INF Informatik
Hochschule Reutlingen

Intelligente Informationssysteme
Retrieval Augmented Generation

Dominik Neumann

Bock 3 – Retrieval Augmented Generation

Design Patterns

Informatik
Hochschule Reutlingen

Conversional AI

Simulate a conversation with the feeling of having a conversation with a human.

- conversational memory
- dialogue generation

- Examples: ChatGPT, Anthropic Claude, Perplexity.AI

Retrieval Augmented Generation

Knowledge Retrieval and understanding is key

- Access to contextual data
- Retrieval and augmentation strategies

CoPilot

Assists a human in his work.
Key differentiator for becoming a CoPilot is understanding of the environment in which the user works.

- access to tools and data,
- reasoning and planning capabilities,
- and specialized profiles
- Examples: GitHub CoPilot

Multi Agent Problem Solver

Multiple agents collaborate to solve a problem. Each agent

- has access to its own set of tools and
- can assume a very specific role while reasoning
- is planning (and executing) its actions.

- Example: Devin (https://preview.devin.ai)

Why Retrieval Augmented Generation?

INF Informatik
Hochschule Reutlingen

LLM Challenges

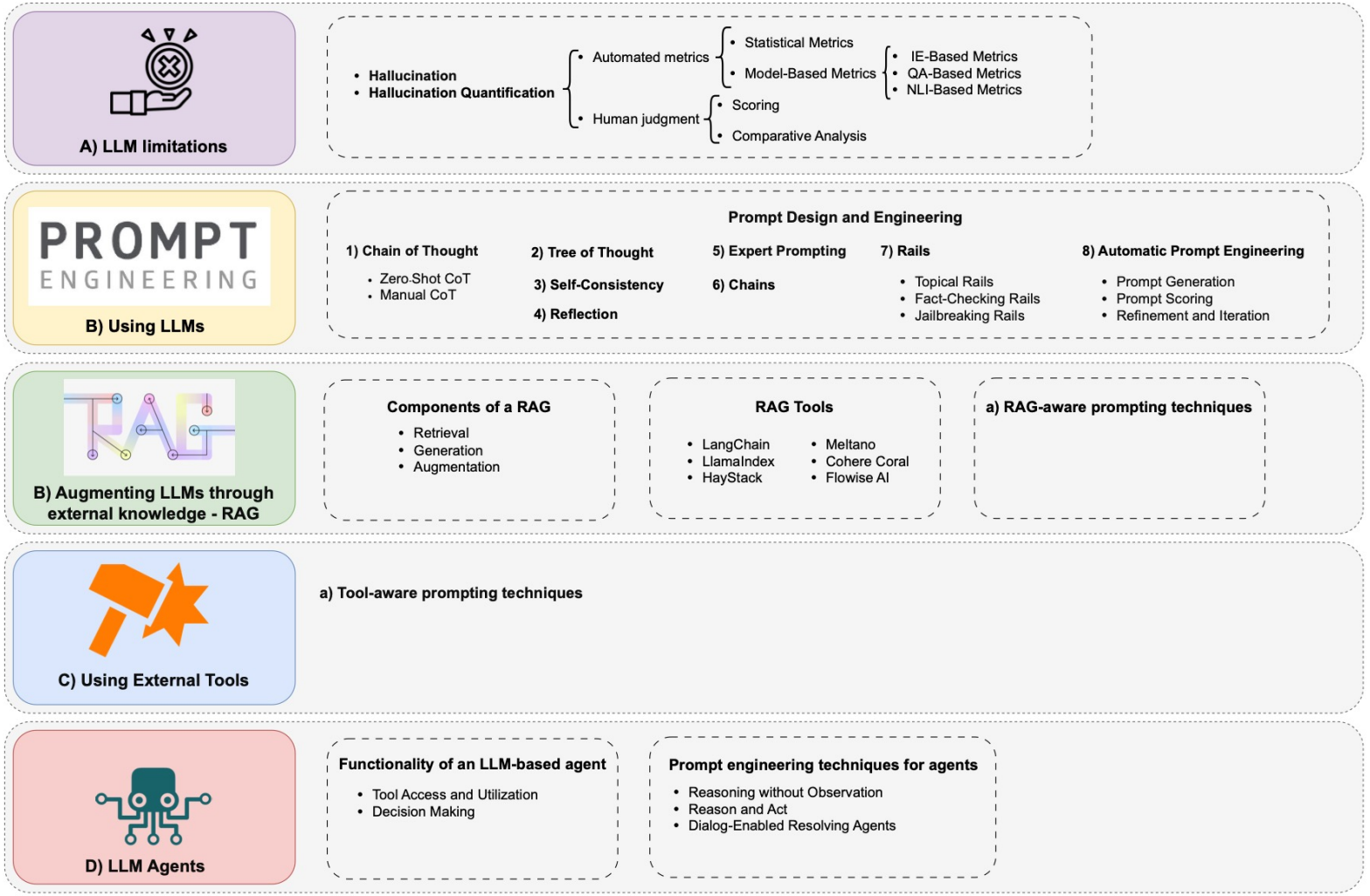
Challenge	
Hallucination	Models always generate text as an extrapolation from the prompt you provided. Hallucination is the default. Sometimes the model generates text that is incorrect, nonsensical, or not real.
Attribution	Since LLMs are not databases or search engines, they would not cite where their response is based on. We have neither traceability nor explainability.
Staleness	Model contain incomplete, outdated, or wrong knowledge due to outdated training data.
Generalization	If we have use cases with non-publicly accessible knowledge, then the knowledge is not part of the chosen LLM. We want to retrain the LLM or enhance it with the knowledge.
Revision	How can incorrect knowledge or facts be replaced in an LLM?

Model Limitation

Despite the successes achieved through scaling, a language model is limited in its area of application.

1. Models are limited in what they know about the world.
2. Models are limited in kind of tasks they can solve
3. And, in addition, models are hard to adapt

