

AUTOMATED BOE QUERY PROJECT

JAVIER MANCHO



REPOSITORY

[https://github.com/javiermancho/
HPE-CDS-LLM](https://github.com/javiermancho/HPE-CDS-LLM)

INDEX

1 Project Definition

2 Components

3 Use Cases

4 Demo

5 Next Steps Roadmap

6 Conclusion

Project Definition

Introduction

- Develop a microservices architecture to automate the download of Boletín Oficial del Estado (BOE) files.
- Integrated a Large Language Model (LLM) with Retrieval-Augmented Generation (RAG) to enable intelligent querying of these documents, with date filtering capabilities.

Scope

- Design and implement microservices for downloading, storing, and indexing BOE files.
- Enable document search with date filters via a user-friendly interface.

Objectives

- Build a microservices-based system for BOE file management.
- Implement LLM with RAG for intelligent document queries.
- Develop a user interface for date-filtered searches.

Components



Client

- Built using NextJS.
- Allows the user to update the database to the current date.
- Includes a chat feature that enables queries.



Server

- Built using SpringBoot and Gson.
- Coordinates communications between microservices.
- Performs queries to the database and the LLM.



Database

- Built using Flask and ChromaDB.
- Obtains chunks by splitting the original text using Recursive Splitting.
- Stores chunks along with their embeddings, IDs, and date.
- Returns the three documents closest to the query.

Components



LLM

- Built using Flask and Llama-cpp.
- Its core is based on a Llama 2 model.
- Generates responses based on context provided by the server and the database.



Embeddings

- Built using Flask and LangChain.
- Returns the calculated vectors.
- Uses Jina embeddings model fine-tuned for Spanish.

