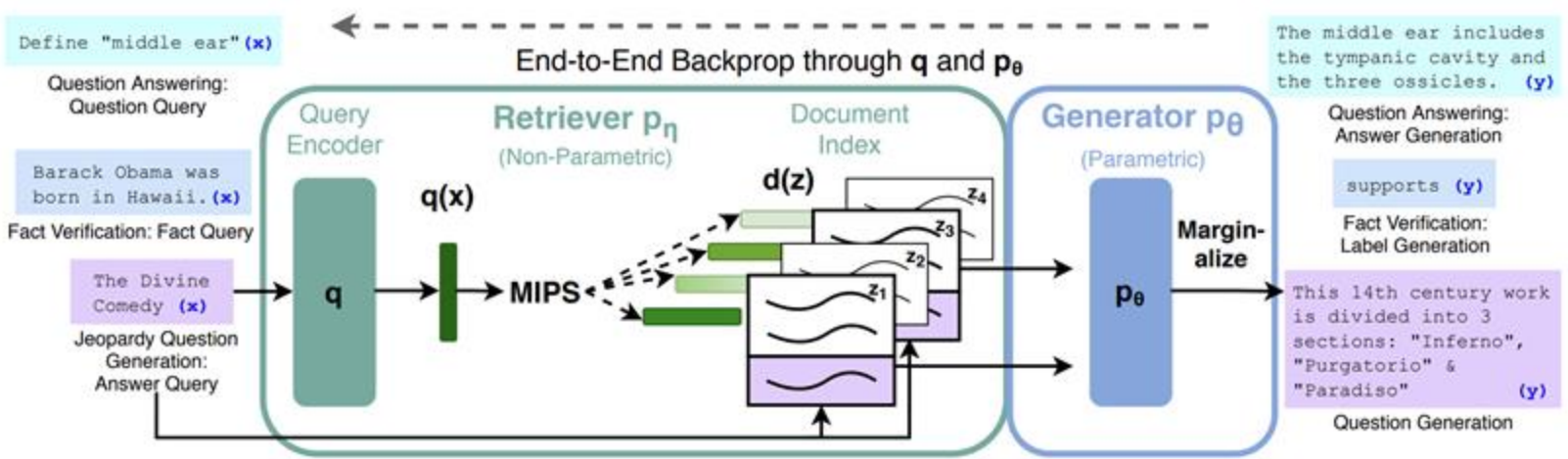
Summary of Previous Lecture

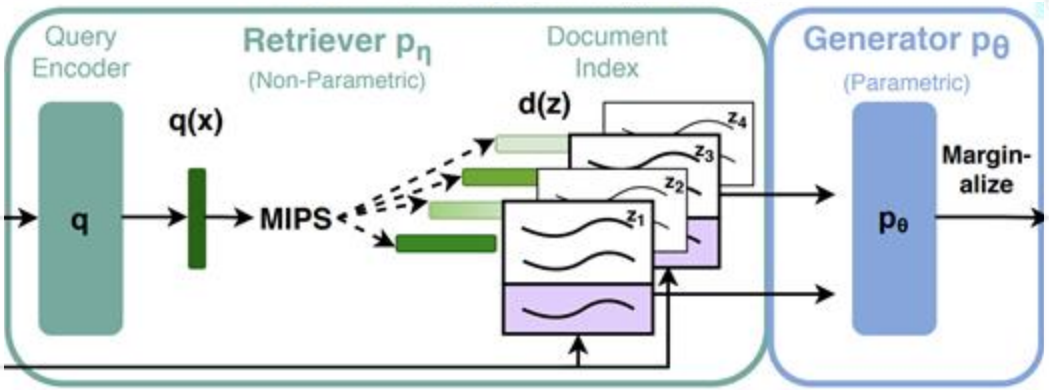
2025. 04. 09

Data Systems Lab

Overview of RAG



Component of RAG



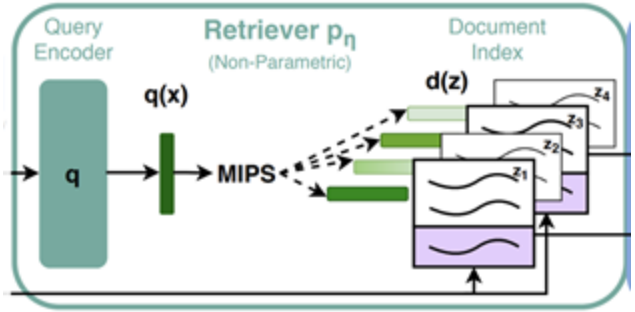
Retriever

Generator (Reader)

