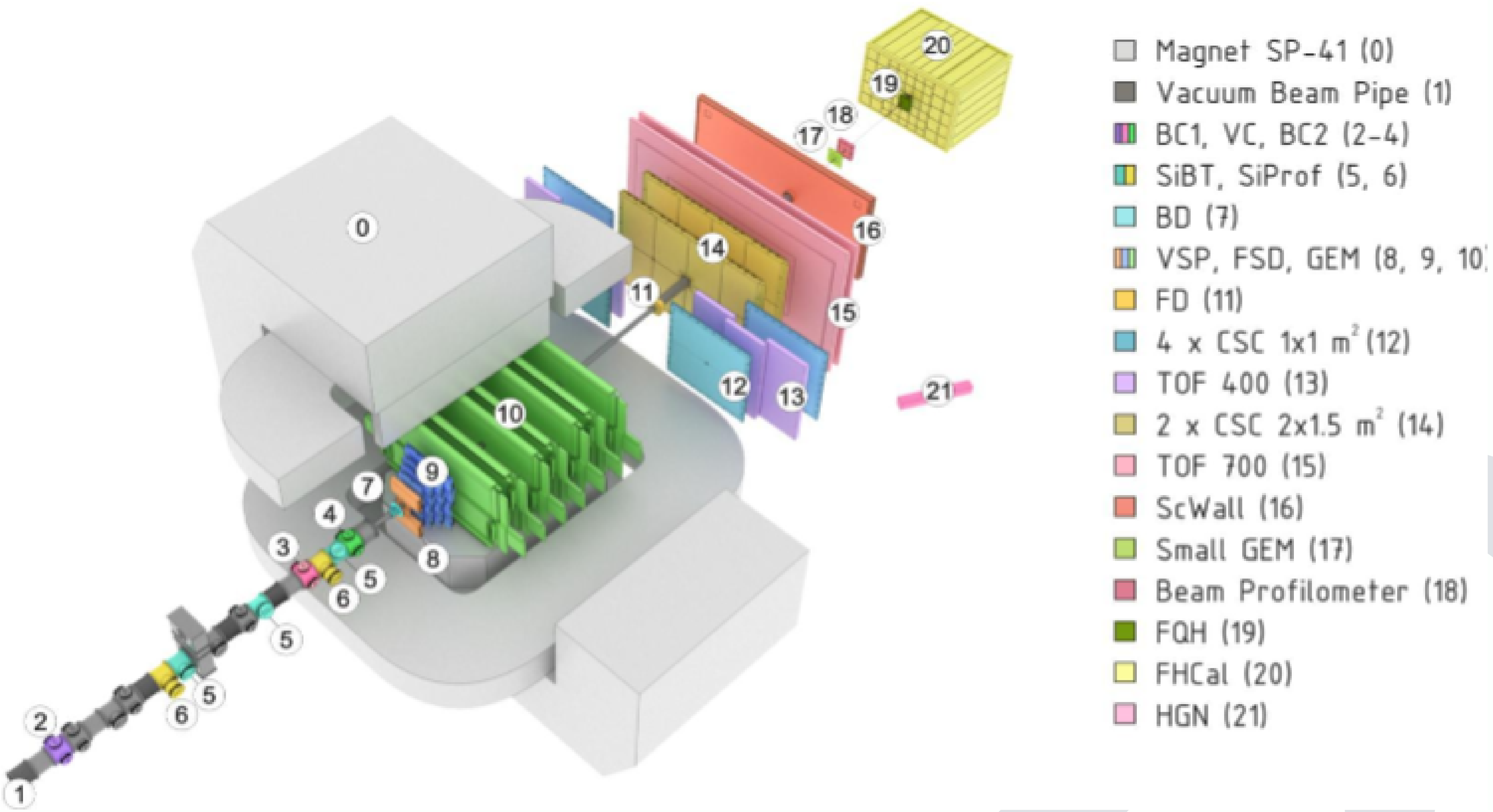
Development of a **virtual assistant** for shift operators of the BM@N experiment

Ilya Romanov, Konstantin Gertsenberger

Introduction

BM@N (Baryonic Matter at Nuclotron) is the first experiment operating at the NICA accelerator complex.

