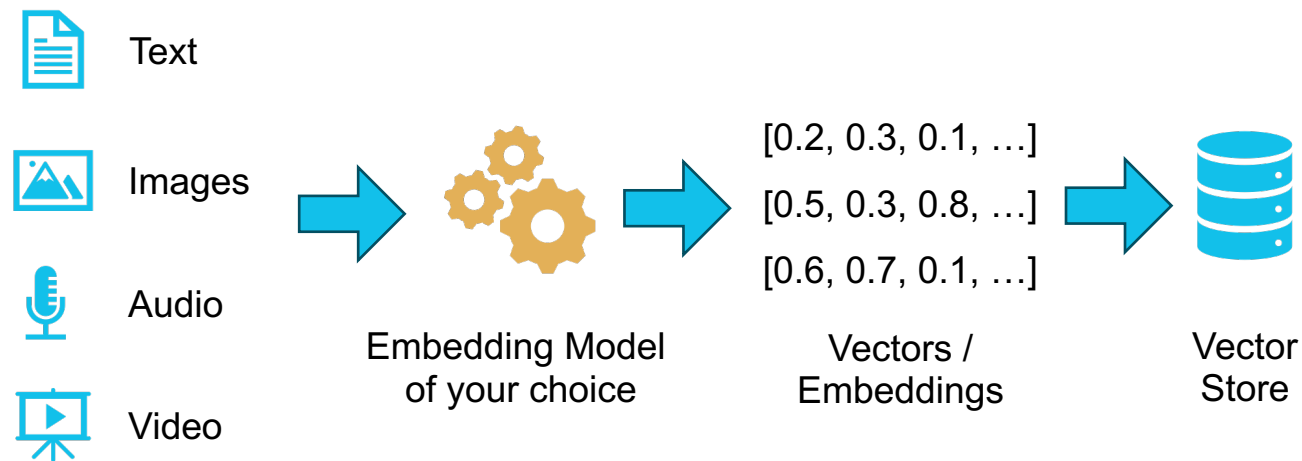


Benefits of RAG

- **Advantages:**
 - Access control, knowledge not incorporated into the LLM
 - Real-time data available
 - Reduced training needs
 - Flexibility when integrating with different data sources and formats
 - Flexibility in choosing embedding algorithms and LLMs
- **Challenges:**
 - Depends on the efficiency of the underlying search system
 - Limitations on the amount of context LLMs can consider
 - Hallucinations can be reduced, but still might happen

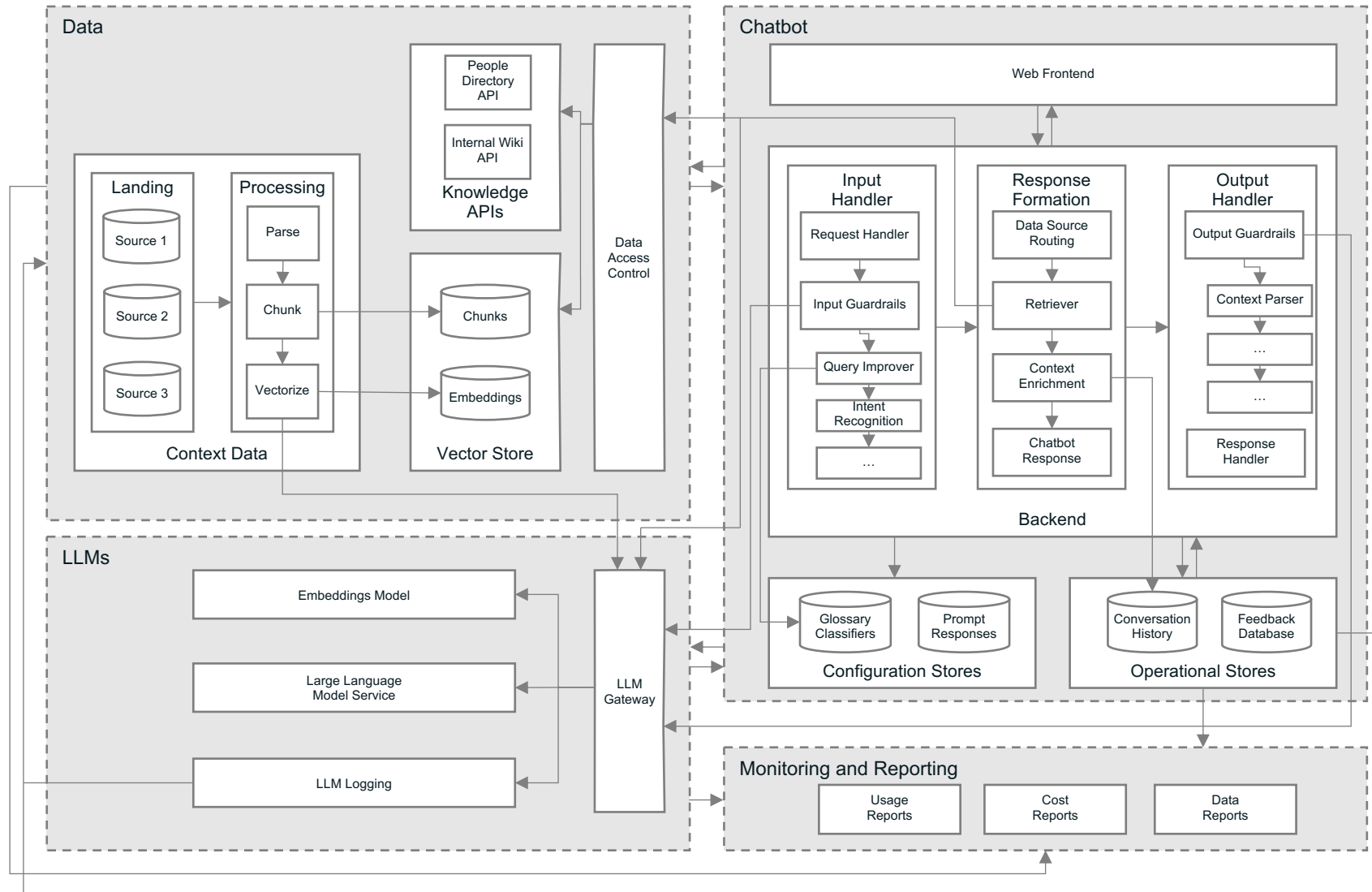
How is semantics of language captured? Vectors!