Previous Lecture Content

- ❑ 1-NN / k-NN algorithm
- ❑ Document representation
- ❑ Distance metric
- ❑ Approximate k-NN
- ❑ KD-tree
- ❑ Locality sensitive hashing

…



Retriever

A Tutorial on Retrieval-Augmented Generation

2025. 04. 09

Data Systems Lab

Overview

- Introduction
  - Practice Purpose
  - Retrieval Augmented Generation (RAG) & Practice Instruction
- Background
  - LlamaIndex
- Implementation Detail
  - How to Evaluate
  - Changeable Configuration of RAG

Overview

- Introduction
  - Practice Purpose
  - Retrieval Augmented Generation (RAG) & Practice Instruction
- Background
  - LlamaIndex
- Implementation Detail
  - How to Evaluate
  - Changeable Configuration of RAG

Practice Purpose

Understand the RAG.

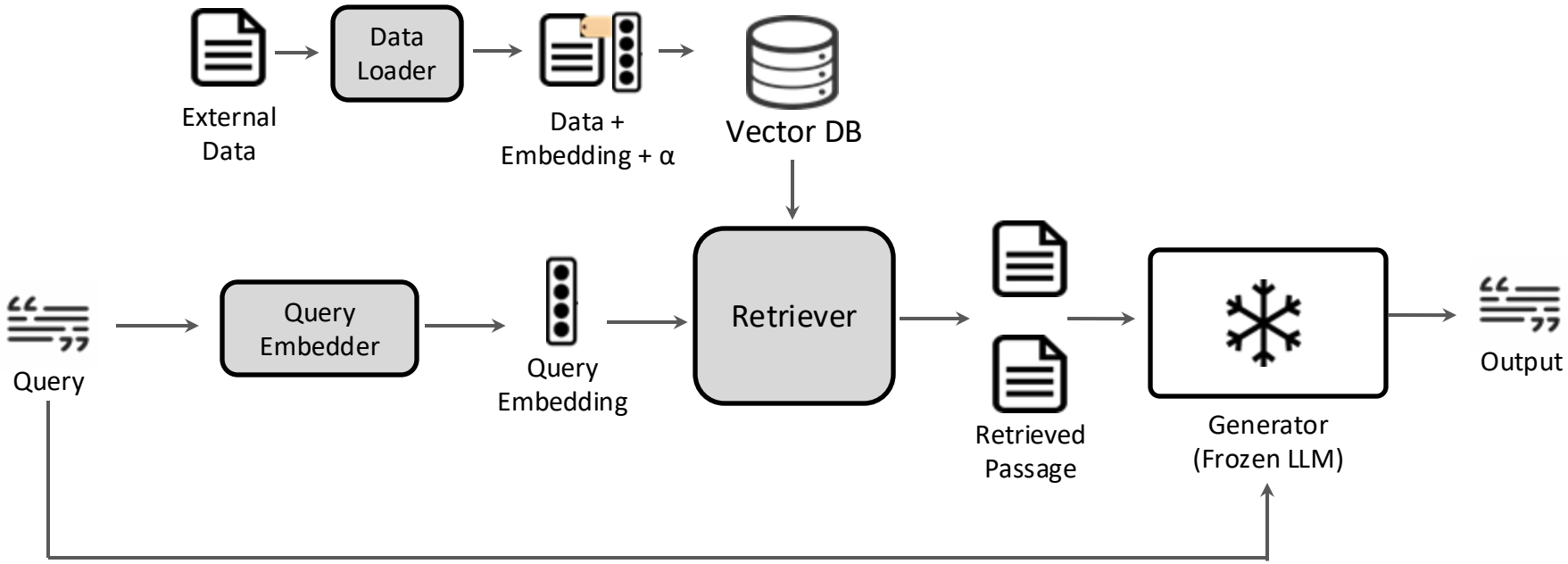
- ❏ Think about the role of retriever and generator, and see what is input and output of retriever and generator
- ❏ Learn how to connect between the database and query engine
- ❏ Explore how to evaluate the performance of RAG, and how to improve the performance of RAG.