During the experiment session, shift operators manage a complex set of systems, requiring extensive knowledge. Sometimes, they turn to **experts** or **manuals** to resolve issues.



Introduction

Problem

The fragmentation of knowledge required by shift operators leads to **slow decision-making** and **complicates their training process**.

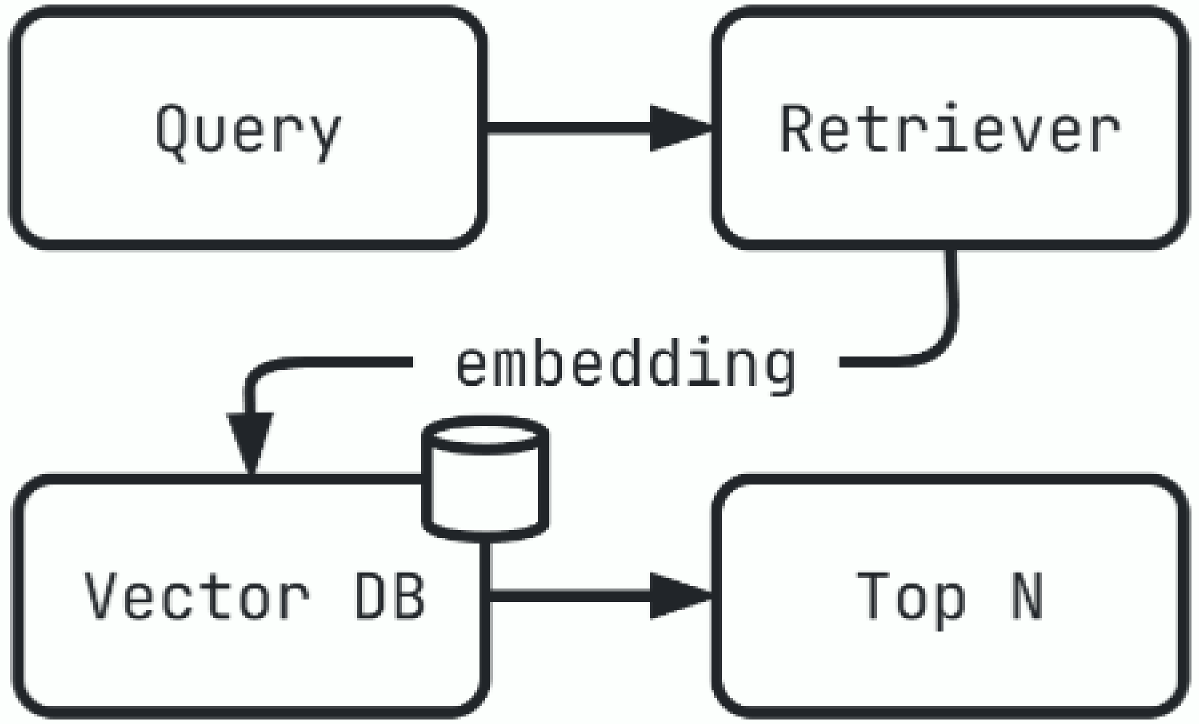
Solution

- conduct a review of existing documentation;
- collect them into a single corpus;
- convert them to a single Markdown format;
- develop an **semantic search engine**.

Semantic search engine

Text retrieval model converts text into **embedding**.

Vector database uses the **HNSW** index to construct a semantic similarity graph between texts and implements search logic.



Searching

Retriever creates an embedding of the query. The vector database finds the top N relevant texts based on a similarity measure, e.g. a dot product.

Malkov, Yu A., and Dmitry A. Yashunin. "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs."

Text retrieval methods

Full-text retrieval

It is based on an **exact match** of key tokens between the query and the documents in the search corpus. **Sparse embeddings** are used for retrieval.

Limitations

- you need to understand what you want to find;
- it is hard to work with synonyms and errors;
- it fails to consider the contextualized meaning of the tokens.

Semantic retrieval

It interprets the **meaning** of tokens taking into account the **context** of their use. **Dense embeddings** are used for retrieval.