Vectors / Embeddings are numerical representation of data objects (like words, phrases, entire documents, images, audio, etc.) in a high-dimensional space. They enable systematic access to unstructured data like similarity search and therefore enable processing and understanding text in a mathematical form.



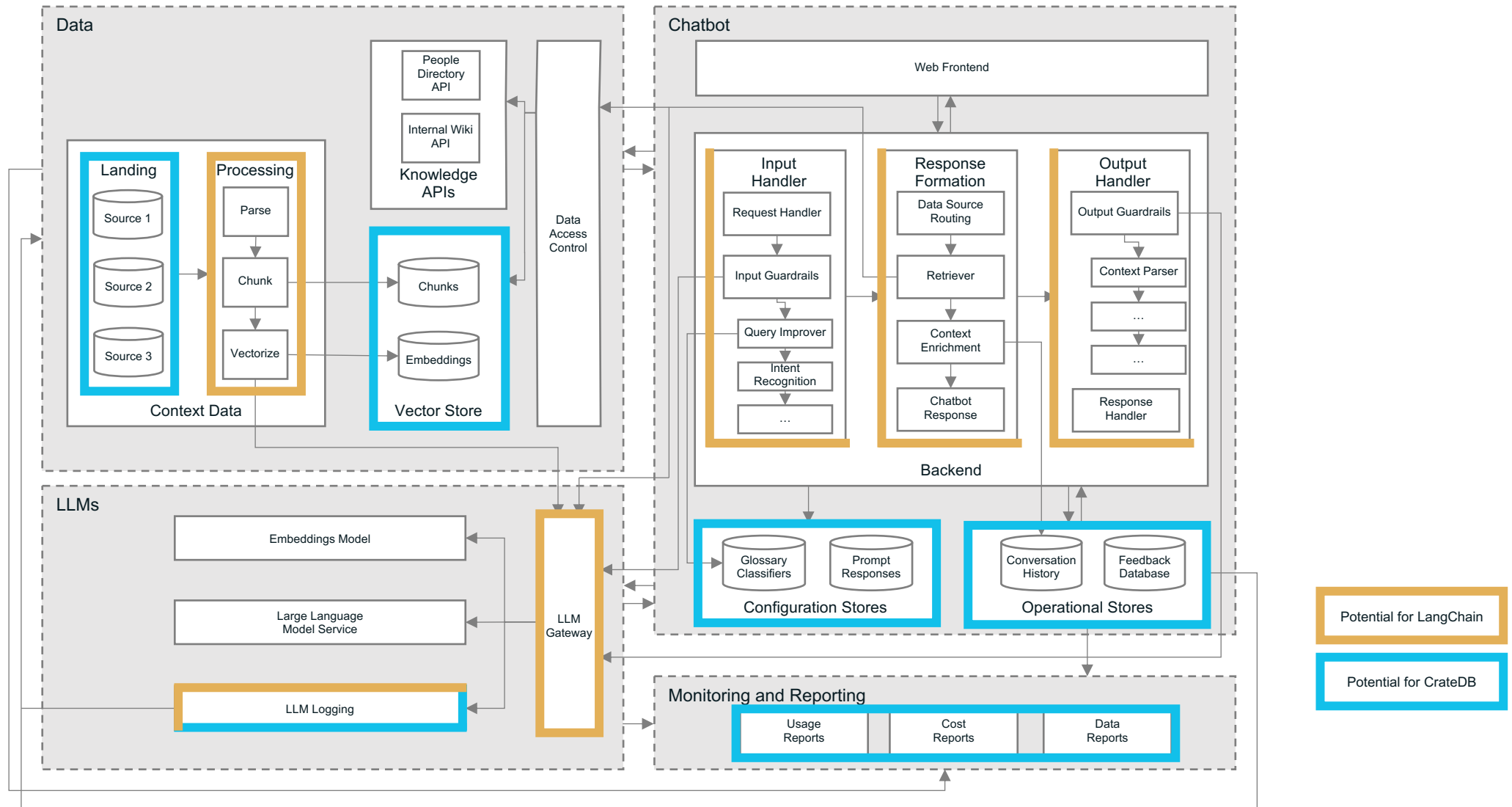


Knowledge Assistants - Architecture





Knowledge Assistants – Open Source 😊



Why CrateDB and LangChain?



- **Data comes in many formats:** structured, semi-structured, unstructured; while typical databases can only cope with one type of data and come with custom APIs
- 80% of data is **unstructured** (Gartner)
- Generative AI requires efficient data management, especially **contextualization**
- Foundational Models are **only trained on public data**
- **(Too?) many alternatives** regarding embedding models and LLMs