Retrieval-Augmented Generation (RAG)

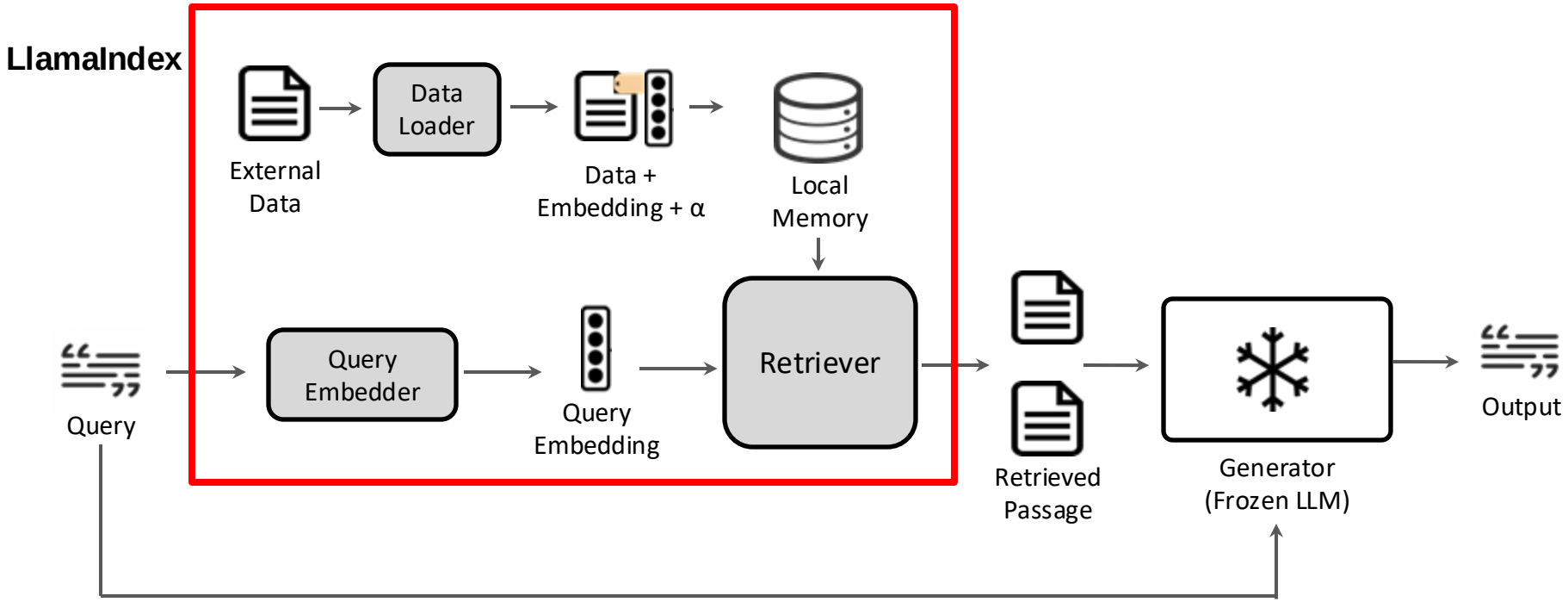
- AI framework for improving the quality of LLM-generated responses by grounding the model on external sources of knowledge to supplement the LLM's internal representation of information [1]
- Components
  - Retriever
    - Input: query, database
    - Output: relevant passage from database
  - Generator (Frozen LLM)
    - Input: prompt (query + response)
    - Output: model-generated output (answer for QA task)