Limitations

- it may misinterprets specialized terms, alphanumeric identifiers, or abbreviations to specific domains.

Patil, Rajvardhan, et al. "A survey of text representation and embedding techniques in nlp."

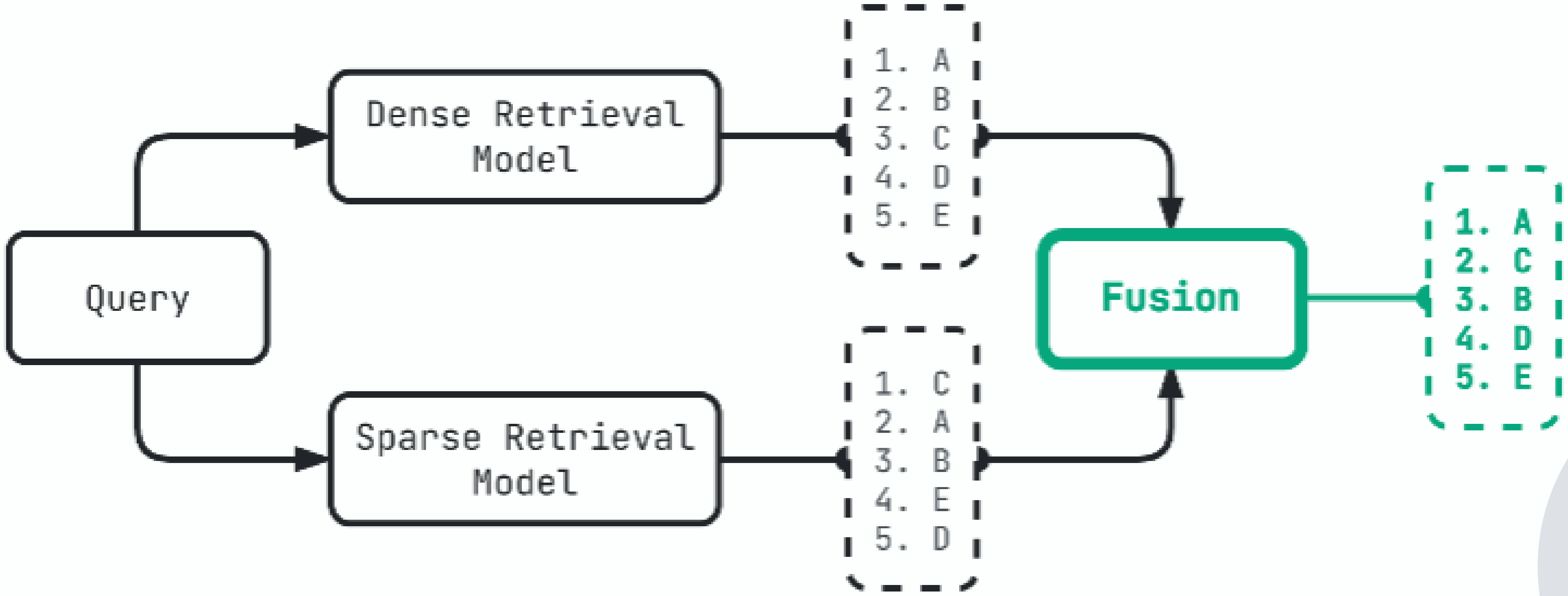
Improving semantic retrieval

Hybrid retrieval is a combination of **semantic** and **full-text** retrieval to overcome the shortcomings of both methods.

There are other methods, but this one offers the best compromise between speed and performance.

$$\mathrm{RRF}(d) = \sum_{r \in R} \frac{1}{k + r(d)},$$

where d — search results element, d — ranking method from set R, k — smoothing parameter (usually k = 60), $r(d)$ — the rank of the element d calculated by the ranking method r.



Cormack, Gordon V., Charles LA Clarke, and Stefan Buettcher. "Reciprocal rank fusion outperforms condorcet and individual rank learning methods."

Selecting the retrieval model

BGE-M3 is modern text retrieval model.

Main features

- large context window (8192 tokens);
- multilingual;
- simultaneous **dense** and **sparse retrieval**.