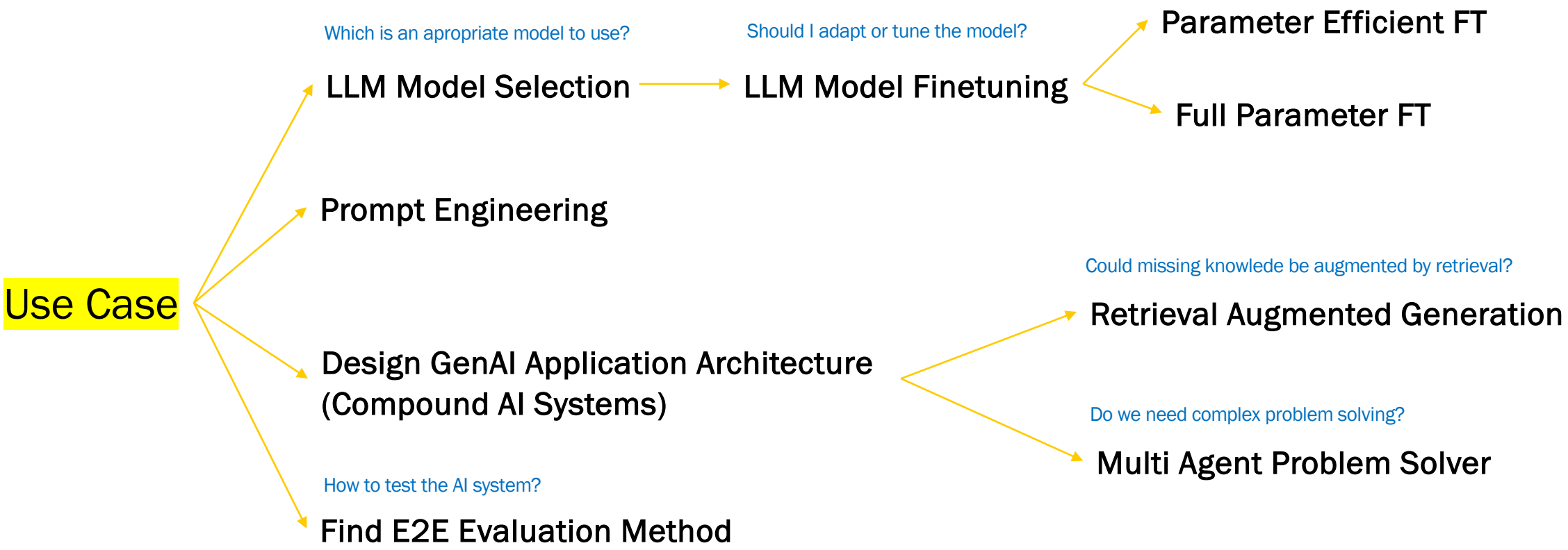
LLM Augmentation



Source: https://arxiv.org/pdf/2402.06196 -Fig36

Informatik
Hochschule Reutlingen

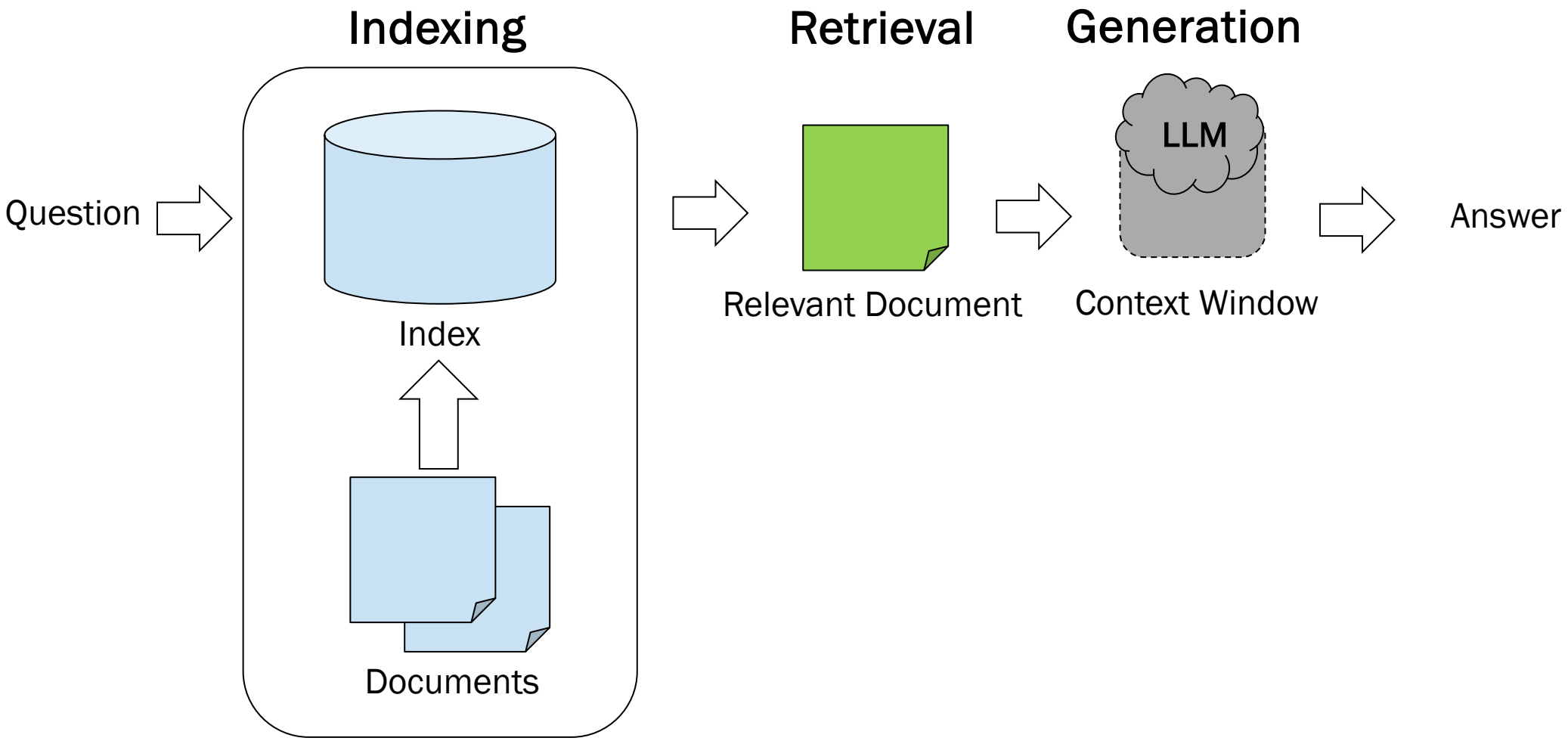
How to deal with challenges



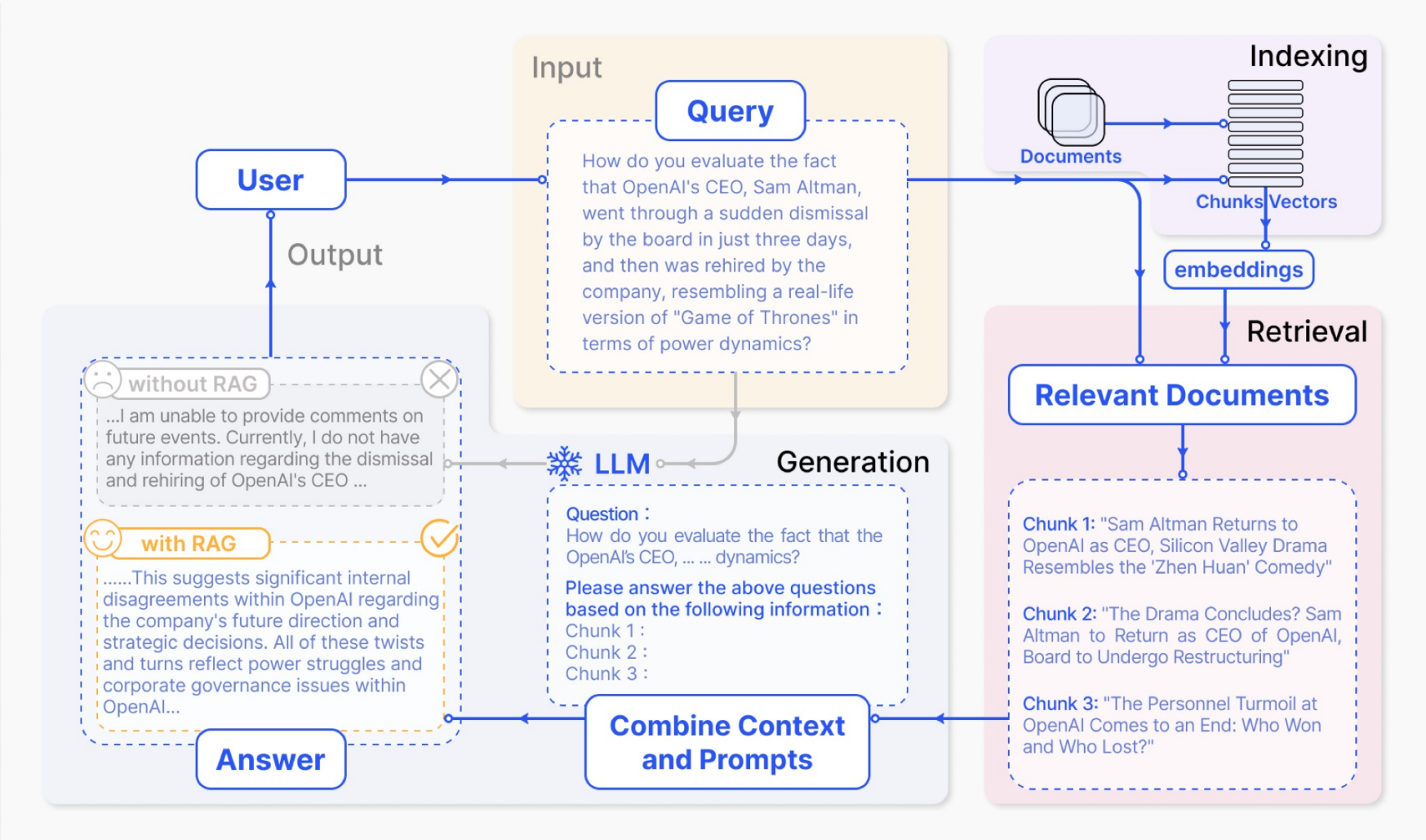
Short Introduction into Retrieval Augmented Generation

INF Informatik
Hochschule Reutlingen

Retrieval Augmented Generation

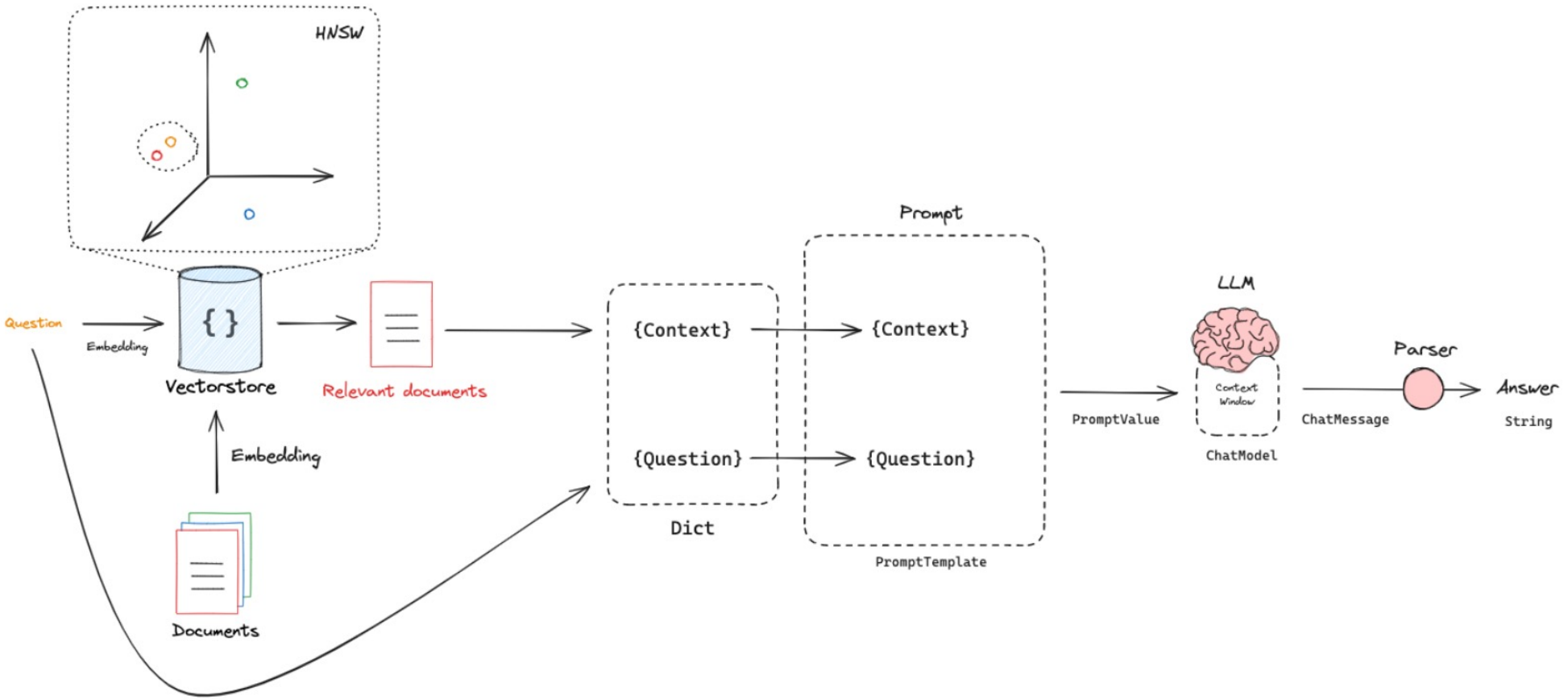


Retrieval Augmented Generation



Source: https://arxiv.org/pdf/2312.10997

Retrieval Augmented Generation



Source: https://github.com/langchain-ai/rag-from-scratch/blob/main/rag_from_scratch_1_to_4.ipynb

Start with some Notebooks

Informatik
Hochschule Reutlingen

As Vectore Database we use ChromaDB:

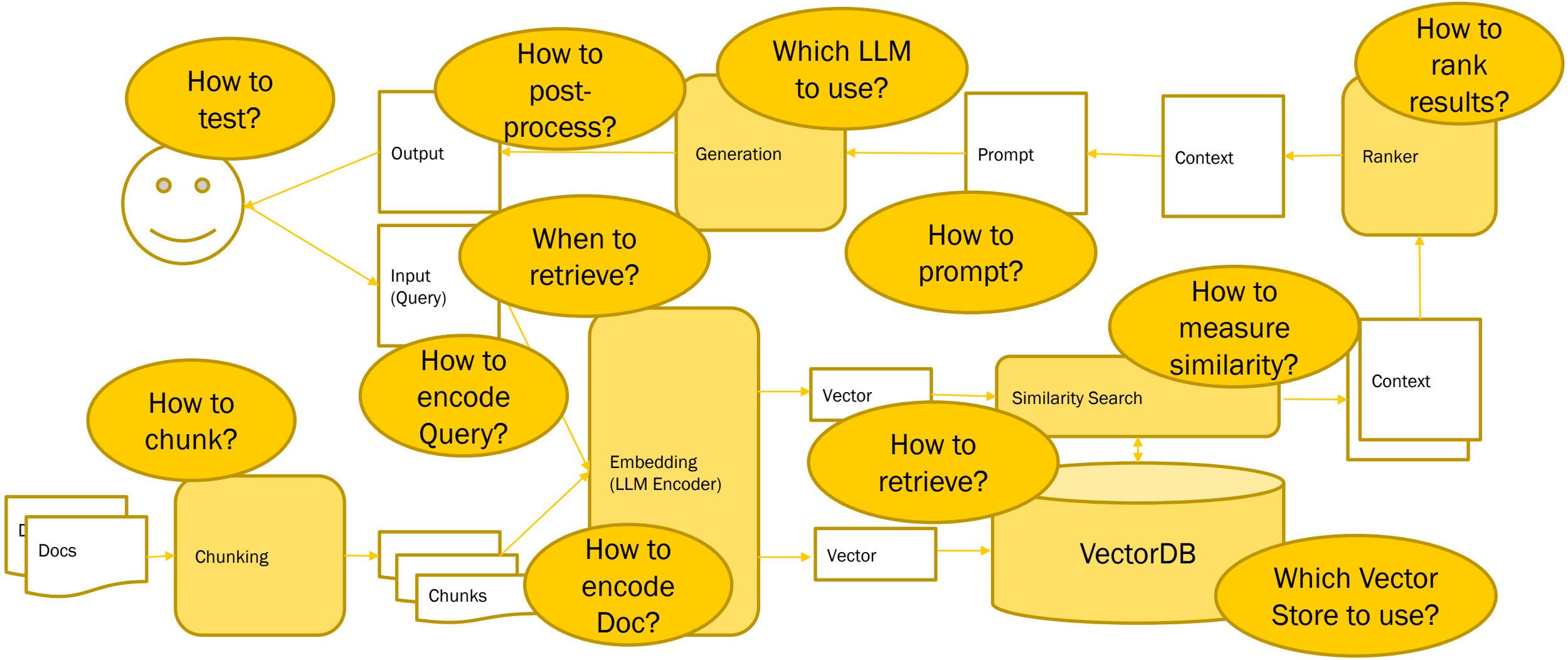
- Understand Vector Databases: 00_Vectorstore.ipynb

As RAG Framework we use: LangChain and LlamaIndex

- Understand RAG – Indexing: 01_Simple_RAG_Indexing.ipynb
- Understand RAG – Retrieval: 02_Simple_RAG_Retrieval.ipynb

What are the challenges with RAG solutions?