[1]https://research.ibm.com/blog/retrieval-augmented-generation-RAG

Architecture of RAG



Implementation Method



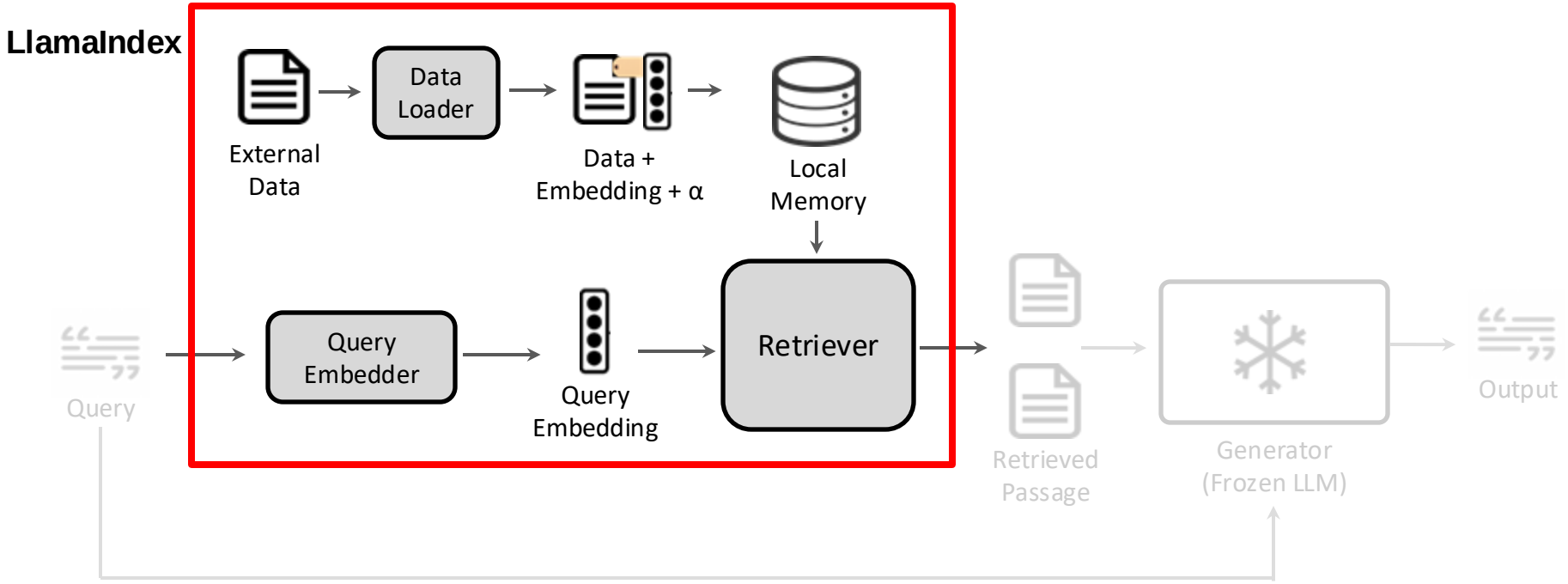
Practice Instruction

- Building a prototype RAG
  - Input
    - Prompt: City-related question
    - External Knowledge Corpus: Wikipedia pages in several cities
  - Output: Answer to the question
- Initializing(Implement) RAG evaluation metrics
- Finding the best RAG app configuration using evaluation

Overview

- ❑ Introduction
  - ❑ Practice Purpose
  - ❑ Retrieval Augmented Generation (RAG) & Practice Instruction
- ❑ Background
  - ❑ LlamaIndex
- ❑ Implementation Detail
  - ❑ How to Evaluate
  - ❑ Changeable Configuration of RAG