Model	Max Length	nDCG@10
Baselines (*Prior Work*)		
mDPR	512	16.3
mContriever	512	23.3
$\text{mE5}_{\text{large}}$	512	24.2
bge-large-en-v1.5	512	27.3
$\text{E5}_{\text{mistral-7b}}$	8192	49.9
text-embedding-ada-002	8191	41.1
text-embedding-3-large	8192	51.6
jina-embeddings-v2-base-en	8192	39.4
M3-Embedding (*Our Work*)		
Dense	8192	48.7
Sparse	8192	57.5
Multi-vec	8192	55.4
Dense+Sparse	8192	60.1
All	8192	**61.7**

Table 5: Evaluation on NarrativeQA (nDCG@10).

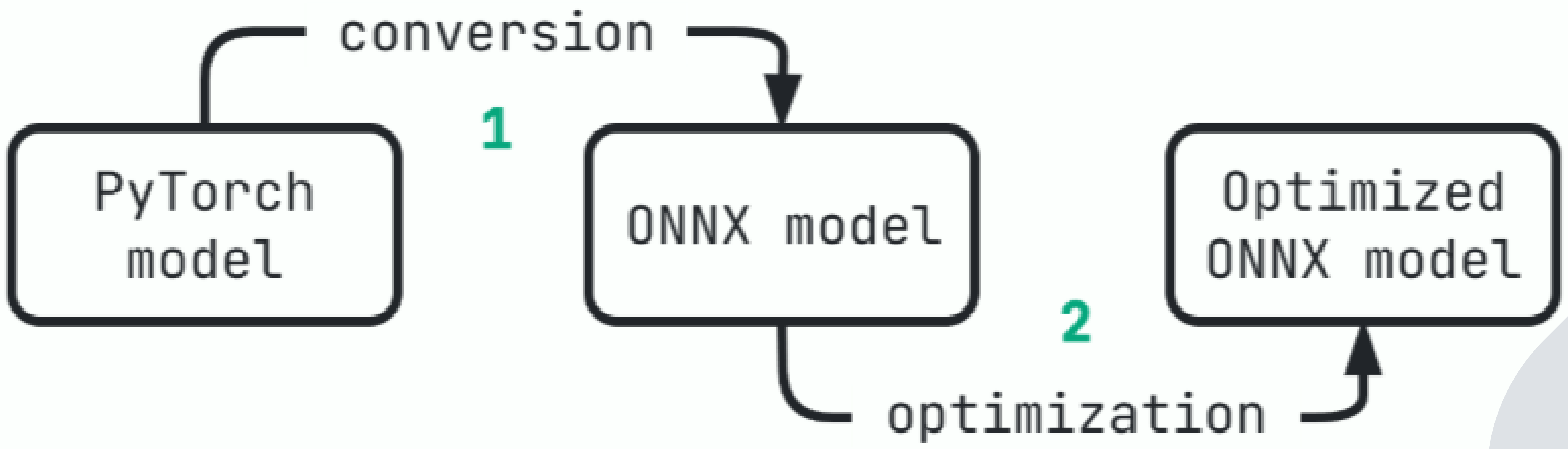
Chen, Jianlv, et al. "BGE M3-embedding: Multi-lingual, Multi-functionality, Multi-granularity text embeddings through self-knowledge distillation."

Preparing the retrieval model

ONNX is the open standard for representing ML models as a **DAG**.

ONNX Runtime is the **high-performance** runtime environment for ONNX models, optimized for many hardware platforms.