Informatik
Hochschule Reutlingen

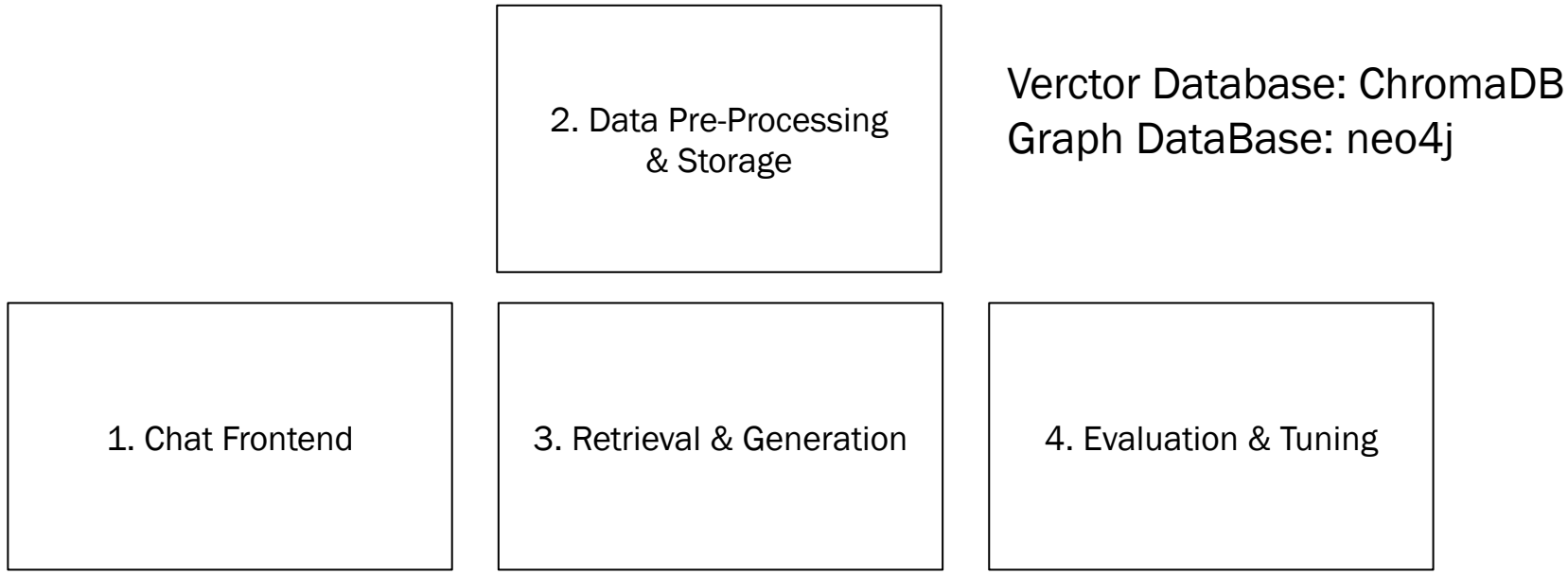


Short Introduction into our Projektaufgabe

Projektaufgabe

Hochschule Reutlingen

- Idee: Wir erstellen als gemeinsames Projekt ein Retrieval Augmented Generation System, das Vorlesungen aus Youtube verarbeitet.



INF
Informatik
Hochschule Reutlingen

01
Chat Frontend

INF
Informatik
Hochschule Reutlingen

02
Data Pre-Processing & Storage

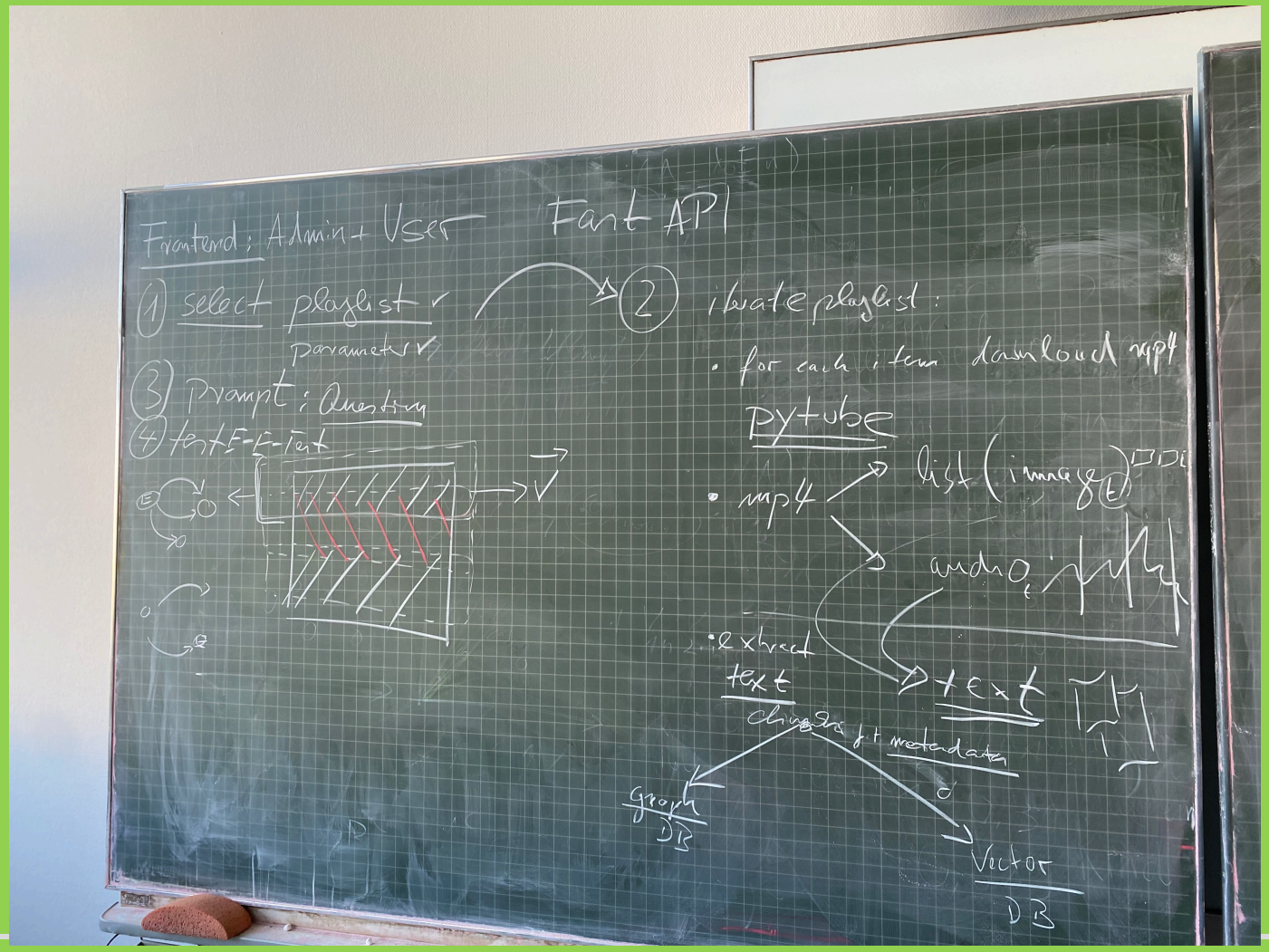
Discussion about Data Pre-Processing & Storage

Data

- Video — video -> image & video -> audio
  - Image — image -> text
  - Audio — audio -> text
- Text

Databases:

- Vector Database
- Graph Database
- SQL Database
- Index



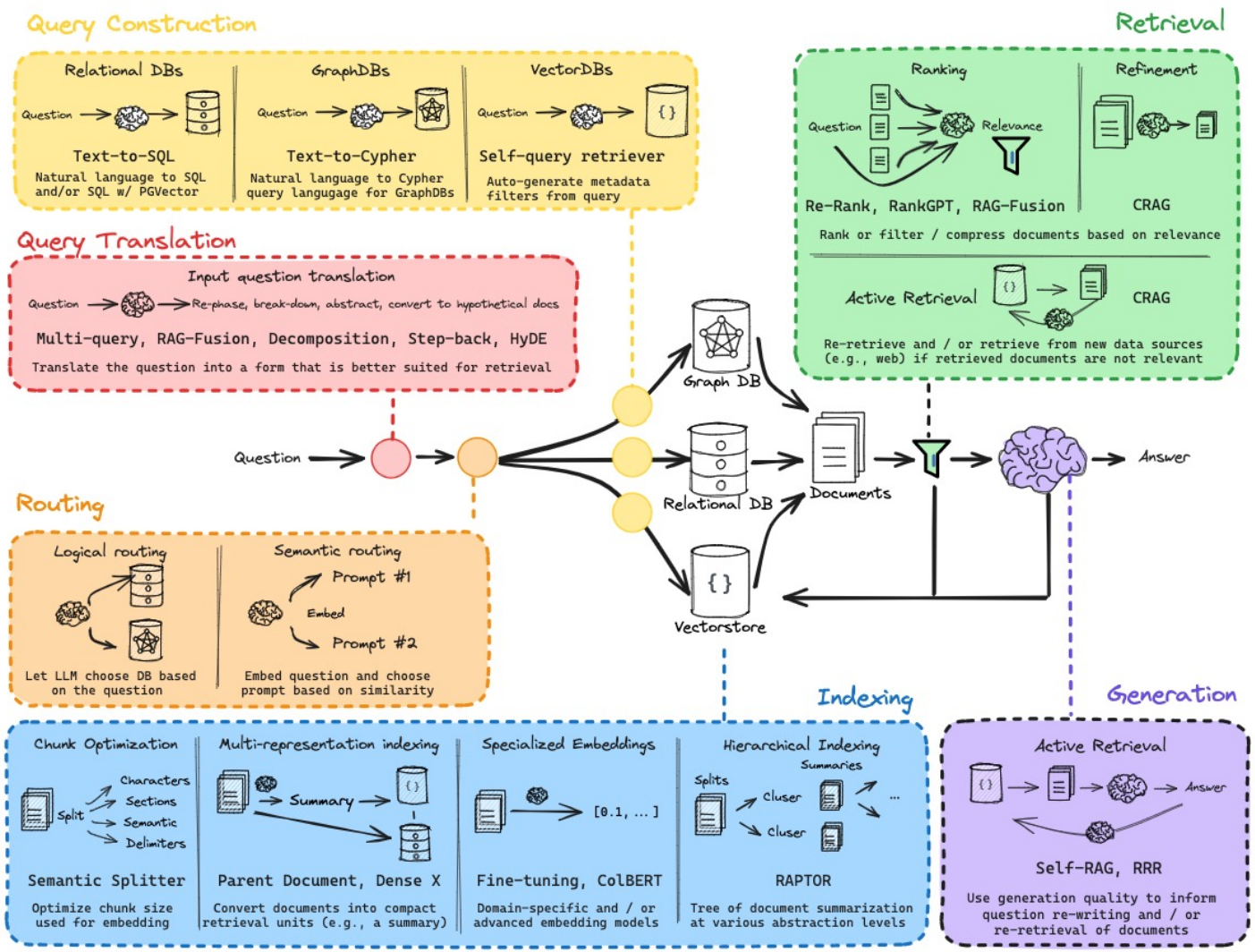
03
Retrieval Augmented Generation

We follow the tutorial RAG From Scratch from LanngChain:
https://github.com/langchain-ai/rag-from-scratch

The tutorial uses langchain version 0.2 – we migrated the notebooks to version 0.3.