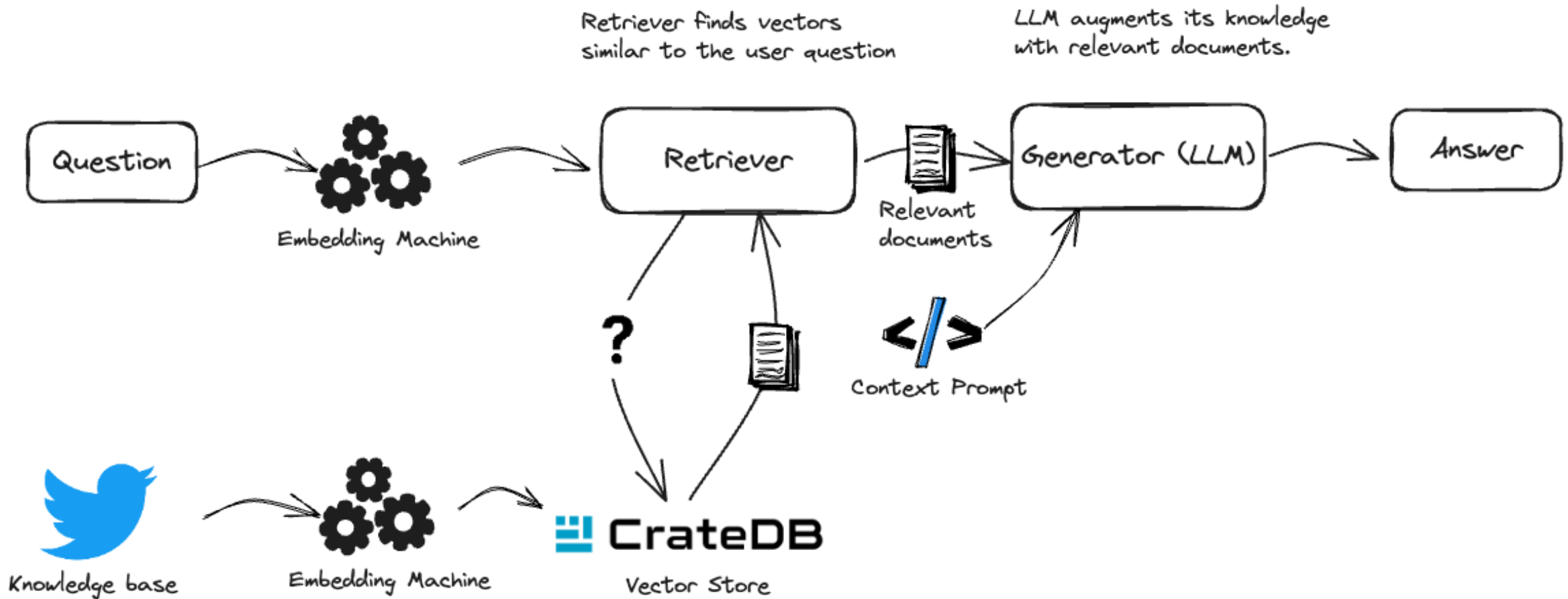
Robust data management: distributed, highly scalable database natively supporting tables, time-series, geospatial, full-text, vector; accessible via **standard SQL**

Comprehensive set of building blocks and swappable libraries to access models, vector stores, text splitters, output parsers, and pre-built chains; covering development, serving and observability; available in **Python and JavaScript**



Demo: Chat for Support Knowledge Base

CrateDB + LangChain for RAG



<https://github.com/crate/cratedb-examples/tree/main/topic/machine-learning/llm-langchain>

Demo



Get Started Today!

marija@crate.io

christian.kurze@crate.io

LangChain: langchain.com

LangChain Docs: <https://python.langchain.com/docs>

CrateDB Cloud: console.cratedb.com

CrateDB Community: community.cratedb.com

CrateDB Docs: docs.cratedb.com