

Master Thesis and

End-of-Studies Internship Report

Daniel Stulberg Huf

daniel.stulberg-huf@student-cs.fr

Promotion 2024
Mention Architecture des Systèmes Informatiques
Filière Conception de Systèmes Complexes
Master Big Data Management and Analytics

AI for ESG: Development and Industrialization of a Full Stack RAG Application for ESG Monitoring

AI for ESG : Développement et Industrialisation d'une Application RAG Full Stack pour le Suivi ESG

Internship Mentor:	Jérôme Feroldi	-	Emerton Data	jerome.feroldi@emerton-data.com
Mention Tutor:	Nacéra Seghouani	-	CentraleSupélec	nacera.seghouani@centralesupelec.fr
Filière Tutor:	Xavier Mouton	-	Renault	xavier.mouton@renault.com

Internship duration: 6th May 2024 - 4th November 2024
Defended on: 4th November 2024

Acknowledgments

These last two years of living abroad have felt unreal. I never imagined that I would get to meet so many incredible people, discover so many new places, and live so intensely in such a short period of time.

Firstly, I would like to thank my supervisors, Nacéra Seghouani and Xavier Mouton from CentraleSupélec, and Jérôme Feroldi from Emerton Data, for their thorough support throughout my Master's studies and my internship.

I would like to express my heartfelt gratitude to my parents for the education they have imparted throughout my entire life, as well as for their encouragement and unconditional support, even from afar, in my continuous search for enriching experiences.

Finally, I would like to thank my long-time friends from Brazil and the friends I made during my stay in France for sharing such great memories and endless lessons that I will carry with me throughout my lifetime.

Embarking on this journey, full of uncertainties and challenges to overcome, would not have been possible without these people, to whom I will always be grateful.

"Pouco importa o objeto da ambição; ela vale por si, independente do alvo. Sempre necessitamos ambicionar alguma coisa que, alcançada, não nos faz desambiciosos."

— Carlos Drummond de Andrade

Contents

List of Figures

List of Tables

List of Algorithms

École CentraleSupélec

Summary of the end-of-studies internship report

FILIÈRE: CSC MENTION: ASI MASTER: BDMA PROMOTION: 2024

LAST NAME and First Name of the Student: STULBERG HUF Daniel
NAME and address of the Company: EMERTON DATA, 16 Av. Hoche, 75008 Paris
Internship topic: AI for ESG: Development and Industrialization of a Full Stack RAG Application for ESG Monitoring

Abstract: This thesis presents the development and industrialization of a Generative AI product employing Retrieval-Augmented Generation (RAG) to assist a major French conglomerate in answering questions about their Environmental, Social, and Governance (ESG) reports. By implementing advanced RAG techniques, the performance of a naive RAG pipeline was enhanced. A full-stack web application was developed, providing authenticated users with an interface to submit queries and receive AI-generated answers sourced from an ESG knowledge base. The application was deployed using a CI/CD pipeline integrated with Azure infrastructure. An evaluation using the RAGAS framework demonstrated high levels of answer generation quality for different metrics. This work not only delivered a valuable tool for the client but also contributed to the broader understanding of deploying full-stack GenAI technologies in industrial settings.

Keywords: *Generative AI, Retrieval-Augmented Generation, Environmental, Social and Governance, Full-stack web application, Azure infrastructure, RAGAS framework*

. . . . . .

Résumé: Cette thèse présente le développement et l'industrialisation d'un produit d'IA générative utilisant la Génération Augmentée de Récupération (RAG) pour aider un grand conglomérat français à répondre à des questions sur leurs rapports Environnementaux, Sociaux et de Gouvernance (ESG). En mettant en œuvre des techniques RAG avancées, les performances d'un pipeline RAG naïf ont été améliorées. Une application web full-stack a été développée, offrant aux utilisateurs authentifiés une interface pour soumettre des requêtes et recevoir des réponses générées par l'IA à partir d'une base de connaissances ESG. L'application a été déployée en utilisant une pipeline CI/CD intégrée à l'infrastructure Azure. Une évaluation utilisant le cadre RAGAS a démontré des niveaux élevés de qualité de génération de réponses pour différents métriques. Ce travail a non seulement fourni un outil précieux pour le client, mais a également contribué à une meilleure compréhension du déploiement des technologies GenAI full-stack dans des environnements industriels.

Mots clés: *IA générative, Génération Augmentée de Récupération, Environnement, Social et Gouvernance, Application web full-stack, Infrastructure Azure, Cadre RAGAS*

Date: 04/11/2024

Chapter 1

Introduction

1.1 Context

The rapid advancement in Generative Artificial Intelligence (GenAI) technologies has radically changed the landscape of software development and industrial applications. GenAI, with its capability to synthesize new content from existing data, offers immense potential in various domains including text, image, audio, and video generation.

The use of such technologies has the potential to increase operational efficiency and deliver personalized products for players in many different industries. However, deploying GenAI pipelines in real-world scenarios brings in challenges that require a full-stack approach, encompassing everything from infrastructure to application layers [2].

Technologically and financially, the deployment of a GenAI product requires robust infrastructure, including high-performance computing resources, data storage solutions, and machine learning operations (MLOps) frameworks. Security and compliance also play important roles, since clients' sensitive data is being constantly handled in the entire operation chain [7]. From a managerial perspective, the integration of these technologies translates to the upskilling of managers, tech leads, data scientists and software developers to work together for delivering a complete product that meets all business requirements.

This thesis deep dives into the process of developing and industrializing a GenAI product using the Retrieval-Augmented Generation (RAG) process in a full-stack manner. The objective of this internship is then to explore the scientific and architectural dimensions of the RAG within the context of Emerton Data, the AI and Tech unit of Emerton Group, a top-tier Strategy and Transformations Consulting firm. In this sense, not only the tasks performed during the internship are aimed to accelerate the data transformation of Emerton Data's clients by launching an innovative GenAI product, but also to provide global insights that can be applied in general industrial practices.