LlamaIndex in RAG

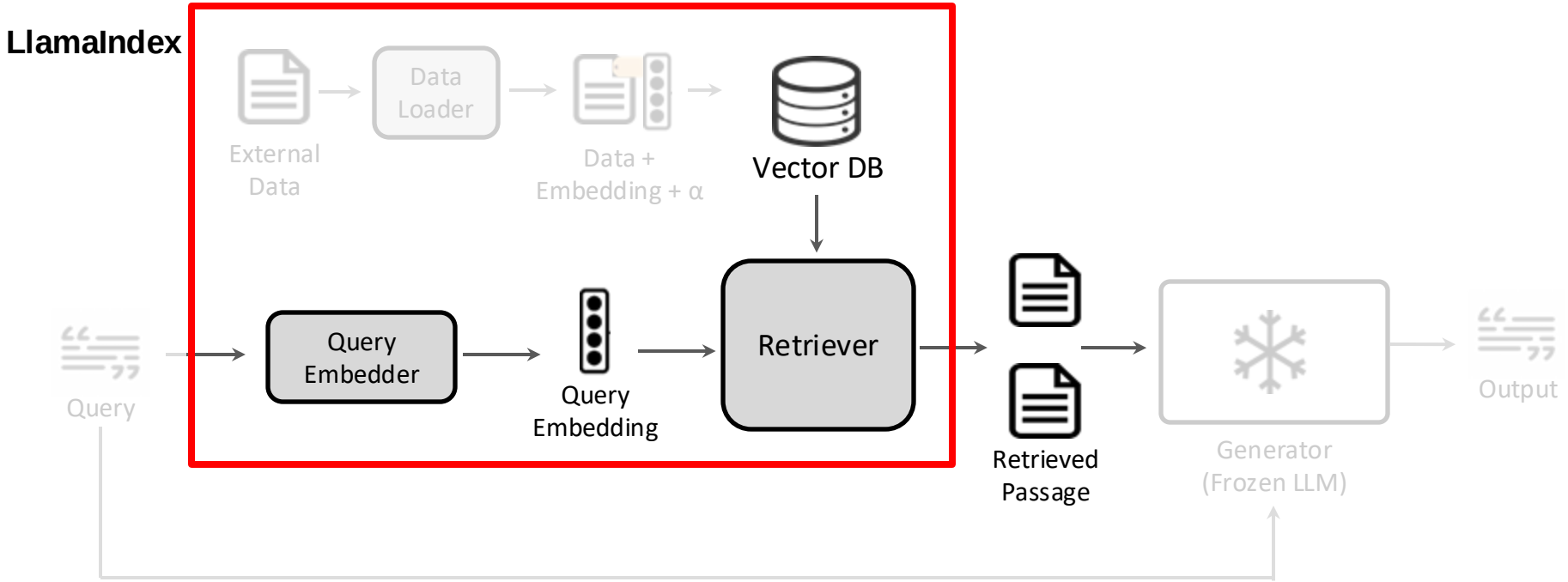


LlamaIndex

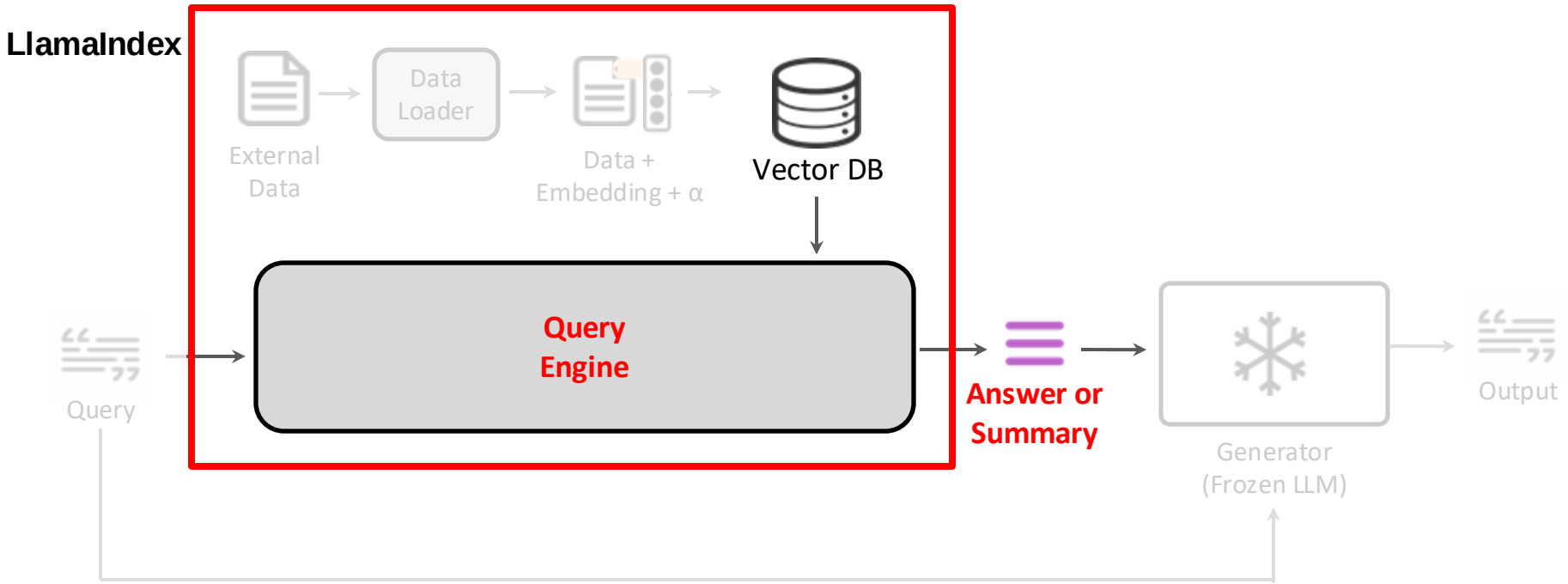
- Framework for building context-augmented LLM applications [2]
- Context augmentation: Any use case that applies LLMs on top of private or domain-specific data [2]
- Role in RAG
  - Query Engine
  - External Knowledge Database Management
  - Embedder