1. Convert from PyTorch to **ONNX format**.
2. Apply **O3 level optimization** to Add, MatMul and Attention operators.



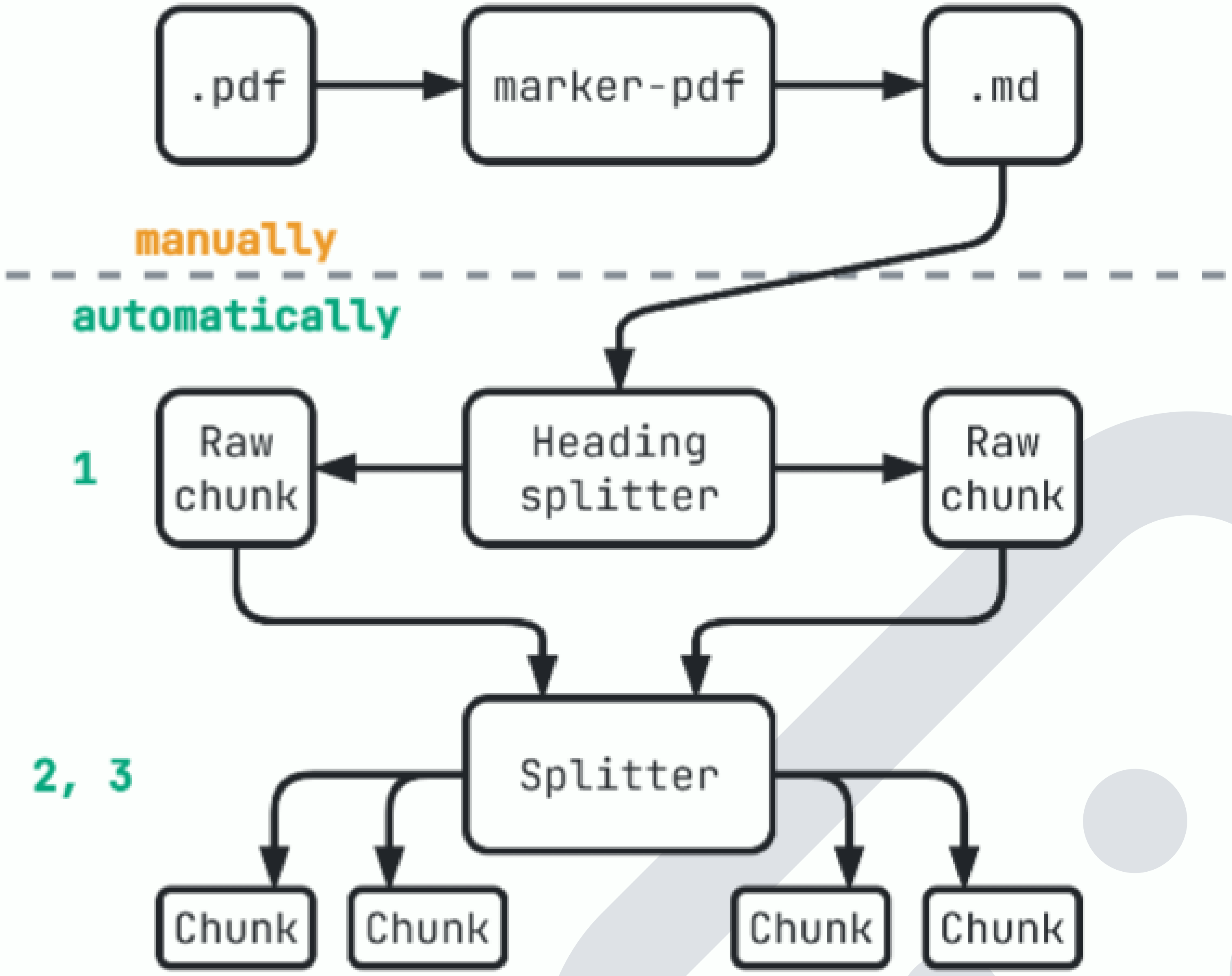
Collecting of the corpus and indexing algorithm

30+ documents were collected.

Conversion to markdown was done using **marker-pdf**.

Splitting process

1. Split by headings to get raw chunks.
2. Split each raw chunk into chunks of up to 200 tokens, with an overlap of 128.
3. Chunks are supplemented with headings info.