1.2 Mission Objectives

Sphinx (real name hidden for confidentiality purposes) is a major French conglomerate operating in the sectors of Construction, Telecommunications, and Energy. Recently, Sphinx has been facing challenges related to Environmental, Social, and Governance (ESG) concerns. The monitoring of ESG indicators impacts all major divisions of Sphinx, affecting the entire professional chain from suppliers and investors to clients and collaborators. In such a dynamic environment, the Data & AI team of Sphinx has identified several high-value use cases where GenAI tools can automate the search for information. Among these, assistance with responding to forms, particularly on the ESG theme, has been prioritized. These forms would ideally retrieve relevant information from the organization's sustainability reports without needing to manually read them one by one. Such reports assess the

environmental effects, social responsibilities, and commitment to governance standards of Sphinx, with the aim of providing transparency in its operations.

A preliminary Make or Buy analysis revealed that existing tools on the market are not well-suited to this specific need as they lack custom features such as handling parallel forms at scale, creating forms in multiple languages, and allowing for collaborative work in the making of a form. In this context, Sphinx wishes to capitalize on the expertise of Emerton Data through a product that can meet the case guidelines for automating responses to ESG questions. The project includes three main dimensions:

1. **Generative AI**: A complete Retrieved-Augmented Generation (RAG) pipeline capable of processing a question from a client, searching the context of the question within an ESG knowledge base sourced from various materials (press releases, public information available on the company's website, confidential data), and feeding the question along with the relevant context into a Large Language Model (LLM) to generate a pertinent final answer.

2. **User Interface**: A web application that allows authenticated users from Sphinx to perform queries with one or more questions and receive responses from the RAG pipeline.

3. **Infrastructure**: A functional Continuous Integration / Continuous Delivery (CI/CD) framework between GitHub and Azure for deploying and updating the codebase of the application.

1.3 Personal Contribution

As a Software Engineer intern at Emerton Data, I play an active role in the entire innovation chain required for the mission I was assigned to: from defining the needs and requirements of the project to smartly designing the solution, demonstrating a proof of concept, and ultimately launching it into production with ongoing evolutionary maintenance. In the specific context of the mission described above, I will focus on two complementary roadmaps.

1. **Research**: The first roadmap resolved into a scientific investigation on advanced techniques for RAG implementation. Specifically, for this thesis, the goal is to improve the performance of a "naive RAG" pipeline by utilizing novel approaches such as recursive chunking, Maximal Marginal Relevance for similarity search, and semantic caching, and analyzing the performance of the generated answers according to standardized evaluation metrics. The final deliverable for this roadmap will be a proof of concept (PoC) that includes the implementation of these approaches. The main challenge in this case consists of preprocessing the documents provided by the client so that they can be ingested into the RAG pipeline for optimal answer generation, as well as making the good architectural choices for the pipeline to work in real-time when put to production.

2. **Full stack development**: The second roadmap resolved into working jointly with other software engineers from Emerton Data to develop a complete web application

that retrieves a question from a user, feeds it to the RAG pipeline, and displays the generated answer. The application should comprehend several usecase scenarios, such as being able to submit multiple questions at the same time, saving answers as "golden answers", and consulting the knowledge base of ingested documents. The application will then be integrated into production and deployed in the cloud using GitHub Actions and Azure tools. The final deliverable for this roadmap is a functional product ready to be used by the Data & AI team of Sphinx. The main challenge in this case resides in meeting all of the business and design requirements requested by the client while limited to computing and budgeting constraints, as well as safekeeping the client's private data that is being fed and generated during the use of the application.

1.4 Final Perspectives

On a personal level, this thesis and internship aim to enhance both my technical skills in full-stack development and Generative AI, and my project management capabilities by working with a team of developers, consultants, tech leads, and clients to deliver smart solutions transparently and consistently.

Finally, the impact of this work lies in its potential to make the scientific analysis and conceived frameworks reusable in future missions that Emerton Data may undertake. Beyond that, by developing such a robust product, this thesis contributes to the growing body of knowledge on how to effectively integrate full-stack GenAI technologies into enterprise operations, thereby driving innovation and competitive advantage in such a fast-paced environment.

Chapter 2

Related Work

2.1 Overview

This section explores the application of Generative AI technologies, with a particular focus on the RAG approach, in both corporate and academic settings, emphasizing their role in addressing challenges within the ESG domain. The discussion begins with an overview of the adoption of Generative AI in the development processes of large corporations. Following this, state-of-the-art approaches in ESG-related LLM applications are presented. Finally, the section highlights some use cases of RAG implementations integrated with cloud-based infrastructures. The aim is to evaluate the potential implications of these approaches for the project at hand.

2.2 Literature Review

Generative AI has garnered significant attention in recent years, establishing itself across various industries. According to the McKinsey Global Survey on AI in 2024, 65% of companies worldwide have deployed GenAI tools, representing nearly a twofold increase compared to 2023 [22].

The panorama of GenAI adoption in France looks even more promising. Emerton Data's white paper, "The Real Deal of Generative AI," states that all French companies surveyed in their poll have industrialized GenAI use cases [13]. For example, Qwant, a French search engine provider, recently introduced a feature for summarizing pages in search results using generative AI. Larger groups like Safran, Sodexo, and Total Energies have deployed off-the-shelf solutions such as custom chatbot assistants, now available to thousands of their employees.

One of the potential scenarios in which developers from large companies can benefit from the use of AI is in the analysis of ESG reports. Given the rapid expansion of available information in the field, researchers have been implementing natural language processing techniques to improve the accuracy and efficiency in which ESG data is extracted and analyzed. For example, [21] employed BERT-like architectures to automate the classification of paragraphs from ESG reports in line with Global Reporting Initiative (GRI) standards, improving their existing framework for assessing ESG factors at the Brazilian Development Bank. In another study, [19] identified historical trends in ESG discussions by examining the transcripts of corporate earning calls. The analysis of the linguistic structure of ESG-related texts in corporate reports has shown that ESG factors are integral to business strategies. Additionally, [16] have created the custom *ClimateQA* model, which leverages NLP techniques to identify climate change information within financial reports based on a question answering approach.

In the specific domain of GenAI, Large Language Models have been increasingly applied to text analytics in the ESG sector. For instance, [17] introduced the *SDG Prospector*, a tool that leverages LLMs to pinpoint paragraphs in Public Development Banks' sustainability reports that correspond to the seventeen Sustainable Development Goals (SDGs) outlined in the United Nations' 2030 Agenda for Sustainable Development. Similarly, [4] used LLMs to extract structured insights concerning ESG-related information from companies' sustainability reports. Their analysis revealed similarities across different companies in terms of ESG actions. Moreover, [18] developed *ChatReport*, an LLM-based system designed to automate the analysis of sustainability reports while reducing the occurrence of hallucinations and introducing ESG-experts in the development loop.