[2]https://docs.llamaindex.ai/en/stable/

RAG without Query Engine



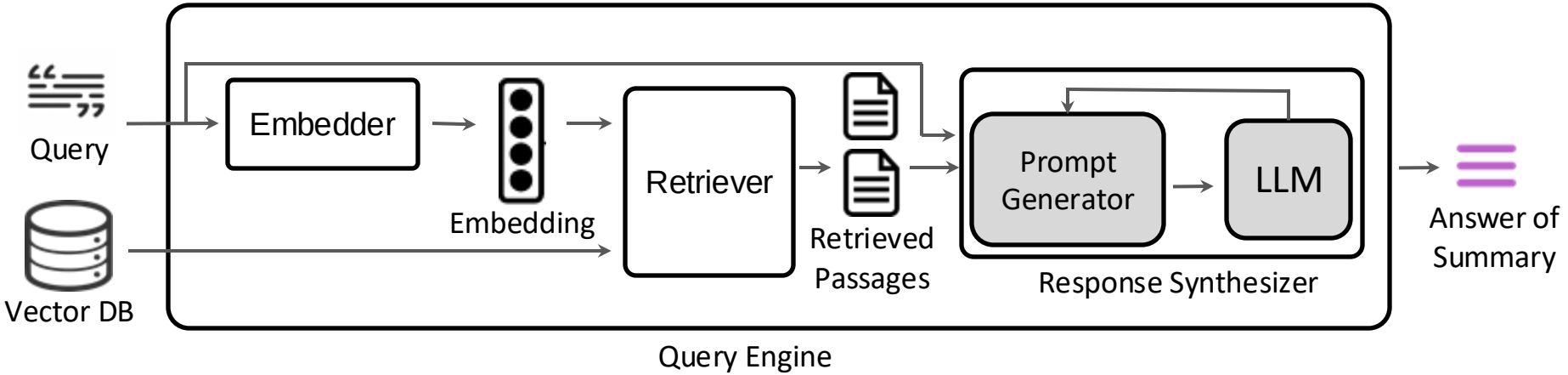
RAG with Query Engine



Query Engine

Generic interface that allows user to ask question over external data [3]

- ❏ Retriever: Fetching relevant passages from the database based on a user query embedding [4]
- ❏ Response Synthesizer: Generating a response from an LLM, using a user query and retrieved passages [5]



[3]https://docs.llamaindex.ai/en/stable/module_guides/deploying/query_engine
[4]https://docs.llamaindex.ai/en/stable/module_guides/querying/retriever
[5]https://docs.llamaindex.ai/en/stable/module_guides/querying/response_synthesizers/?h=response+synthesizer

Nightlife and festivals ===\nBerlin's nightlife has been celebrated as one of the most diverse and vibrant of its kind. In the 1970s and 80s, the SO36 in Kreuzberg was a center for punk music and culture. The SOUND and the Dschungel gained notoriety. Throughout the 1990s, people in their 20s from all over the world, particularly those in
. . .
. . .
International Berlin Music Week. Every year Berlin hosts one of the largest New Year's Eve celebrations in the world, attended by well over a million people. The focal point is the Brandenburg Gate, where midnight fireworks are centered, but various private fireworks displays take place throughout the entire city. Partygoers in Germany often toast the New Year with a glass of sparkling wine.

Retrieved Passage

Input Query:

What's the arts and culture scene in Berlin?

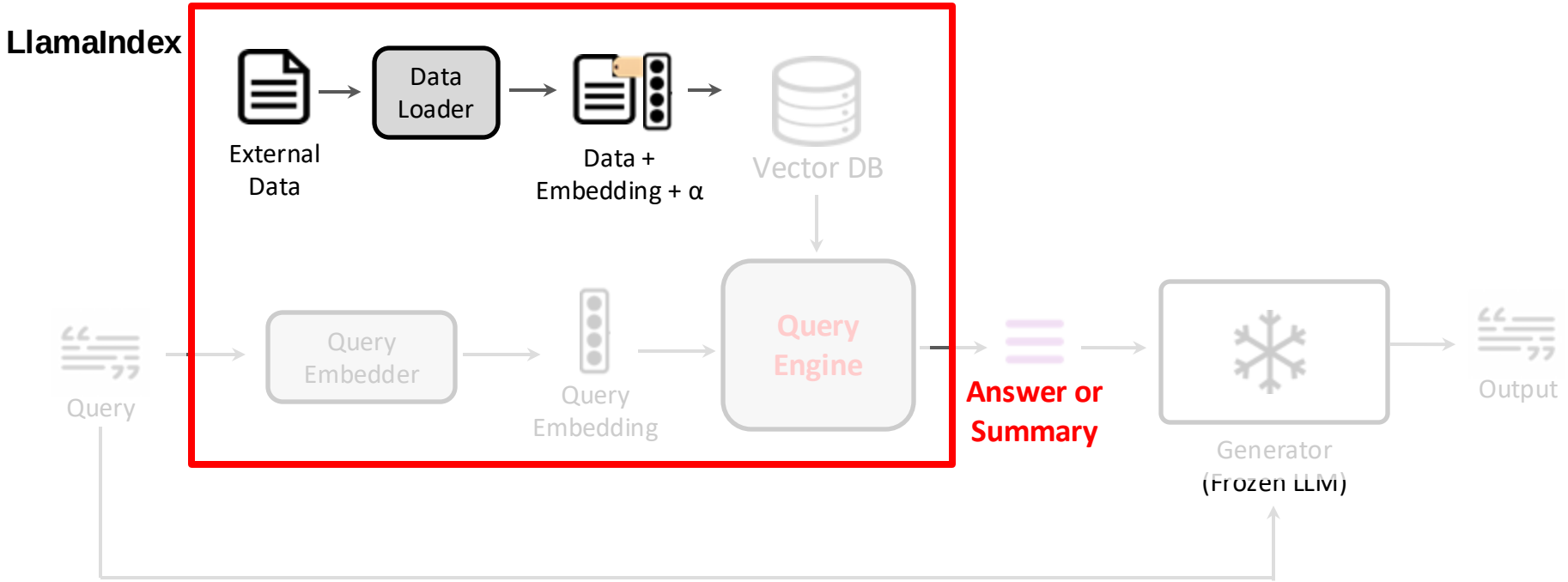
Berlin boasts a rich arts and culture scene with a variety of offerings. The city is home to numerous theaters and stages, including historic venues like the Deutsches Theater, Volksbühne, and . . .
tem V. The city also hosts a range of cultural festivals and events, such as the Berlin International Film Festival, Karneval der Kulturen, and Berlin Festival, making it a hub for artistic expression and creativity.