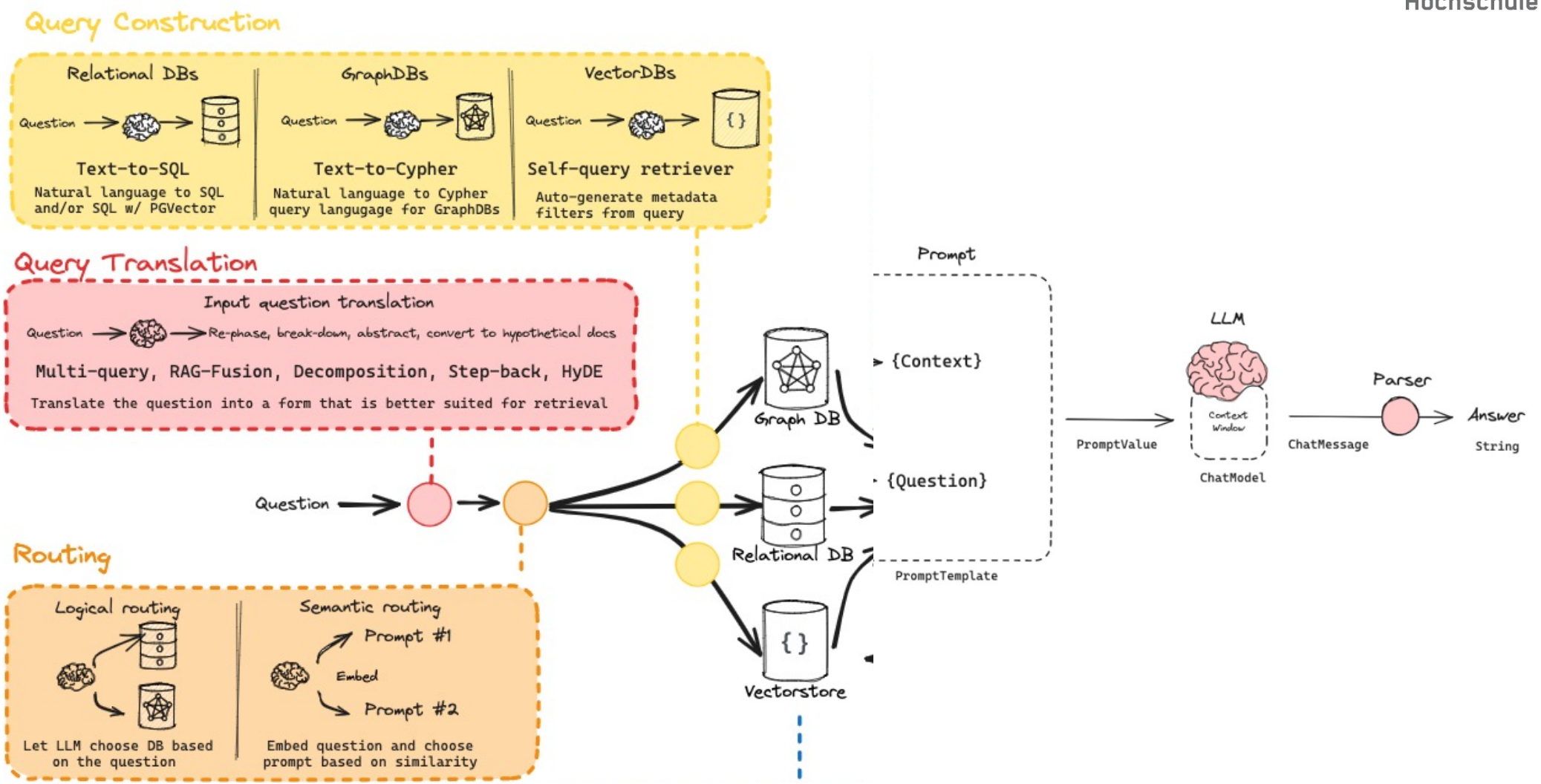
Retrieval Augmented Generation

1. Query Transformation
2. Routing
3. Query construction
4. Retrieval
5. Generation

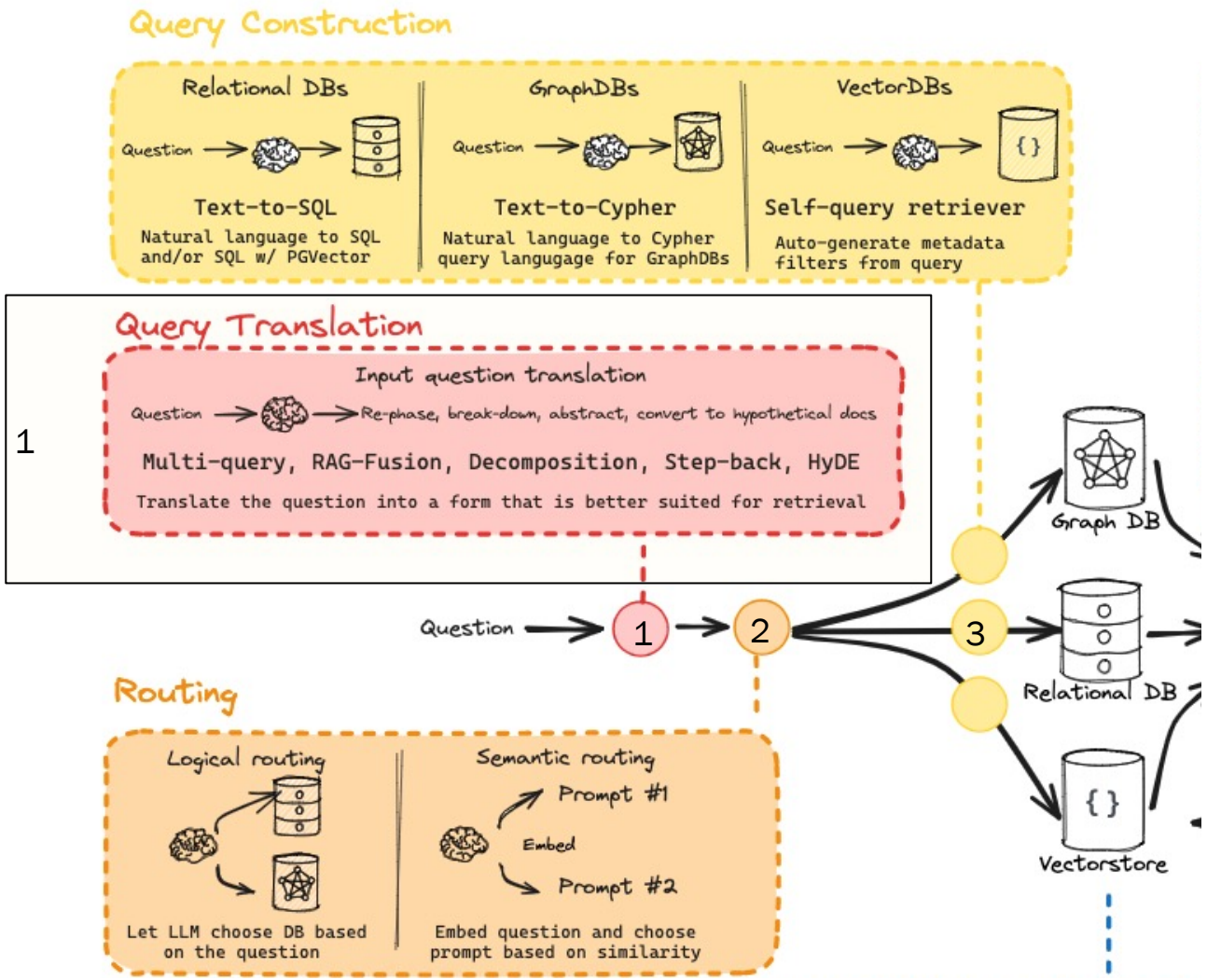


Source: Langchain https://docs.google.com/presentation/d/1C9IaAwHoWcc4RSTqo-pCoN3h0nCgqV2JEYZUJunv_9Q

Question Construction



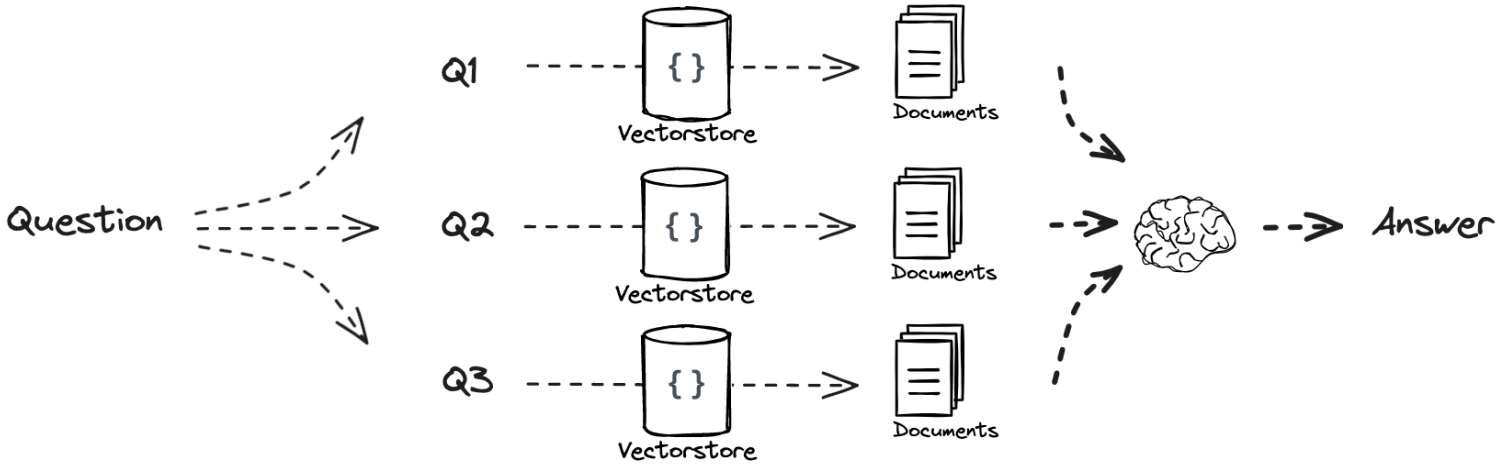
Query Translation



Goal: Translate the input question in a form that is better suited for retrieval:

- Multi-Query
- RAG-Fusion
- Decomposition
- Step-back
- HyDE

Query Translation: Multi-Query

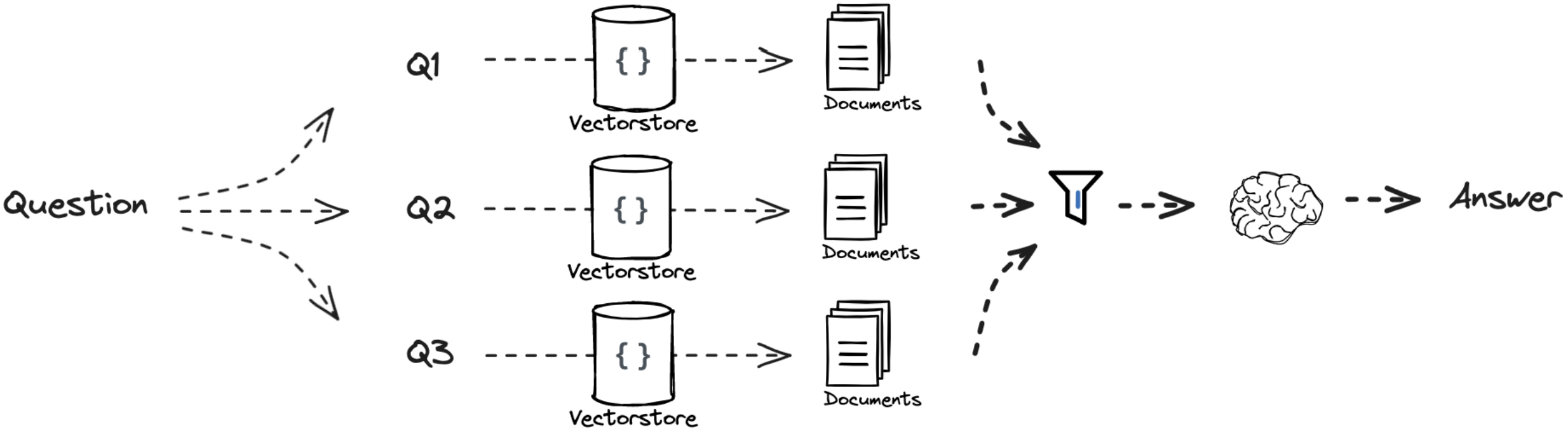


Prompt:

"""You are an AI language model assistant. Your task is to generate five different versions of the given user question to retrieve relevant documents from a vector database. By generating multiple perspectives on the user question, your goal is to help the user overcome some of the limitations of the distance-based similarity search. Provide these alternative questions separated by newlines. Original question: {question}"""

Source: https://github.com/langchain-ai/rag-from-scratch/blob/main/rag_from_scratch_5_to_9.ipynb
https://python.langchain.com/docs/how_to/MultiQueryRetriever/

Query Transition: RAG Fusion



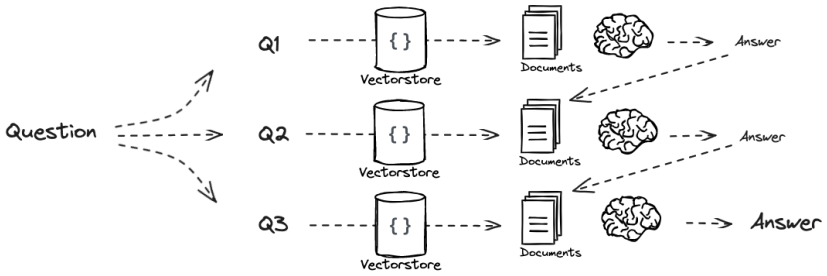
Source: https://github.com/langchain-ai/rag-from-scratch/blob/main/rag_from_scratch_5_to_9.ipynb

Query Translation: Decomposition

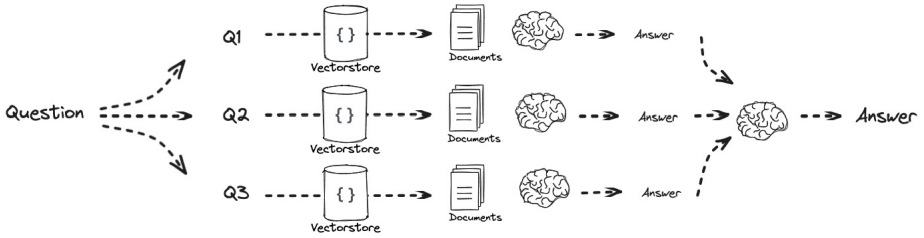
Hochschule Reutlingen

Decompose the question into subqueries and

answer recursively



or individually



Prompt:

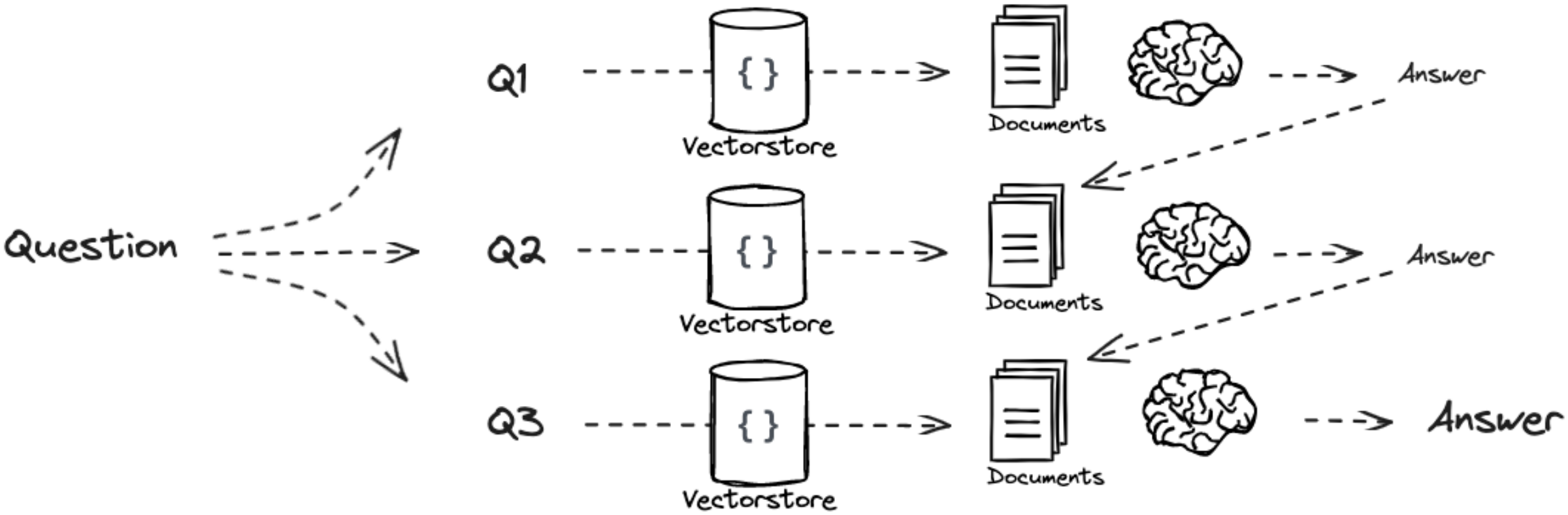
"""You are a helpful assistant that generates multiple sub-questions related to an input question.
The goal is to break down the input into a set of sub-problems / sub-questions that can be answers in isolation.
Generate multiple search queries related to: {question}
Output (3 queries):“““

Source: https://github.com/langchain-ai/rag-from-scratch/blob/main/rag_from_scratch_5_to_9.ipynb

Query Transition: Decomposition

Hochschule Reutlingen

Answer recursively

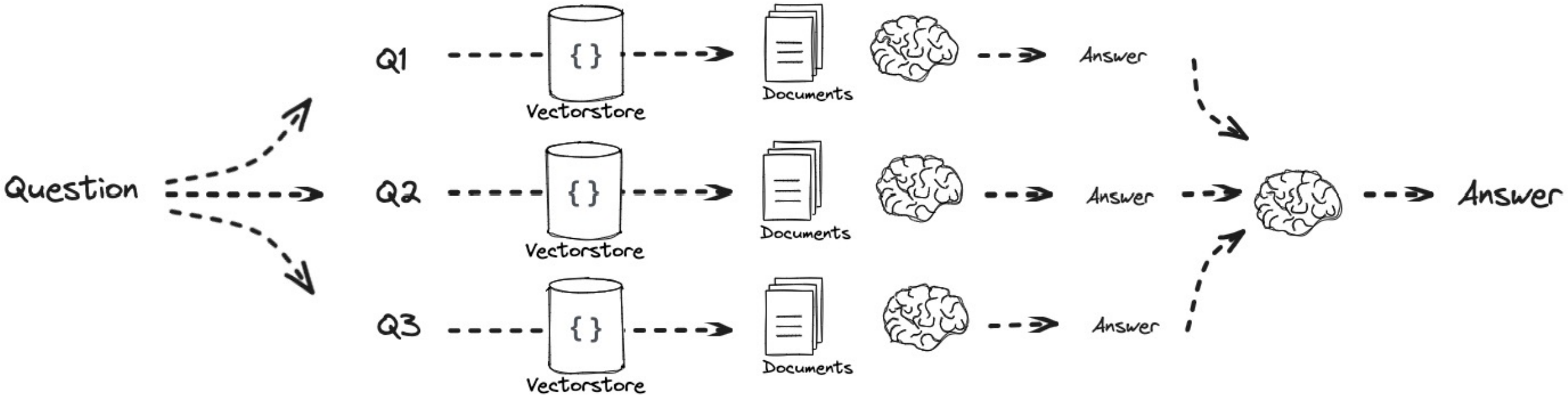


Source: https://github.com/langchain-ai/rag-from-scratch/blob/main/rag_from_scratch_5_to_9.ipynb
https://arxiv.org/pdf/2205.10625.pdf
https://arxiv.org/abs/2212.10509.pdf

Query Transition: Decomposition

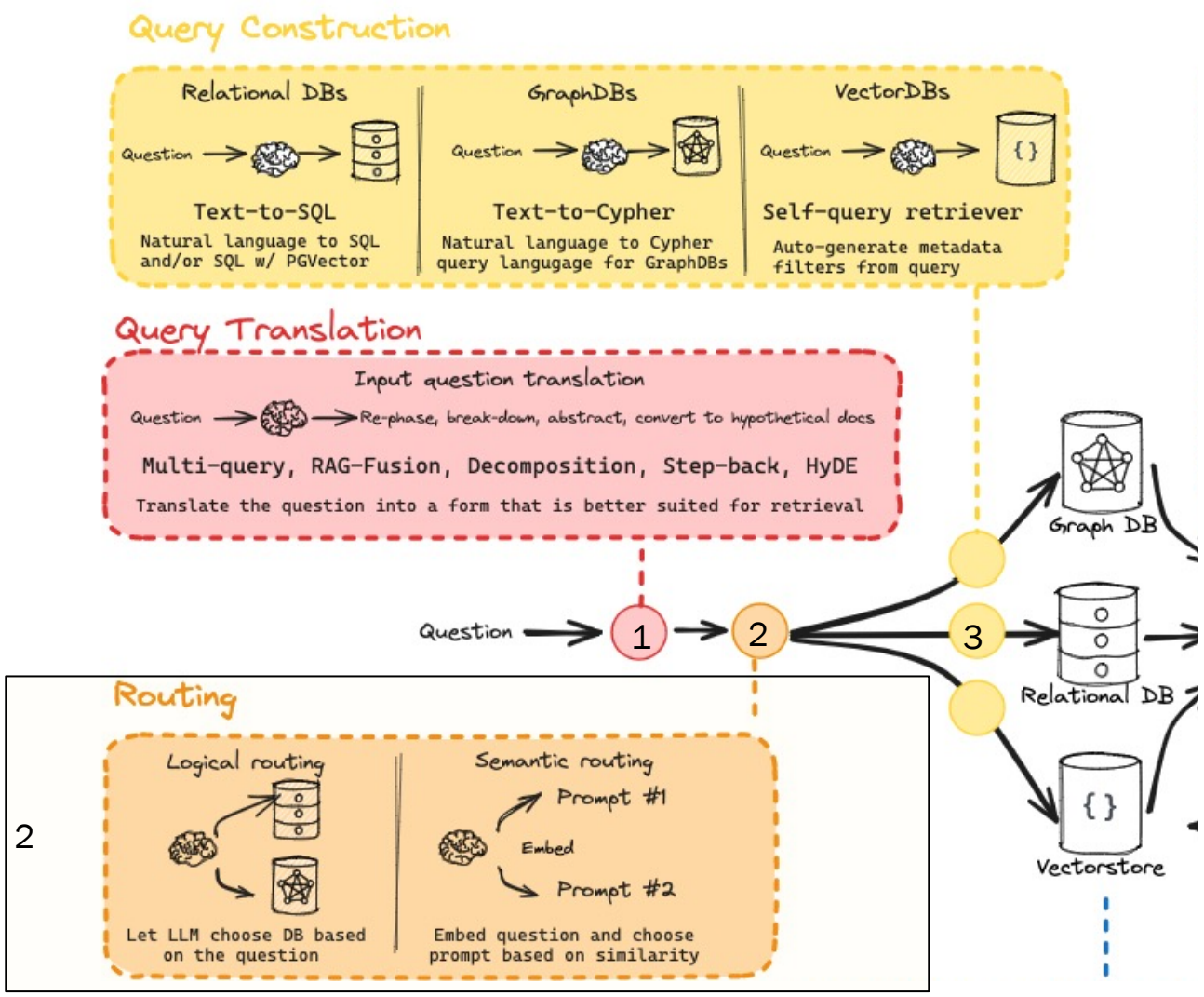
Hochschule Reutlingen

Answer individually



Source: https://github.com/langchain-ai/rag-from-scratch/blob/main/rag_from_scratch_5_to_9.ipynb

Routing

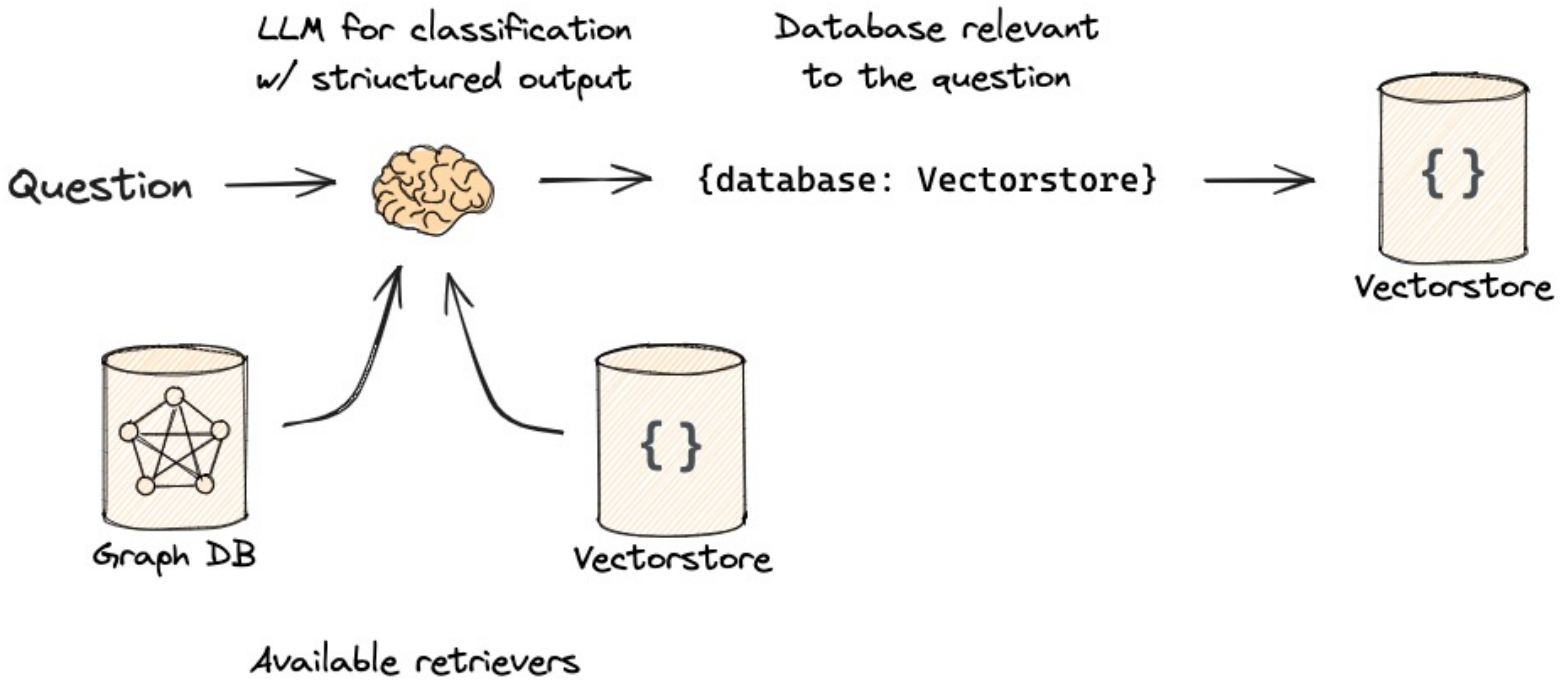


Goal:

- Logical Routing: Find the right Datastore based on question
- Semantic Routing: Choose the right prompt based on similarity

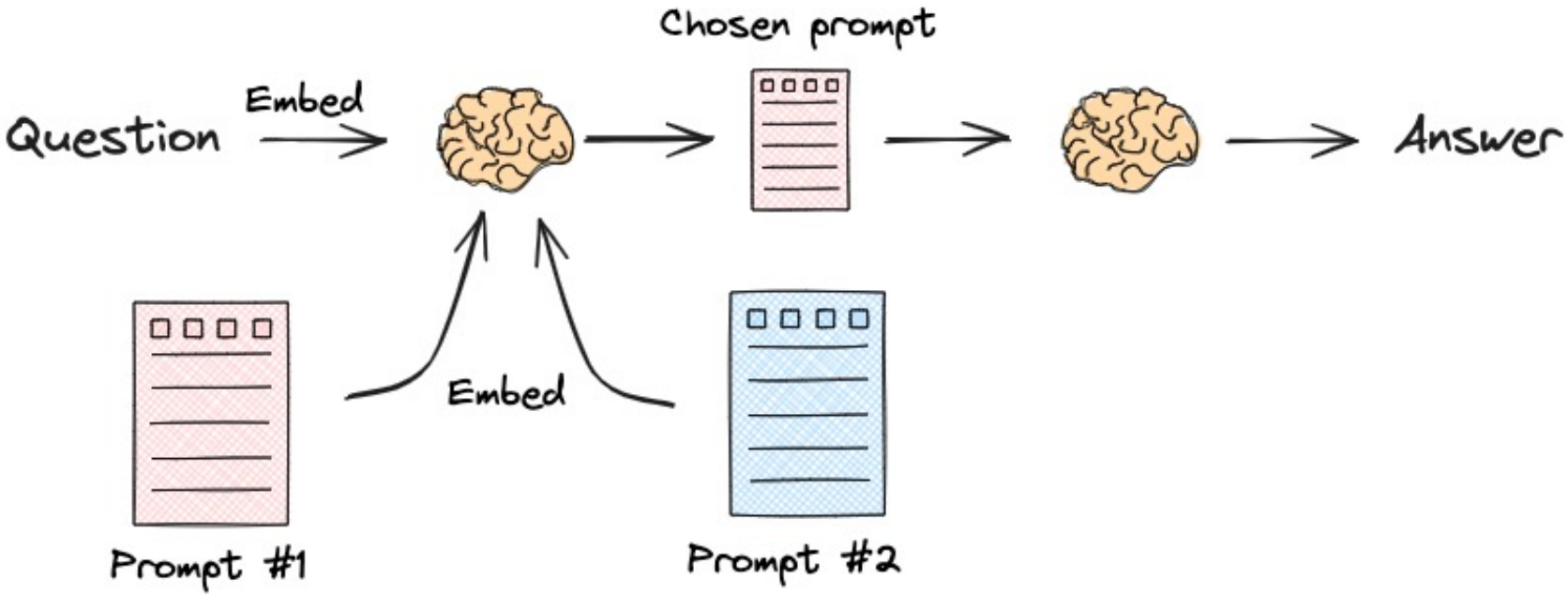
Logical Routing

In case we have different available retrievers: where to retrieve?

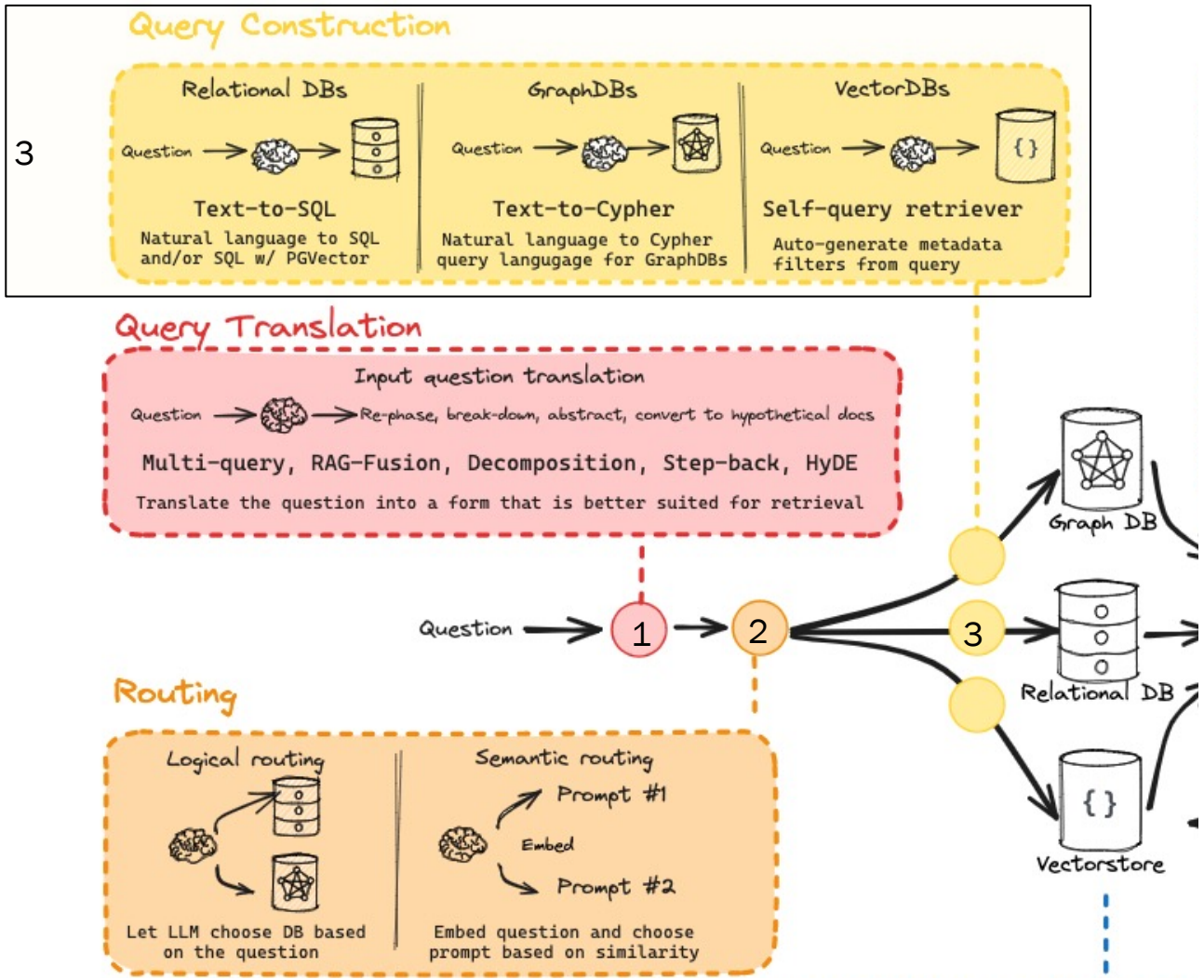


Semantic Routing

In case we have multiple prompts: which prompt should we use?



Query Construction

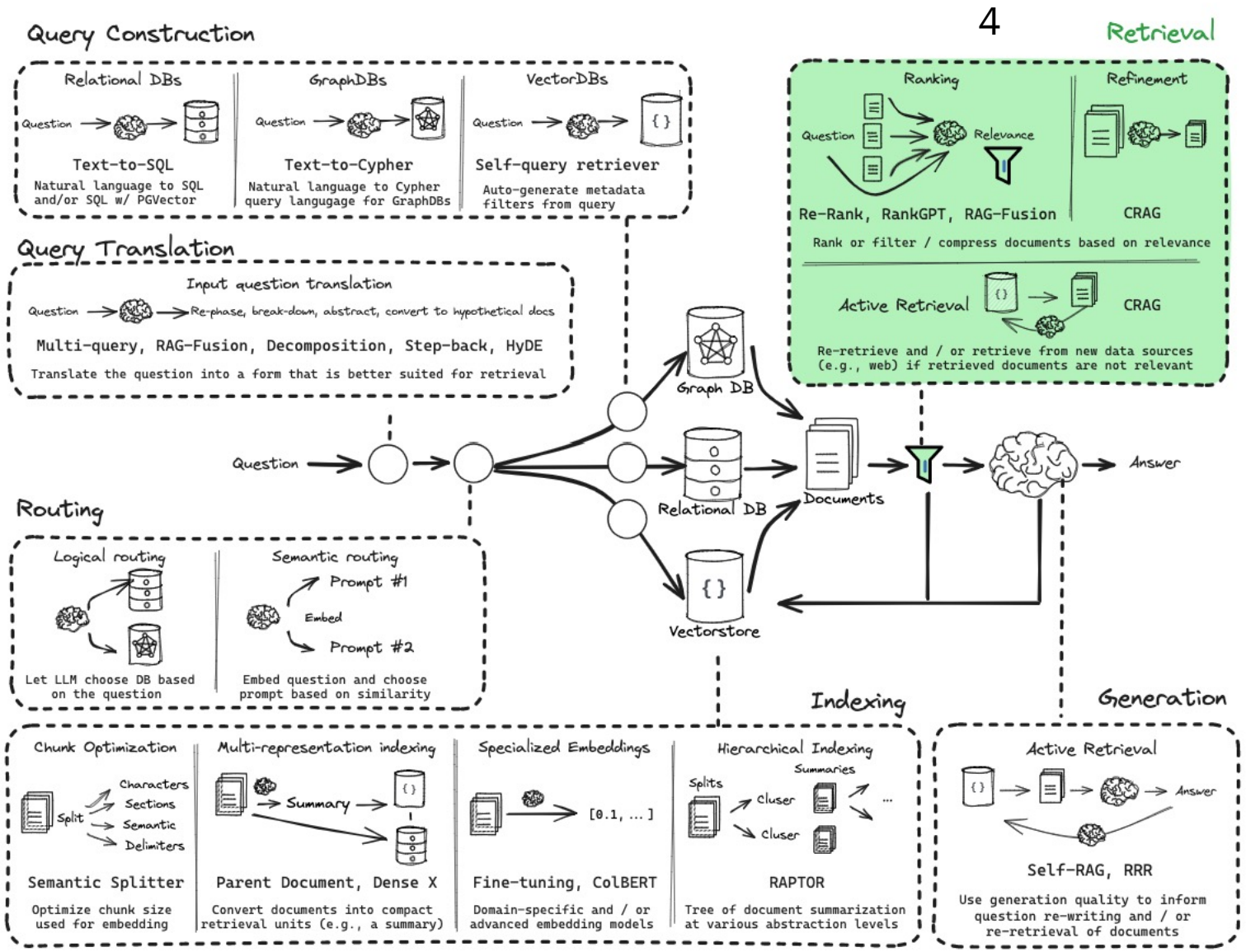


Goal:

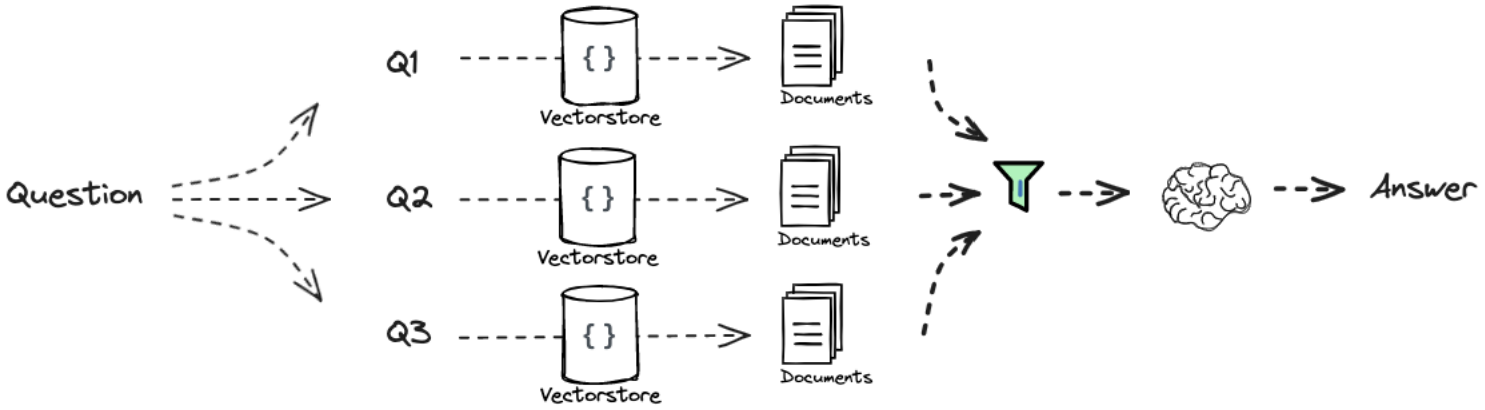
Extract the information from question we need to structure our query:

- Text-to-SQL
- Text-to-Cypher
- Extract Metadata, Entities, ...