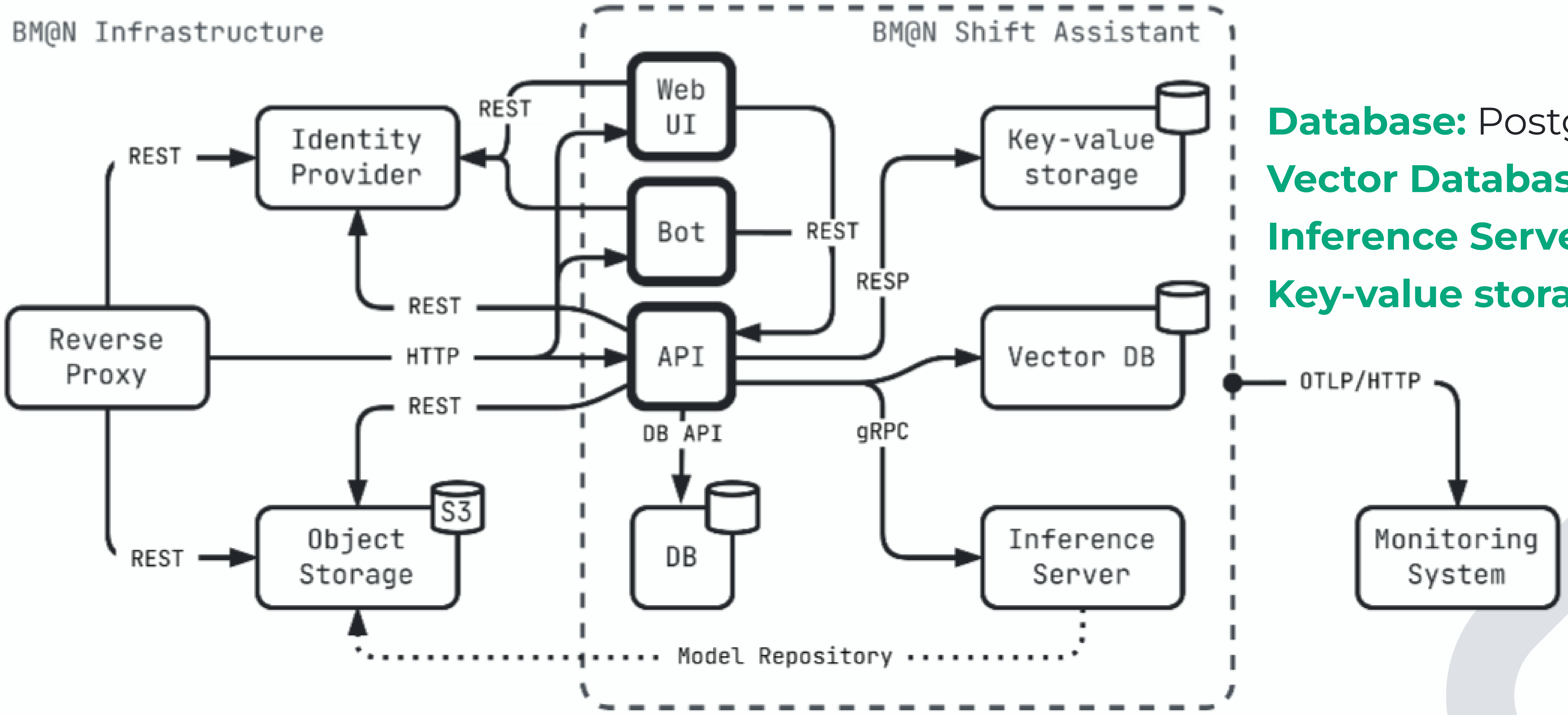


System architecture

API: FastAPI, Pydantic, SQLAlchemy, Taskiq, Logfire.

Telegram Bot: Aiogram.

Web UI: Vite, React.



Database: PostgreSQL.

Vector Database: Qdrant.

Inference Server: NVIDIA Triton.

Key-value storage: Valkey.