Query Engine Response

The arts and culture scene in Berlin is rich and diverse, featuring numerous theaters,
. . .
events such as the Berlin International Film Festival and Karneval der Kulturen.

Generator Output

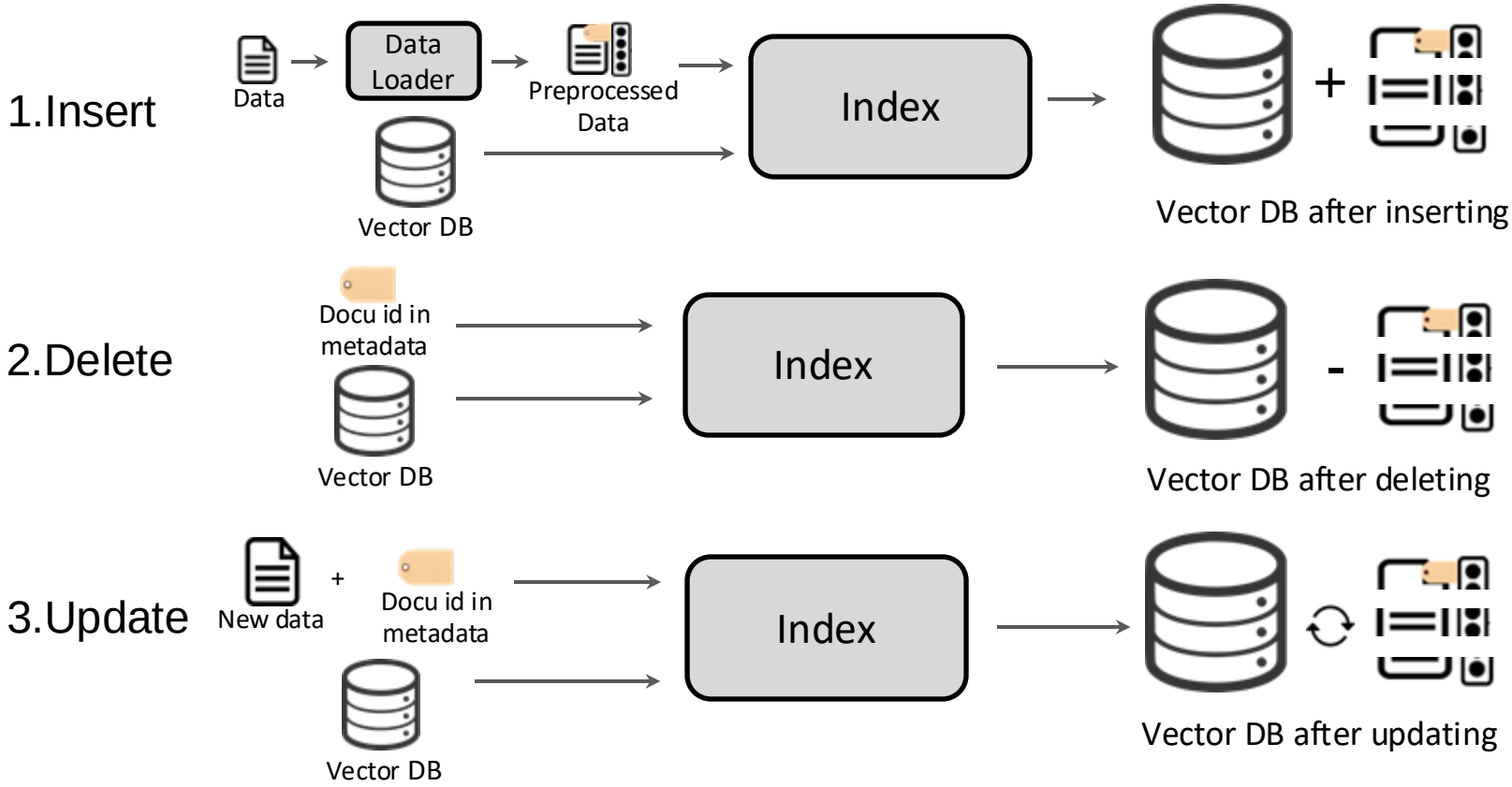
External Knowledge Database Management



Management Principle

- Database Component: Depends on vector database (Only original data vs Embedding with original data)
- Storage Location: Connected vector database
- Indexing Strategy: Index algorithm of vector database
- Supported Operations
  - Insert
  - Delete
  - Update

Supported Operations



Vector Database

Vector database which are specialized systems designed for managing and retrieving unstructured data through vector embeddings and numerical representations [6]

[6]https://milvus.io/intro

Index

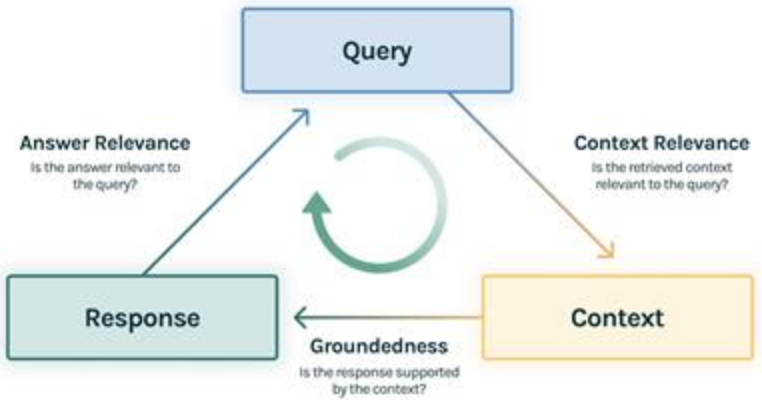
- Motivation: To enable a quick and efficient retrieval for vector similar to a specific vector in a large amount of high-dimensional vector data.
- Examples
  - FLAT (default algorithm)
  - HNSW
  - IVF-FLAT
  - IVF_SQ8

Overview

- ❑ Introduction
  - ❑ Practice Purpose
  - ❑ Retrieval Augmented Generation (RAG) & Practice Instruction
- ❑ Background
  - ❑ LlamaIndex
- ❑ Implementation Detail
  - ❑ How to Evaluate