To overcome challenges related to lengthy context and concurrent data in LLMs, researchers have integrated the Retrieval-Augmented Generation (RAG) paradigm with LLMs, enhancing the precision of responses through dynamic information retrieval [15]. For example, [24] applied the RAG approach to enrich GPT-4 by feeding context retrieved from the Sixth Assessment Report of the United Nations Intergovernmental Panel on Climate Change (IPCC AR6). The authors developed a conversational agent based on their method [6], and demonstrated the ability to provide accurate answers to challenging climate-related questions. Similarly, [3] employed RAG technology to evaluate the ESG factors in Turkish sustainability reports from 10 companies listed in the BIST Sustainability-25 index, resulting in the generation of 47 prompts related to ESG criteria.

Finally, the landscape of RAG has also experienced significant growth in cloud-based implementations, particularly with Azure services. For instance, [1] implemented a RAG-based Teams chatbot for a private organization using Microsoft Azure's AI services and deployed the application within the organization's Azure environment. Similarly, [11] developed a virtual assistant prototype for resolving housing disputes, powered by LLM technologies and RAG on the Azure platform. This product integrates the LLM Azure OpenAI 3.5 turbo model and Azure AI search for the RAG approach, with cloud resources primarily managed through Azure's command-line interface. The adoption of RAG in cloud-based infrastructures has grown not only in academic settings but also within large corporations. As suggested by Modulai, a Swedish machine learning and AI consultancy, RAG systems can be integrated into cloud infrastructures like AWS, Azure, and Google Cloud through a service-oriented architecture, allowing the product to be tailored to different organizational needs — whether scaling up for large enterprises or providing precision for niche markets [12].

2.3 Conclusion

Building on the foundation of existing research, this thesis seeks to employ the latest state-of-the-art techniques for RAG implementation in combination with full-stack development best practices to build and deploy an off-the-shelf product for our client. The work presented in this report distinguishes itself from existing approaches by (i) enabling the RAG pipeline to be parallelized and managed by the user at each single step through a visual interface, (ii) processing questions and answers written in two different languages (English and French), and (iii) deploying the solution within a cloud architecture using an "as-code only" approach.

Chapter 3

Background

3.1 Overview

This chapter provides a synthesis of the methods and tools needed to understand the scientific and architectural implementations explained in following chapters. The choice for each model and framework is also going to be highlighted.

3.2 RAG pipeline

In the field of Artificial Intelligence, LLMs form the foundation for advanced chatbots and other natural language processing (NLP) applications. These models are capable of answering user questions across a range of topics by referencing authoritative sources of knowledge. However, the static nature of LLM training data, which is based on a limited, often insufficient knowledge base for industrial applications, introduces certain challenges. LLMs can sometimes produce unpredictable responses, including generating incorrect information when they lack relevant data, referencing non-authoritative sources, or misunderstanding terminology, leading to inaccurate answers. Such misleading behavior is better known as an hallucination.

To address these limitations, a technique called Retrieval Augmented Generation (RAG) can be employed. Such technique was first introduced in 2020 by the Facebook AI Research team, University College London and New York University [15]. RAG enhances the capabilities of LLMs by directing them to fetch relevant information from predefined, authoritative knowledge sources. This approach gives organizations better control over the generated content and provides users with transparency regarding the origins of the responses. Essentially, RAG combines a knowledge base search with LLM prompting. When a user poses a question, the model retrieves pertinent information using a search algorithm and uses this context to formulate its answer. Both the query and the retrieved context are included in the prompt sent to the LLM, ensuring more accurate and reliable responses. A scheme of a naive RAG implementation can be seen in figure 3.1.

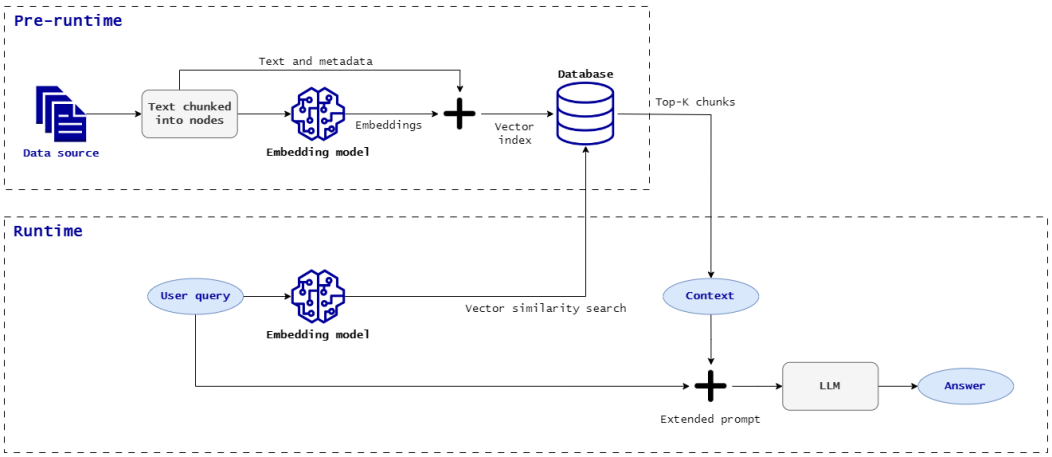


Figure 3.1: Naive RAG pipeline

The steps pictured above are described as follows:

1. **Text Chunking**: The initial step is to read one or more text files and split each one of them into smaller pieces referred to as chunks. This chunking process simplifies the handling of individual concepts within the text.

2. **Chunk Embedding**: Before runtime, an embedding model converts each text chunk into vector representations using a Transformer Encoder model. The embeddings of the chunks, along with their corresponding text representations, are stored in a database for later use.

3. **Query Embedding**: During runtime, the query is embedded into a vector. It is important to use the same embedding model for both the text chunks and the input query to ensure that all embeddings share the same vector space, maintaining semantic consistency.

4. **Similarity Search**: The vectorized query is then compared with the chunk embeddings (retrieved from the database) to identify the most relevant chunks. The comparison is based on similarity scores calculated between the query embedding and each chunk embedding individually.