System features

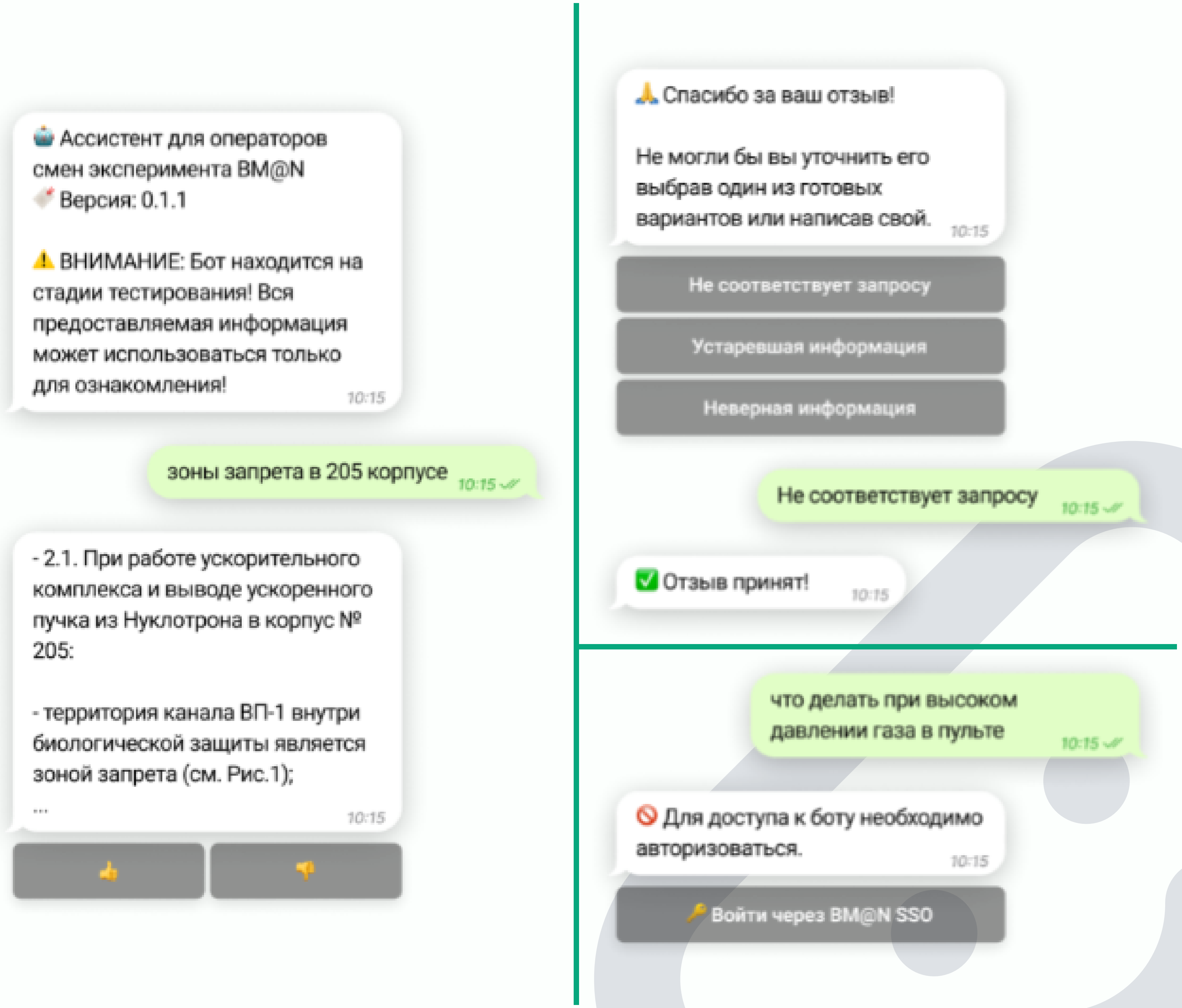
1. Semantic search in the knowledge base.

2. Search results are sorted by relevance and contain a link to the source.

3. Dual Interface: Telegram Bot and Web UI.

4. Access control via SSO.

5. Feedback system and request logging for system monitoring and knowledge base improvement.



Evaluation

The **artificial dataset** was created including questions on documents and test tasks.

Evaluation was performed using the **Mean Reciprocal Rank** (MRR) metric, using the first 5 fragments.