Advanced Retrieval



Re-Ranking



https://medium.com/@sahin.samia/what-is-reranking-in-retrieval-augmented-generation-rag-ee3dd93540ee
https://medium.com/@rossashman/the-art-of-rag-part-3-reranking-with-cross-encoders-688a16b64669

Re-Ranking Techniques

Informatik
Hochschule Reutlingen

1. Neural Rerankers:
   - Cross-Encoders: These models jointly encode the query and each retrieved document, assessing their relevance through a single forward pass. This approach captures intricate interactions between the query and documents, leading to precise relevance scoring. However, it can be computationally intensive.
   - Bi-Encoders with Interaction Layers: While bi-encoders independently encode queries and documents, incorporating interaction layers allows for modeling complex relationships, balancing efficiency and accuracy.
2. Traditional Scoring Techniques:
   - BM25: A probabilistic retrieval model that ranks documents based on term frequency and inverse document frequency, effectively handling exact term matches.
     https://homepages.dcc.ufmg.br/~nivio/cursos/ri10/transp/slideschap04c.pdf
     https://en.wikipedia.org/wiki/Okapi_BM25
   - TF-IDF: Evaluates the importance of terms within documents relative to the entire corpus, aiding in identifying key terms that distinguish relevant documents.
3. Hybrid Approaches:
   - Combining Neural and Traditional Methods: Integrating neural rerankers with traditional models like BM25 can leverage the strengths of both, enhancing reranking performance.

Re-Ranking Techniques

Informatik
Hochschule Reutlingen

4. Graph-Based Reranking:
   - Document Graphs: Constructing graphs where nodes represent documents and edges denote semantic relationships enables the identification of clusters of relevant documents, improving reranking outcomes.

5. Fine-Tuning LLMs for Reranking:
   - Instruction Fine-Tuning: Training LLMs with specific instructions for context ranking and answer generation allows a single model to perform both tasks, streamlining the reranking process.

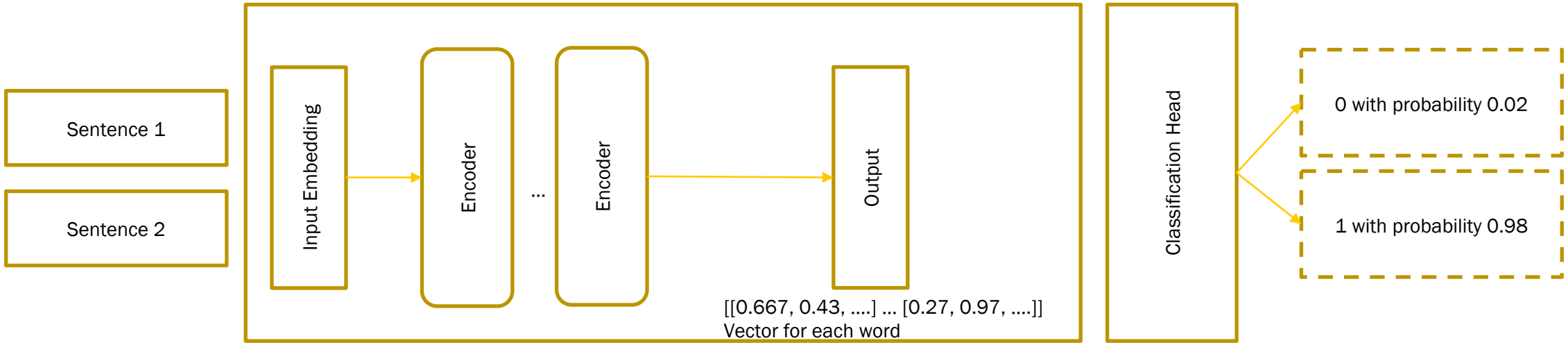
6. Fusion-Based Techniques:
   - Fusion-in-Decoder (FiD): This method processes multiple retrieved documents simultaneously within the decoder, enabling the model to synthesize information from various sources effectively.

Recap: Large Language Models

Hochschule Reutlingen

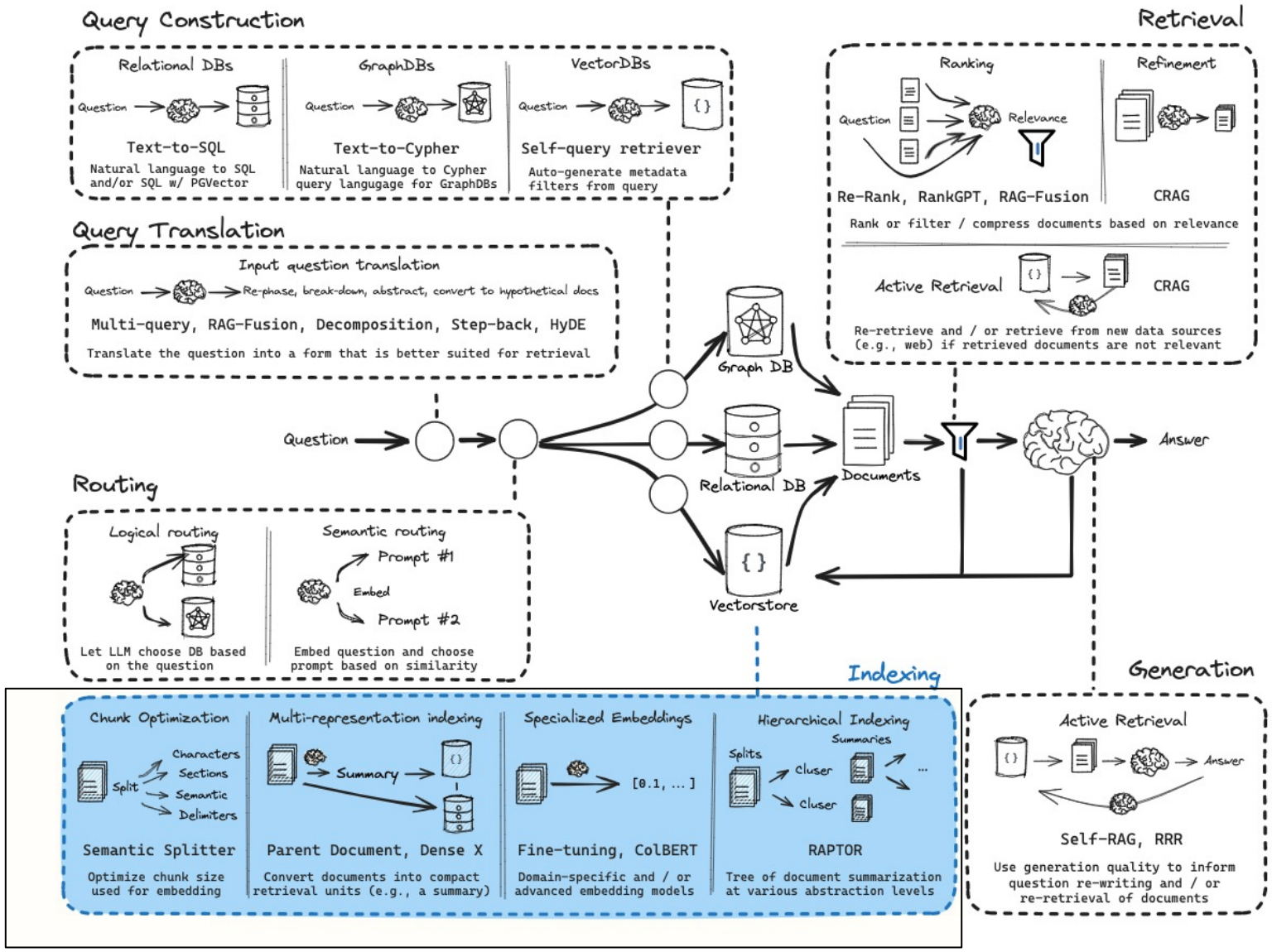
Encoder-Only Models for Semantic Similarity Computation

- CrossEncoder is used to measure similarity between pairs of sentences
- The input of the model always consists of a data pair, for example two sentences, and outputs a value between 0 and 1 indicating the similarity between these two sentences
- can be used as Re-Raker in Retrieval-Augmented Generation (RAG)