5. **Prompt Creation**: The corresponding text chunks for the top-K embedded chunks with the highest similarity score are combined with the initial query. This prompt is designed to provide the LLM with sufficient context to generate a potentially meaningful response.

6. **Answer Generation**: In the final stage, the extended final prompt is fed into the LLM, which uses the provided context to generate a response that potentially answers the query in the most accurate way possible.

In each of those steps, several hyperparameters can be identified for optimizing the overall efficiency of the RAG pipeline. For example, determining the ideal chunk size for a given use case, selecting the right K-number of chunks to extract based on similarity scores, and choosing the most suitable embedding model and LLM to generate accurate and meaningful answers.

3.3 Advanced RAG techniques

The naive implementation of the RAG pattern often lacks production-level requirements for various reasons. For example, users may pose poorly defined questions that lead to irrelevant data retrieval, or the retrieved documents may not all be equally relevant to the question in hand, or even trying to "over-retrieve" information may hit the timing and pricing quota available for the custom application. To overcome these challenges, several advanced RAG techniques have been developed for each step of the pipeline. Some of those techniques will be now explained in depth.

3.3.1 Recursive Chunking

The very first step of the RAG pipeline, and one of the most crucial steps for enhancing the efficiency of LLM applications, is breaking down large data files into manageable segments. As transformer models have a fixed input sequence length, even with large input context windows, the vector of a few sentences better represents their semantic meaning than a vector averaged over larger bodies of text. For that reason, chunking the text provides the LLM with precisely the information needed with a sufficient amount of semantic representation.

There are several text splitting implementations capable of this task. The most basic chunking method simply divides text into chunks of a given number of characters, regardless of content or structure. Although this method, called fixed size chunking, is easier to be implemented, it doesn't consider the text's structure, which can lead to less coherent chunks.

The recursive chunking method offers a more structured approach. Text is divided into smaller chunks in a hierarchical and iterative manner using a set of separators until the chunks are small enough according to a predefined chunk size parameter. This technique aims to keep related text elements (first paragraphs, and then sentences, and then words) together as long as possible, as those would normally seem to be the strongest semantically related pieces of text. If the initial split doesn't produce the desired chunk size, the method recursively splits the text further using different separators until the chunks fit under the upper boundary limit. It has already been demonstrated that recursive chunking can outperform other retrieval techniques such as semantic chunking and HyDE, both at retrieval performance and computational efficiency [8]. Figure 3.2 depicts the difference between the fixed size and recursive techniques for generating chunks for the same text. The example demonstrates that, for the same number of generated chunks, the recursive approach is more effective in keeping the structure of phrases and paragraphs of the text.

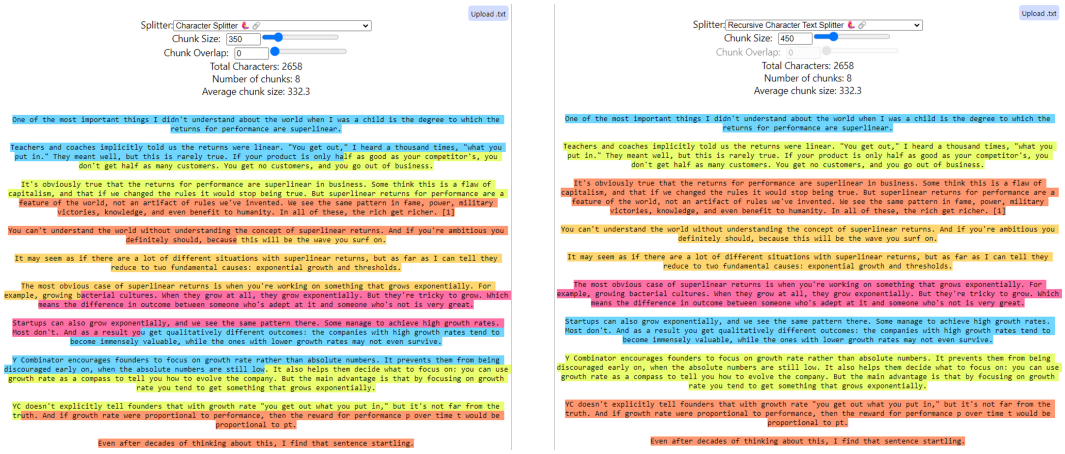


Figure 3.2: Fixed size chunking vs. Recursive chunking

LangChain, one of the most popular and consolidated frameworks for building and orchestrating LLMs, provides the `RecursiveCharacterTextSplitter` class for this purpose [14]. This class splits text using default separators (e.g. paragraph breaks, new lines, spaces, and individual characters) and allows for setting the desired number of characters per chunk. An additional `chunkOverlap` parameter can also be configured to specify how much overlap should exist between chunks, ensuring that the text is split with smoother transitions.

3.3.2 Maximal Marginal Relevance

Maximal Marginal Relevance (MMR) is a technique used in information retrieval to choose documents that are both relevant to the query and diverse compared to previously selected documents. MMR has been introduced in a paper by researchers from the Carnegie Mellon University in 1998. According to the authors: "*a new document ranking method is one where each document in the ranked list is selected according to a combined criterion of query relevance and novelty of information. The latter measures the degree of dissimilarity between the document being considered and previously selected ones already in the ranked list*" [5]. Researchers have already shown that the MMR technique can improve the learning performance for LLM retrievals due to the complementary of the explanation set [26]. The maximum marginal relevance algorithm is as follows:

$$\mathrm{MMR}(D_i) = \arg \max_{D_i \in R \setminus S} \left[\lambda \cdot \mathrm{Sim}(D_i, Q) - (1 - \lambda) \cdot \max_{D_j \in S} \mathrm{Sim}(D_i, D_j) \right] \tag{3.1}$$

Here, R is the set of all candidate documents, S is the set of already selected documents, λ is the trade-off parameter between relevancy and diversity ($0 \leqslant \lambda \leqslant 1$), $\mathrm{Sim}(D_i, Q)$ is the similarity measure between document D_i and the input query Q, $(1 - \lambda) \cdot \max_{D_j \in S} \mathrm{Sim}(D_i, D_j)$ is the maximum similarity between document D_i and any document D_j in the selected set S, and $\mathrm{MMR}(D_i)$ is the final MMR score for a document D_i.

The algorithm starts with S being empty and selects the first document D_i with the highest relevance score with respect to the query Q. This document is added to the set S.