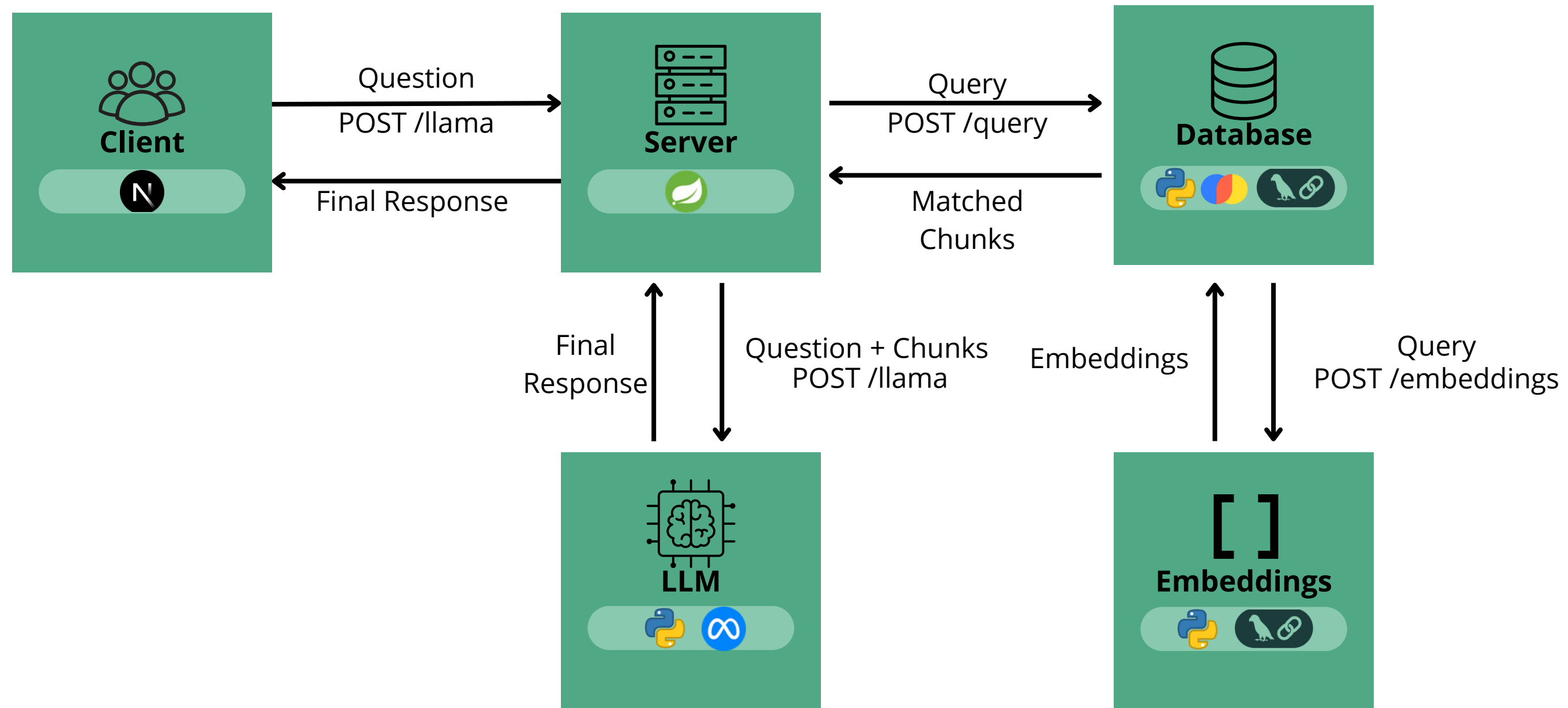


BOE-Script

- Built using Flask and Requests.
- Queries the BOE API to retrieve official gazettes (BOEs).
- Stores the PDFs and sends the text to the database to store chunks.