  - ❑ Changeable Configuration of RAG

Evaluation Metric

- Answer relevance: ‘query’-’answer’ relevance ∈ [0, 1]
- *Context* relevance: ‘query’-’query engine response’ relevance ∈ [0, 1]
  - *Context*: Retrieved Passages
- Groundedness: ‘answer’-’query engine response’ relevance ∈ [0, 1]



Evaluation API

- TruLens: Open source library for evaluating and tracking large language model-based applications
- How it works
  - Build LLM application (RAG)
  - Connect application to TruLens and start logging the records.
  - Add feedback functions to log and evaluate LLM application.
  - Check records in dashboard

Changeable Configuration of RAG

- Chunk size of Retriever
- Custom Query Engine
  - Generating summary
  - Generating direct answer
- Prompt Design

Reference

[1] https://research.ibm.com/blog/retrieval-augmented-generation-RAG
[2] https://docs.llamaindex.ai/en/stable/
[3] https://docs.llamaindex.ai/en/stable/module_guides/deploying/query_engine
[4] https://docs.llamaindex.ai/en/stable/module_guides/querying/retriever
[5] https://docs.llamaindex.ai/en/stable/module_guides/querying/response_synthesizers/?h=response+synthesizer
[6] https://milvus.io/intro
[7] https://docs.llamaindex.ai/en/stable/module_guides/loading/documents_and_nodes/