Then, for each subsequent iteration of the algorithm, the MMR score for each remaining candidate document D_i is calculated according to the formula above. After calculating the MMR score for all candidate documents, the document with the highest score is added to the selected set S. The algorithm iterates until all documents have been selected, or until a predefined number of documents have been selected, or until a relevance threshold is met.

For the purpose of the RAG, MMR can be applied during the similarity search step to address the potential issue of ranking similar text chunks. Instead of simply retrieving the most relevant chunks, which might be very similar to each other, MMR ensures a balance between relevancy and diversity by penalizing the choice of text embeddings that are too similar to those already selected. Such approach helps in reducing redundancy of the top K selected chunks and enhancing the coverage of different aspects of the query in the chosen documents. Figure 3.3 illustrates an example of the difference between a standard simimarilty search and the implementation of the MMR to retrieve more diverse contexts.

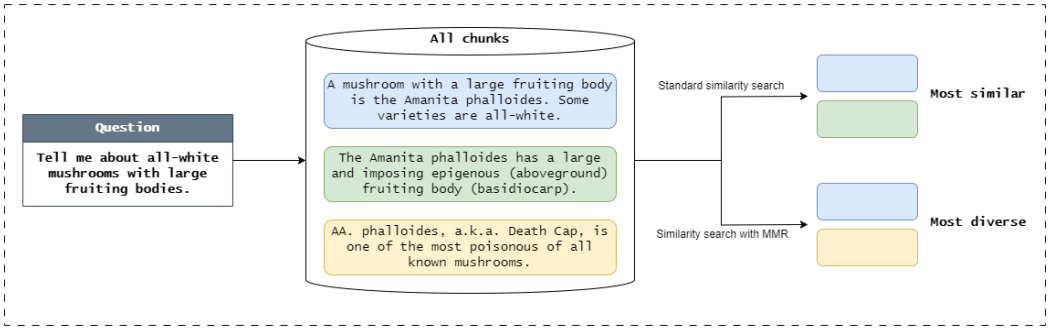


Figure 3.3: MMR example

3.3.3 Semantic Caching

One of the most cost-effective RAG patterns is called semantic caching. This method involves storing the embeddings of previously asked questions and their corresponding answers in a high-speed cache. Instead of executing the full RAG pipeline for each new query, the system first checks this semantic cache. If it finds a similar question based on embedding similarity, it retrieves the corresponding cached answer. This process avoids the costly steps of searching the vector database and generating responses with the LLM, as well as improving the response time. A standard implementation of semantic caching is shown in figure 3.4.

As a result, semantic caching serves as a powerful optimization technique for managing high volumes of repetitive user queries. It eliminates the need to repeatedly run the full RAG pipeline, saving on operational costs and reducing the time required to send requests to the LLM. This approach is particularly beneficial for production-ready applications, where multiple users assessing the application might frequently pose similar questions.

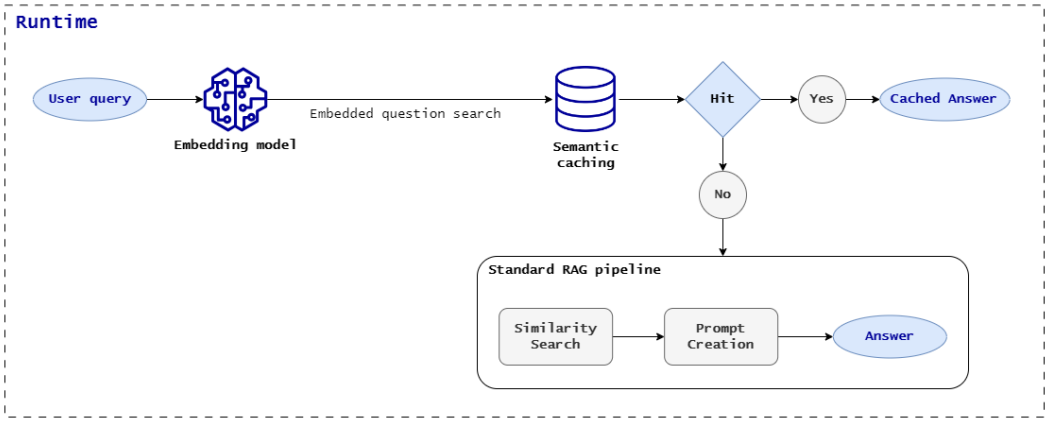


Figure 3.4: Semantic Caching pipeline

3.4 Remix framework

When it comes to full-stack development, Remix is an open-source framework that uses modern web standards to rapidly build production-ready applications and deliver high-performance user experiences. Built on top of React and React Router, and unlike traditional frameworks that fetch data on the frontend, Remix retrieves data on the backend and serves HTML directly to users, therefore reducing Javascript load and consequently improving performance.

Remix can be considered a full-stack framework because it allows developers to code both the client side and server side together. This means that web developers can shift their focus from just creating user interfaces to building entire web applications. Some of the core features of Remix are its Server-Side Rendering (SSR) capabilities, support for nested and dynamic route segments, and automatic loading state management [20].

Despite being relatively new, having been released in 2021, Remix is particularly well-suited for building robust web applications that require high performance and flexible data handling. In addition, having a single codebase for both client and server sides makes development smoother and applications more production-ready. Given these capabilities, as well as the need for a fast development pace focused on the development experience, Remix was chosen as the framework for building the web application for this mission.

3.5 Azure infrastructure

One of the most fundamental tools for making software products ready for clients is Microsoft Azure. As Microsoft's public cloud computing platform, Azure allows for building, testing, deploying, and monitoring applications and services through Microsoft-managed data centers. Azure offers a set of tools specifically suitable for building robust web applications. Some key tools available are:

1. **Azure Resources**: These are manageable items created, configured, and managed on the Azure platform. They range from virtual machines, databases, and network-

ing components to more specialized services like Azure Functions, Logic Apps, and Cognitive Search.

2. **Infrastructure Management**: Azure provides a command-line interface (CLI) through which resources can be managed and deployed automatically, avoiding the need for manual setup via the visual interface. This feature allows for seamless integration with Infrastructure as a Service (IaaS) tools like Terraform, which uses declarative coding to streamline the entire infrastructure management process.

3. **Integration with development tools**: To enforce CI/CD best practices, Azure can be seamlessly integrated with GitHub Actions for automating workflows, from code commits and pull requests to deployment and testing.