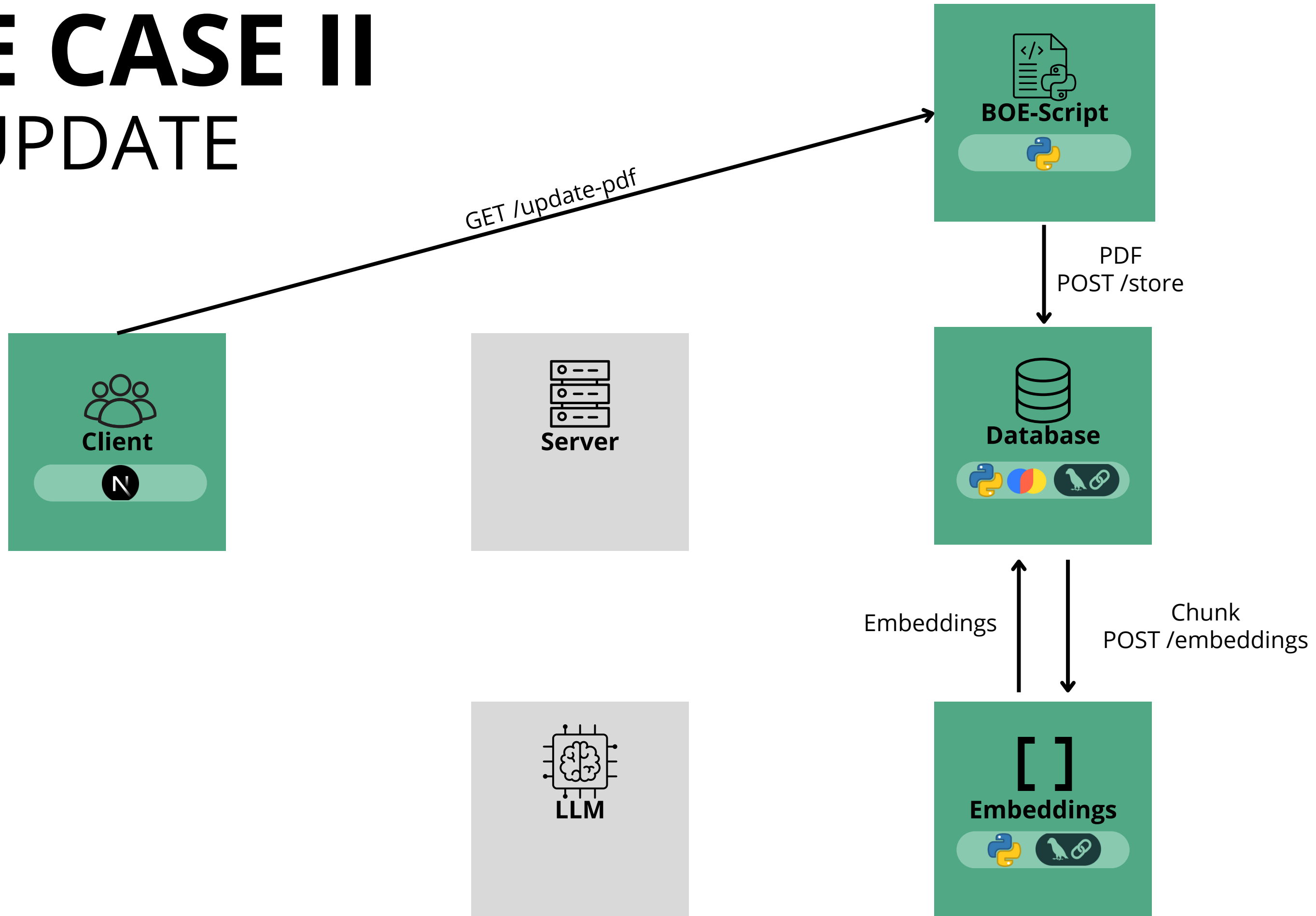
USE CASE I

USER REQUEST



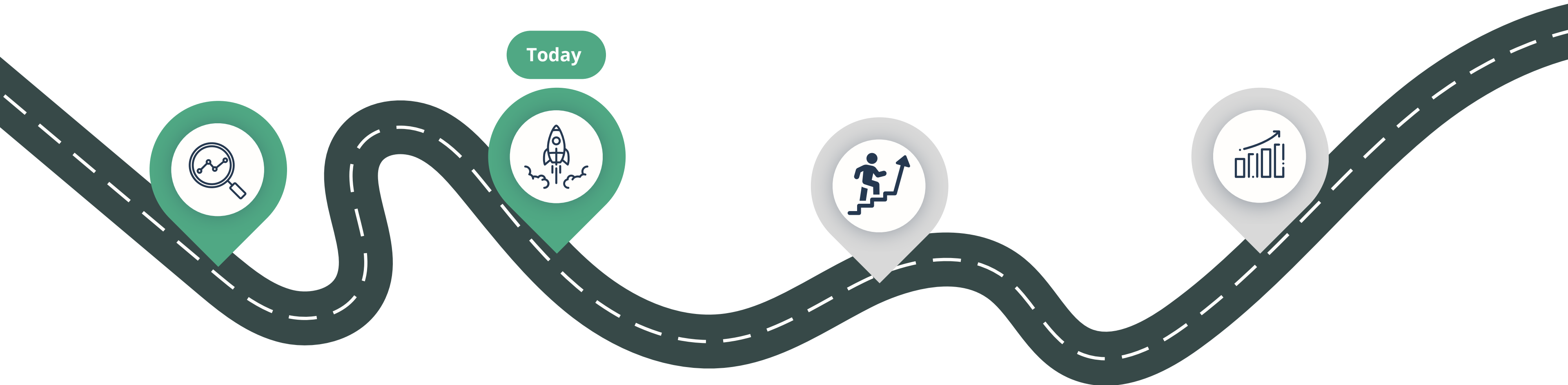
USE CASE II

DB UPDATE



DEMO

Next Steps Roadmap



Analysis Phase

Problem Statement and Requirements.
This phase includes the design of functionalities.

Release 1.0 - MVP

Delivery of the **first release**. The project meets the minimum functional requirements.