The result was MMR@5 = **0.68** — the relevant answer ranks in the **2nd position** on average.

$$\mathrm{MRR}(d) = \frac{1}{|U|} \sum_{u \in U} \frac{1}{r_u}$$

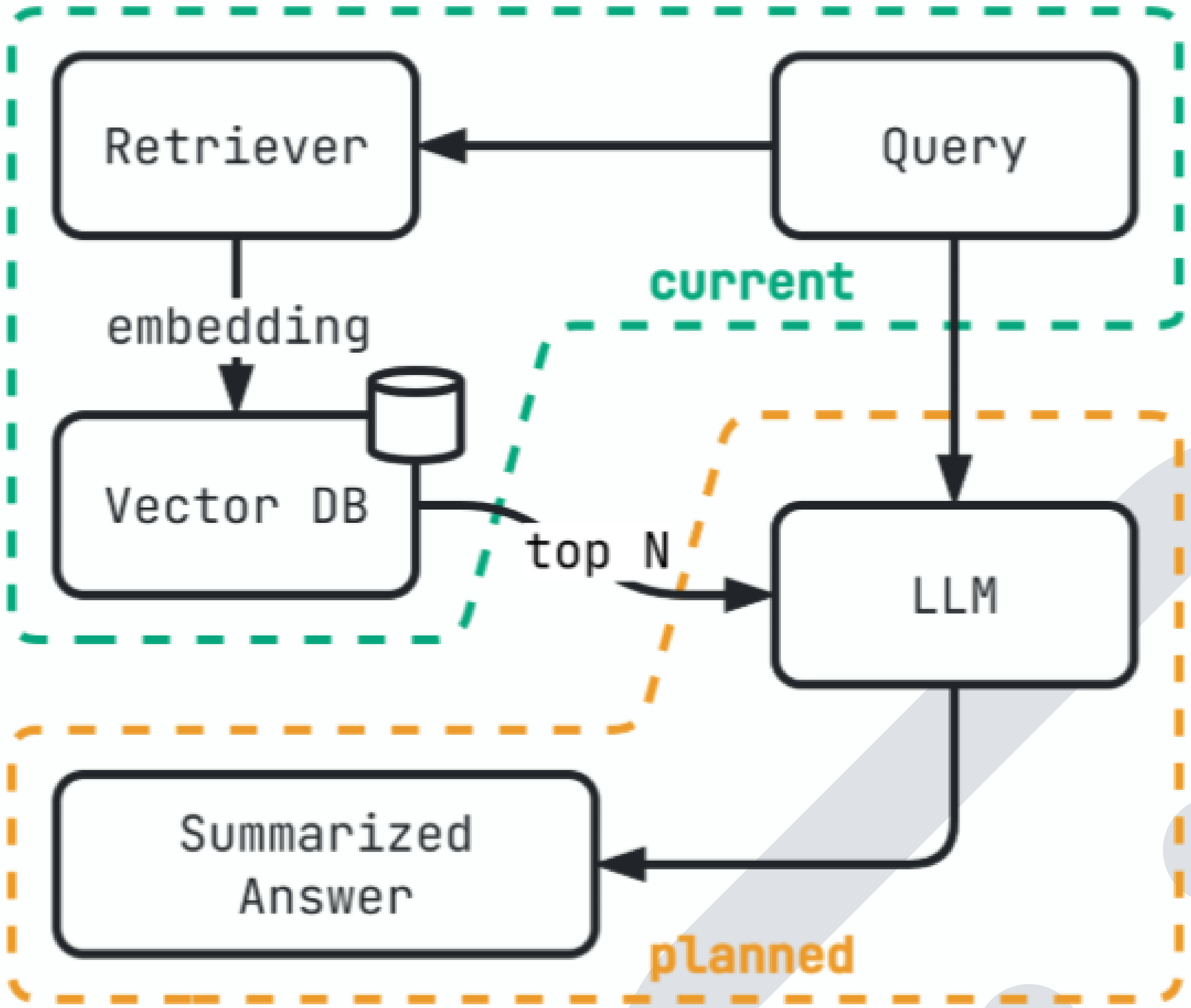
where $U = \{u_1, \cdots, u_n\}$ is the set of relevant fragments, r_u is the rank of the fragment in the search results.

Conclusion

The virtual assistant for BM@N experiment shift operators is currently a **semantic search engine**. A transition to a **Retrieval Augmented Generation** (RAG) architecture is planned for the future.

The **initial search corpus** has been compiled and indexed.

An initial evaluation confirmed the system's ability to **find relevant information** within the experiment's terminology.



Thank you for **your attention**!