4. **Security and Compliance**: Azure ensures multi-layered protection across services and operations. Access management is handled with Microsoft Entra ID, data is encrypted both at rest and in transit, and tools such as Azure Security Center and Azure Sentinel provide real-time protection and monitoring.

3.6 Conclusion

This chapter has provided an overview of the key techniques and tools that will be employed in the subsequent development of the RAG pipeline and in the full-stack application. The implementation of Recursive Chunking improves text segmentation by maintaining semantic integrity, the Maximal Marginal Relevance enhances the combination of relevance and diversity of retrieved information during the similarity search, and Semantic Caching boosts speed and saves costs by reusing previously generated responses from similar queries. In addition, the Remix development framework is going to be used due to its high performance SSR capabilities and seamless development experience, while Azure's infrastructure provides a secure and scalable environment for deploying and managing the final application. These choices ultimately address the challenges of handling a large-scale and real-time LLM application that is aligned with the goal of delivering a production-ready solution that meets Sphinx's requirements.

Chapter 4

Methodology

4.1 Overview

This chapter provides an overview of the methodology of the mission. The chapter details the processes of defining the systems, stakeholders, needs, and constraints of the mission. Following this, the steps of development and industrialization of the final solution will be detailed with reference to the concepts introduced in Chapter 3.

4.2 Systems' Organization

Before diving into the mission itself, a reflective step was taken to consider the project's structure and resolution plan. In line with Systems Engineering principles, the main stakeholders, needs, and constraints of the mission were identified. This first analysis contributed to the development of a final solution to be delivered on time for the client. The tables 4.1, 4.2 and 4.3 provide detailed descriptions of each of these elements.

Stakeholders
Emerton Data developers
Emerton Data consultants
Client's AI & Tech unit
Client's final users
External entities (Azure, GitHub, OpenAI)

Table 4.1: Mission stakeholders

Stakeholder	**Need**
Client's final users	Have an AI system for answering questions on the ESG topic.
Client's final users	Be able to send multiple questions at once.
Emerton Data developers	Develop the system using a RAG pipeline.
Emerton Data developers	Be able to propose the following functionalities: user workflows, upload of questions and answers, access to the documentary sources. Each usecase will be defined in a script to precise the user experience that needs to be satisfied.

Table 4.2: Mission needs

Field	Requirement
Documentary source	Use already accessible data from client's public resources.
Documentary source	Be able to parse and vectorize documents of PDF type.
Documentary source	Use a suitable embedding model for vectorization.
Documentary source	Automate the document ingestion flow.
Documentary source	(Optional) Expand the document source to respond to questions on "cybersecurity".
Web App	Build homepage with question input window.
Web App	Build "questionnaire" page for processing a series of questions at once.
Web App	Implement the "Golden Answer" feature.
Web App	Enrich the generated answer by displaying the sources and chunks used, confidence level, tokens consumed, money spent on AI resources, and logs.
Infrastructure	Deploy architecture in as-code only (nothing should be configured "by hand").
Infrastructure	Manage the infrastructure with Terraform.
Infrastructure	Create 2 Azure subscriptions (staging and production) and make it available to the client.
Infrastructure	Communicate the exact roles assigned to the Emerton Data developers on the 2 subscriptions.
Infrastructure	Establish a functional CI/CD between the client's GitHub repository and Azure for all deployments and updates through GitHub Actions.
Infrastructure	Deployment with Open ID Connect (OIDC) protocol to authenticate the GitHub Actions runner.
Infrastructure	Host all services and data in French servers.
Security	Isolate network with Azure Virtual Network.
Security	Integrate app and all Azure resources with SSO and Microsoft Entra ID.
Security	Validate implementation with the client's security team.
Testing	Have a correct response rate of over 70%.
Testing	Minimize cost of AI resources while maintaining the response rate.
Testing	At first, test the RAG pipeline for 29 questions about "environment" and "human rights".
Testing	Later, respond to numerous "ESG reporting" requests (order of magnitude: 1 to 2 times/month, with 100 to 200 questions each time).
Project management	Mobilize team of developers for at least 5 weeks.
Project management	Establish an agile and continuous integration and deployment.
Project management	Maintain a functional product at every stage of development.
Project management	Create a dedicated Teams channel for communicating with the client.

Table 4.3: Mission requirements

In order to identify potential risks that could arise during the development and industrialization phases of the mission, a brief Failure Mode and Effects Analysis (FMEA) was conducted and is presented in table 4.4.

Possible failure	**(F)**	**(G)**	**(D)**	**(C)**	**Preventive actions**
False positives/negatives in AI generated answers	3	4	4	**48**	1. Implement prompt strategies for edge cases. 2. Build a validation layer that flags low-confidence answers.
Not reaching the required 70% accuracy of generated answers	3	4	1	**12**	1. Perform early testing with multiple prompt strategies. 2. Perform iterative development with user feedback.
Not delivering all features in due time	4	3	1	**12**	1. Use Agile methodology with strict sprint goals and deliverables. 2. Prioritize features with high impact.
Downtime issues with OpenAI API	2	4	1	**8**	1. Implement retry logic with timeout handling. 2. Cache results locally to reduce dependency on live API responses.

Possible failure	(F)	(G)	(D)	(C)	Preventive actions
Unexpected bugs in the application	3	3	4	**36**	1. Implement automated testing. 2. Develop new features in separate GitHub branches and push to the deployed branch through a Pull Request reviewed by superiors.
Git conflicts between developers	3	1	1	**3**	1. Code via feature branching. 2. Conduct regular syncs (trunk-based development) and ensure communication between developers.

Table 4.4: FMEA analysis (F = frequency; G = gravity; D = detectability; C = criticality = F x G x D)

Although the list of potential failures is not exhaustive, it covers most of the likely problems and their corresponding mitigation strategies.

4.3 Development

The development phase involved the actual coding and building of the software application, with an initial emphasis on the advanced techniques for RAG implementation previously presented. In this first phase, a local development environment was used. To initiate web app development as quickly as possible, and since the data being used is public, the development started on Emerton Data's own Azure account. The environment was frequently updated among the team of developers, myself included, who worked jointly on the application through a shared repository on GitHub connected with the Azure account. This environment mimicked the actual production environment, allowing for safe testing without affecting final users. The code was later migrated to the client's Azure subscriptions/Github when access and authorizations were made available. An overview of the complete project's architecture can be seen in figure 4.1. Each element of the architecture will be subsequently explained in this chapter.