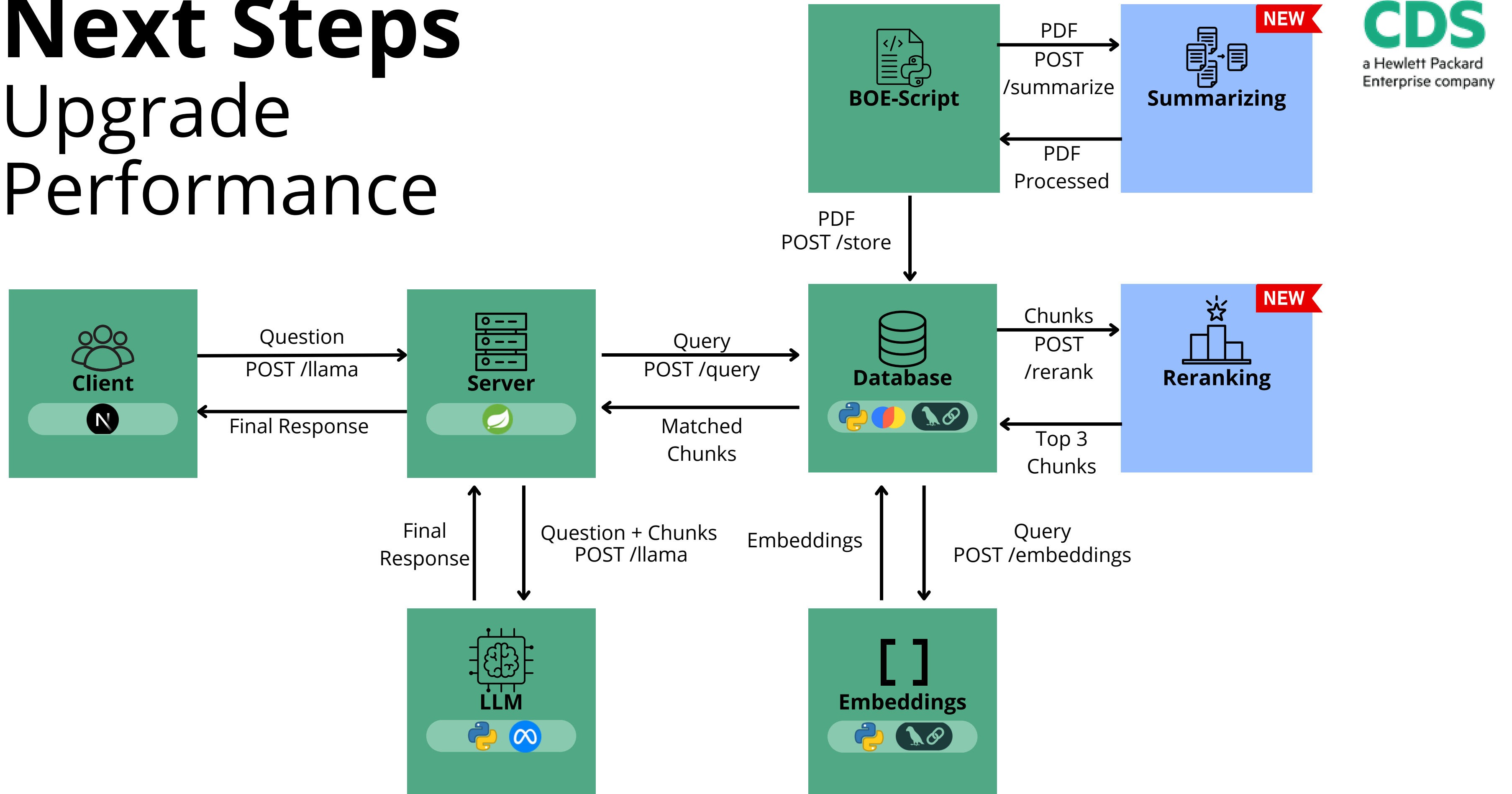
Upgrade Performance

Data preprocessing functionalities, such as **summarizing**, would be added.
Chunk **reranking** would be implemented to improve context accuracy.

Scaling

Add **new functionalities** such as filtering by autonomous communities or ministries.
Release a new, improved UI version to accommodate users.

Next Steps Upgrade Performance



Conclusion

- Data **preprocessing** is crucial for **enhancing the effectiveness** of our system.
- With a model **fine-tuned for Spanish**, It would be achieved **higher-quality results**.
- Performance could be improved by storing **additional metadata** in the database.

AUTOMATED BOE QUERY PROJECT

JAVIER MANCHO



REPOSITORY

[https://github.com/javiermancho/
HPE-CDS-LLM](https://github.com/javiermancho/HPE-CDS-LLM)