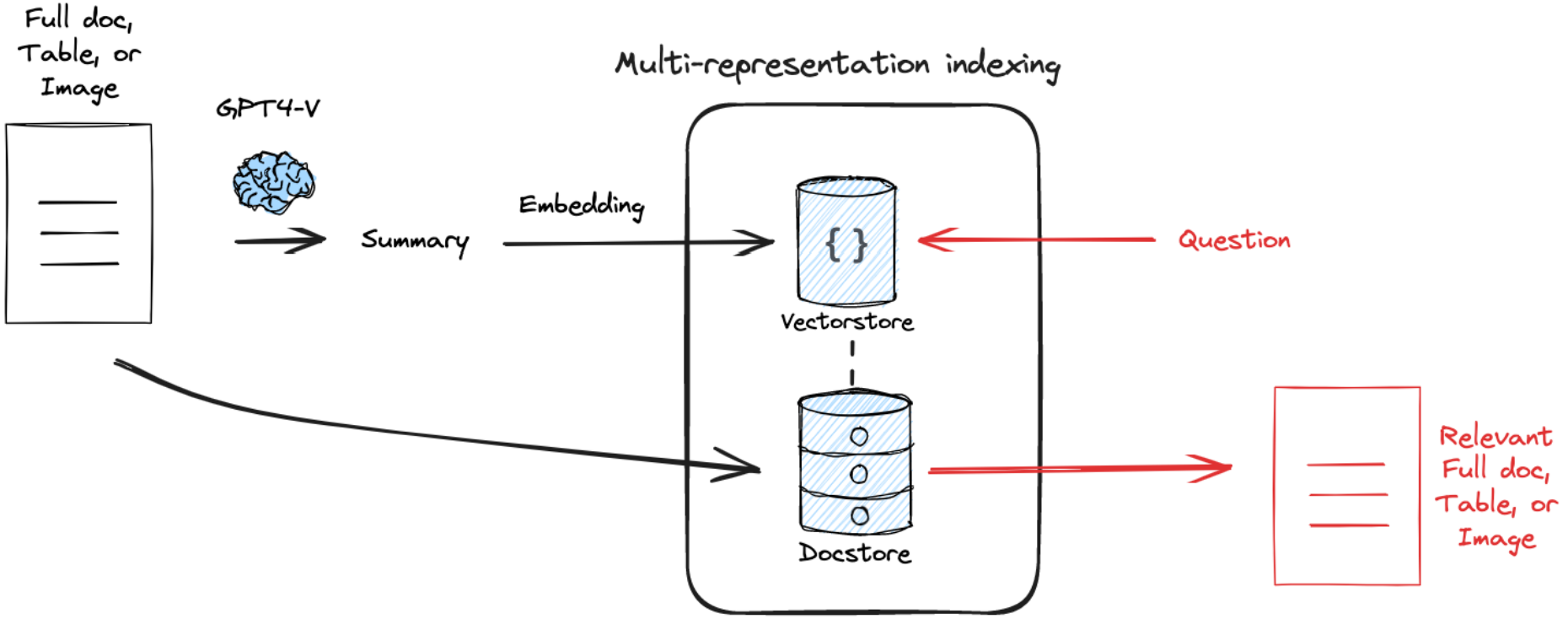


https://www.sbert.net/examples/applications/retrieve_rerank/README.html

Advanced Indexing

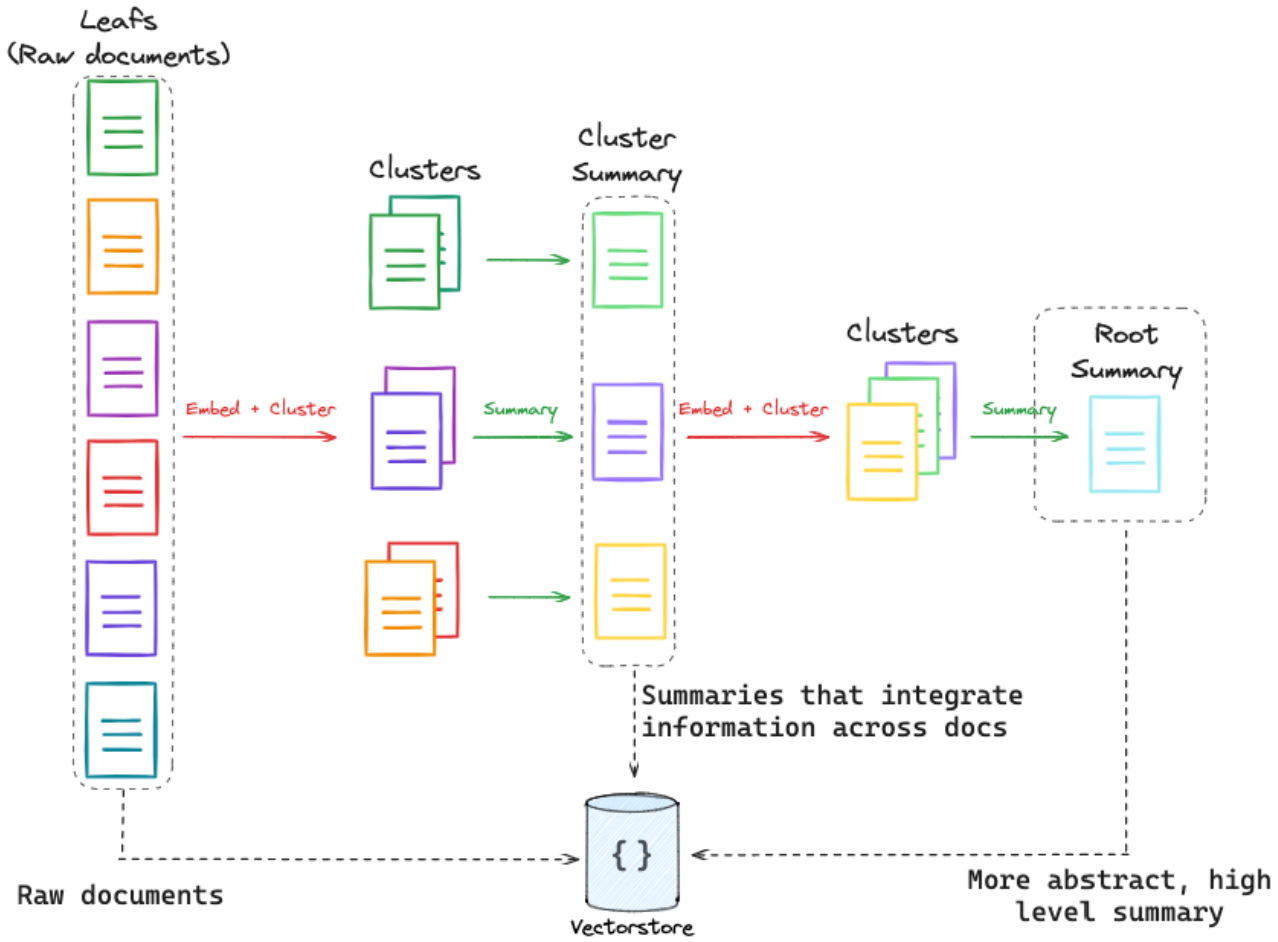


Multi-Representation Indexing



https://github.com/langchain-ai/rag-from-scratch/blob/main/rag_from_scratch_12_to_14.ipynb

RAPTOR



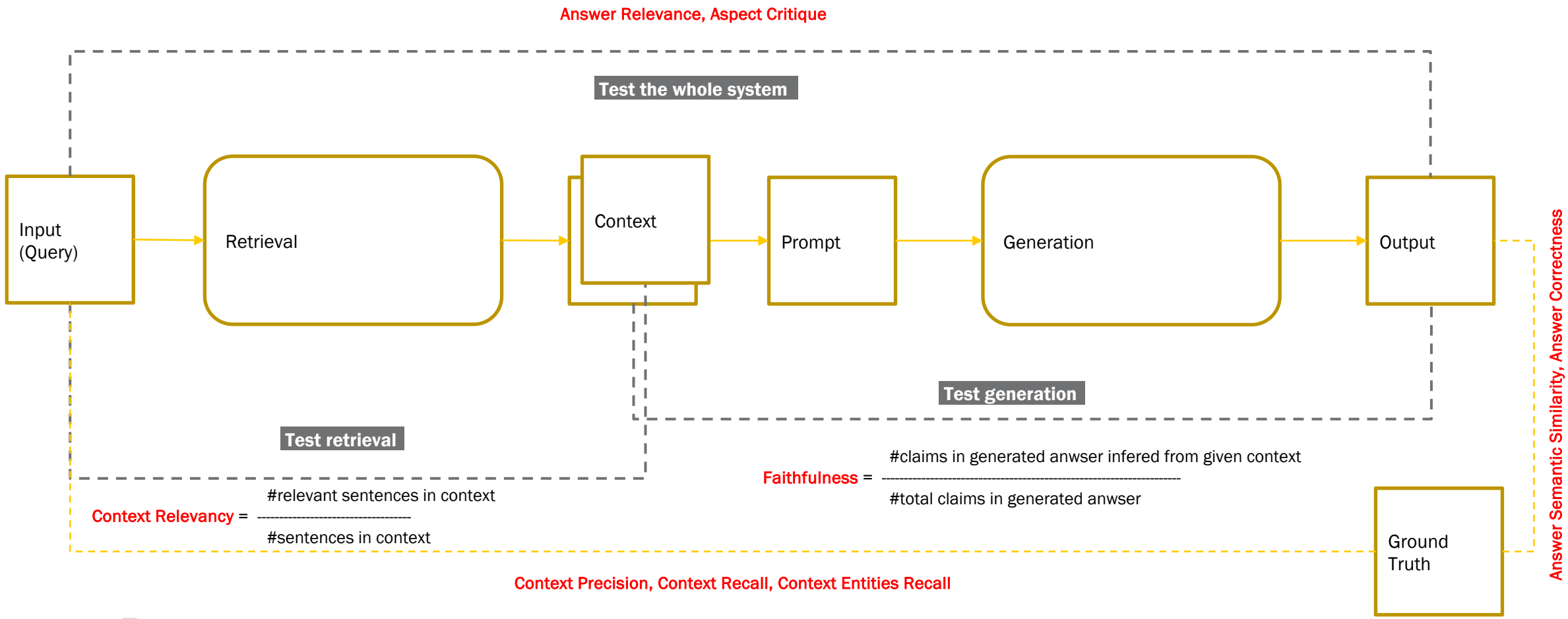
https://github.com/langchain-ai/rag-from-scratch/blob/main/rag_from_scratch_12_to_14.ipynb

INF
Informatik
Hochschule Reutlingen

04
Evaluation & Tuning

Ragas

https://docs.ragas.io/en/latest/



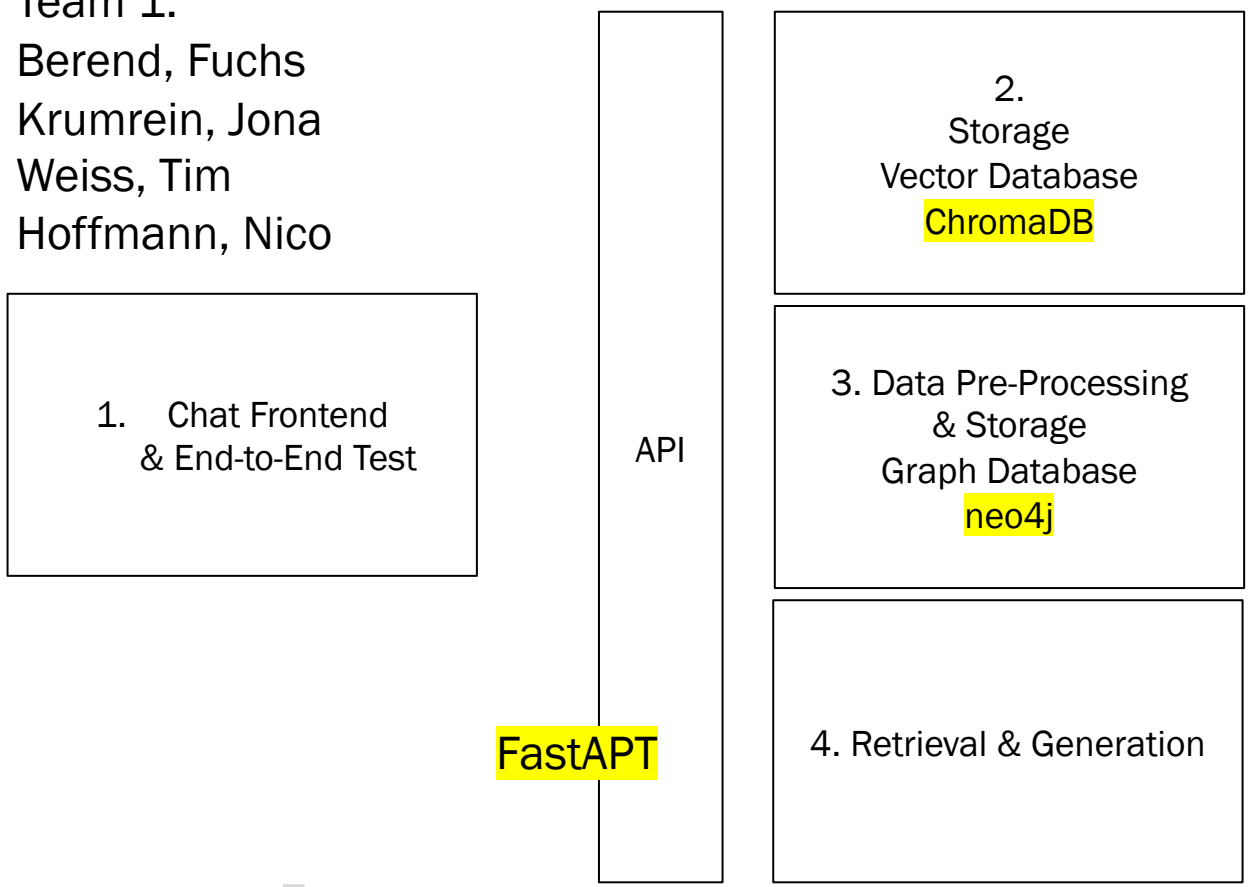
Projektaufgabe

INF
Informatik
Hochschule Reutlingen

Projektaufgabe

- Idee: Wir erstellen als gemeinsames Projekt ein Retrieval Augmented Generation System, das Vorlesungen aus Youtube verarbeitet.

Team 1:
Berend, Fuchs
Krumrein, Jona
Weiss, Tim
Hoffmann, Nico



Team 2:
Buehler, Philipp
Ressler, Malte

Team 3:
Froehner, Thimo
Laib, Lukas
Kraemer, Dominic
Berndt, Nick

Team 4:
Fink, Robin
Doebele, Nico
Zink, Michael

Further Reading

Further Reading

Informatik
Hochschule Reutlingen

- LangChain https://github.com/langchain-ai/rag-from-scratch (old langchain version v02!)
- Retrieval-Augmented Generation for Large Language Models: A Survey https://arxiv.org/pdf/2312.10997
- Large Language Models: A Survey - Chapter „LLMs Are Used and Augmented“: https://arxiv.org/pdf/2402.06196
- Tool Learning with Large Language Models: A Survey: https://arxiv.org/pdf/2405.17935
- RAGAS: Automated Evaluation of Retrieval Augmented Generation: https://arxiv.org/pdf/2309.15217

INF Informatik
Hochschule Reutlingen

Dominik Neumann
Hochschule Reutlingen, Alteburgstraße 150, 72762 Reutlingen
www.reutlingen-university.de
T. +49 172 9861157
dominik.neumann@reutlingen-university.de
dominik.neumann@exxeta.com