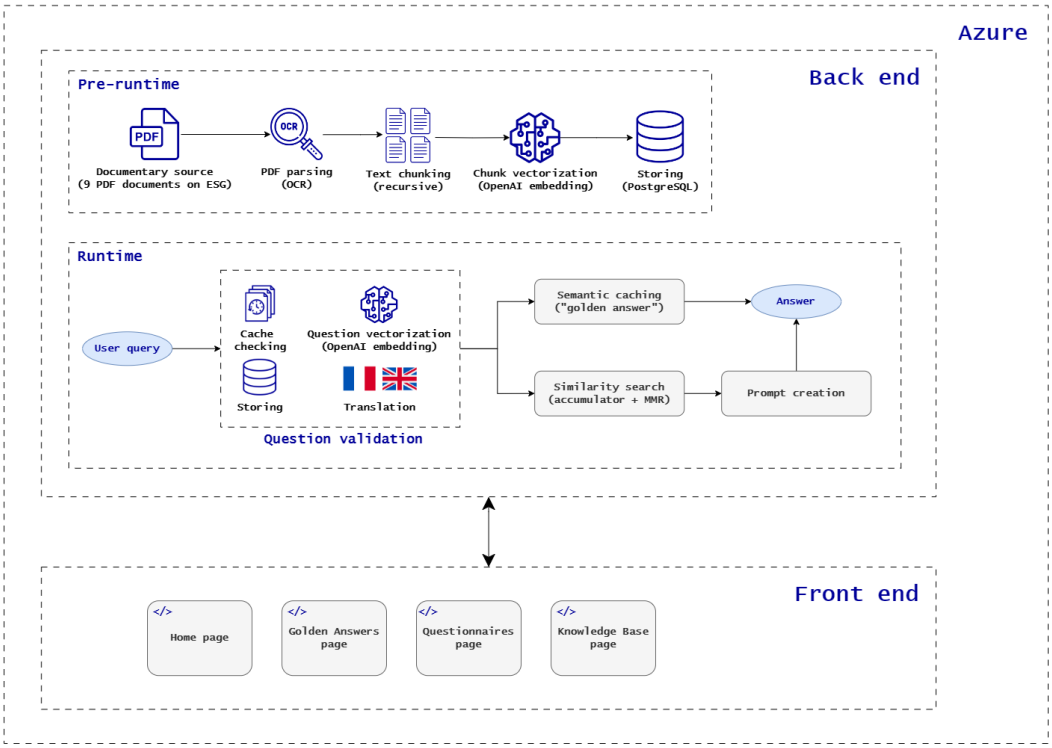


Figure 4.1: Methodology's complete architecture

4.3.1 Data Gathering

The first step after the resolution plan involved gathering all the documentary sources that would form the knowledge base content for the RAG. A total of nine company documents were collected, consisting of three documents written in English and six documents written in French, covering several ESG topics, as detailed in table 4.5. All documents were provided directly by the client in PDF format and are also publicly available.

Unlike the document chunks and their corresponding embeddings, which will be stored in a database for future searches as described later, the original PDF files were kept in the application's `public` directory. This was done solely to allow users to navigate the documents from which the RAG contexts were derived, as further explained in 4.3.10.2.

Document	Pages	Language	Elements	Subject
1	29	FR	Text, image, one single graph	Fight against corruption, risk practices.
2	30	EN	Text, image, one single graph	Fight against corruption, risk practices.
3	24	FR	Text, images, graphs	Foundations, values, code of ethics.
4	24	EN	Text, images, graphs	Foundations, values, code of ethics.
5	37	FR	Text, tables, images, graphs	KPIs, strategies, value creation, governance.
6	37	EN	Text, tables, images, graphs	KPIs, strategies, value creation, governance.
7	476	FR	Text, tables, images, graphs	Presentation, governance, performance, risk management.
8	8	FR	Text, images, graphs	Social and environmental responsibility charter for suppliers and subcontractors.
9	8	FR	Text, images, graphs	KPIs.

Table 4.5: Data explanation

4.3.2 PDF parsing

After gathering all the necessary documentation, the next step was to parse the PDFs into textual data. This was done using an Optical Character Recognition (OCR) approach that first converted each PDF page into an image of text, which was then transformed into a machine-readable text format. For this task, the `Tesseract OCR` package was utilized with the following settings:

```
"tesseract_settings": {
    "psm": 3,
    "oem": 3,
    "lang": ["eng", "fra"]
}
```

Which are defined as follows:
– `psm`: The page segmentation mode for Tesseract OCR. The default value is 3, which is the default mode for automatic page segmentation.

– `oem`: The OCR engine mode for Tesseract OCR. The default value is 3, which is the default mode for using both the LSTM and legacy OCR engines.
– `lang`: The language(s) to use for Tesseract OCR. In this case, English and French were used.

Some tables from the documents could not properly be parsed via OCR. Because of that, Azure's AI Document Intelligence was used to parse the missing tables.

4.3.3 Text Chunking

Once the textual data from the PDFs was successfully retrieved, the next step was to break the text into smaller segments, or chunks. As previously discussed, chunking enhances the efficiency and accuracy of the RAG retrieval process.

A team of Microsoft researchers demonstrated that using text chunks of 2,400 tokens resulted in the extraction of fewer entities compared to using smaller chunks of 600 tokens [9]. They also found that LLMs might not extract all semantic entities spread across documents. To address this issue, they developed a heuristic approach to perform multiple rounds of extraction (gleanings). Their results are shown in figure 4.2.

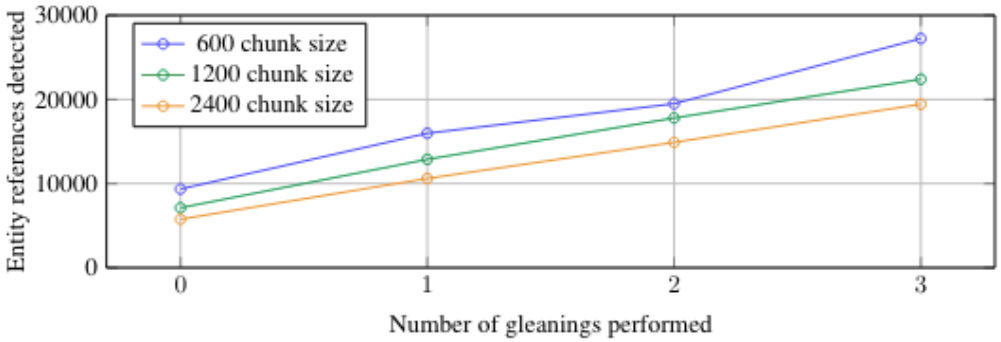


Figure 4.2: How the entity references detected in the HotPotQA dataset varies with chunk size and gleanings for generic entity extraction prompt with gpt-4-turbo [9]

For this project, it was decided to perform only one chunking extraction per document for simplicity. The text files were chunked using an adapted approach from the `RecursiveCharacterTextSplitter` class available in Langchain [14]. The following parameters were used:

```
"chunk_settings": {
    "chunk_size": 1600,
    "chunk_overlap": 200,
    "separators": ["\n\n", "\n", " ", ""]
}
```

These parameters are defined as follows:
– `chunk_size`: The maximum size of each chunk in characters. The value of 1,600 characters corresponds to approximately 400 tokens, although this number depends on the

tokenization method used by each encoder model. The choice for the number of tokens per chunk was based on the text structure of the documents used, ensuring that the chunk size is not too large to slow down retrieval times nor too small to lose context and semantic coreferences within each chunk.
– `chunk_overlap`: The number of characters to overlap between chunks. An overlap of 100 to 200 characters is typically effective in ensuring continuity between chunks.
– `separators`: An ordered list of separators for recursive splitting. The default values were used.

The table 4.6 shows the total number of chunks produced per document.

Document	Number of chunks
1	117
2	389
3	94
4	128
5	100
6	4835
7	330
8	31
9	59

Table 4.6: Number of chunks per document

The histogram in figure 4.3 illustrates the distribution of chunk sizes.

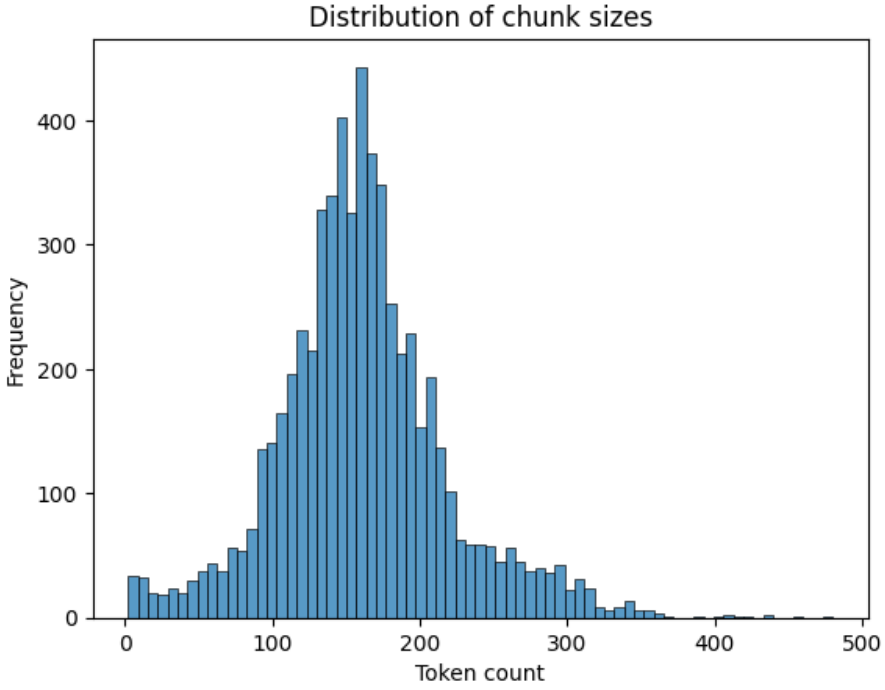


Figure 4.3: Chunk size distribution

The token count distribution follows an approximately normal pattern, peaking at around 160 tokens. The number of chunks increases steadily up to this peak and then decreases symmetrically, indicating that the majority of text chunks contain close to 160 tokens.

4.3.4 Chunk Vectorization and Storing

Following the previous step, all the text chunks were stored in a local PostgreSQL database for future use. Each chunk was stored along with an identifier, which is a Base64-encoded representation of the SHA-256 hash of the chunk's original content. There are several reasons for storing this encoded hash representation. First, hashing the textual content provides a unique fingerprint of the data with a fixed size of 32 bytes, making it easy for future lookup, indexing, and verifying whether the content has been altered by comparing the stored hash with a newly computed one. Additionally, the Base64 encoding translates the SHA-256 binary output into a string format that can be easily handled and displayed in text-based systems.

Along with the hash, each chunk was stored with its corresponding embedding, which will be used for similarity searches in future steps. The embedding model selected for this purpose was OpenAI's `text-embedding-3-large`. While OpenAI's alternative model, `text-embedding-3-small`, is optimized for latency and storage, `text-embedding-3-large` was chosen for its higher accuracy in semantic search, which is one important priority for this project.

4.3.5 Question Validation

All the previous steps involved methods executed before the runtime of the RAG pipeline. During runtime, the first task is to process the input query submitted by the user. The query undergoes a validation process depicted in figure 4.4 and described in the following workflow:

1. **Create the question on database**: The question is recorded in the database, storing its text, its SHA-256 hash encoded in Base64, and a cache status that is either "HIT" or "MISS". This status is determined based on whether the exact same question has already been stored in the database.

2. **Embed question**: If the original text of the question has not yet been embedded, the function calls the OpenAI API to generate an embedding using the `text-embedding-3-large` model. If the embedding generation is successful, it is stored along with its associated metadata (number of tokens used and the embedding cost) in the database.

3. **Translate and embed translated text**: Our application allows users to submit questions in either English or French. Therefore, an additional step is included in the workflow for processing translations. If the input question has not yet been translated, the text is translated via another OpenAI API call to the `gpt-4-1106-Preview` model. A second embedding is then generated for the translated text and stored with its corresponding metadata in the database.

4. **Update cache**: Once all new data (embeddings and translations) has been processed, the workflow updates the database with all the newly generated data.

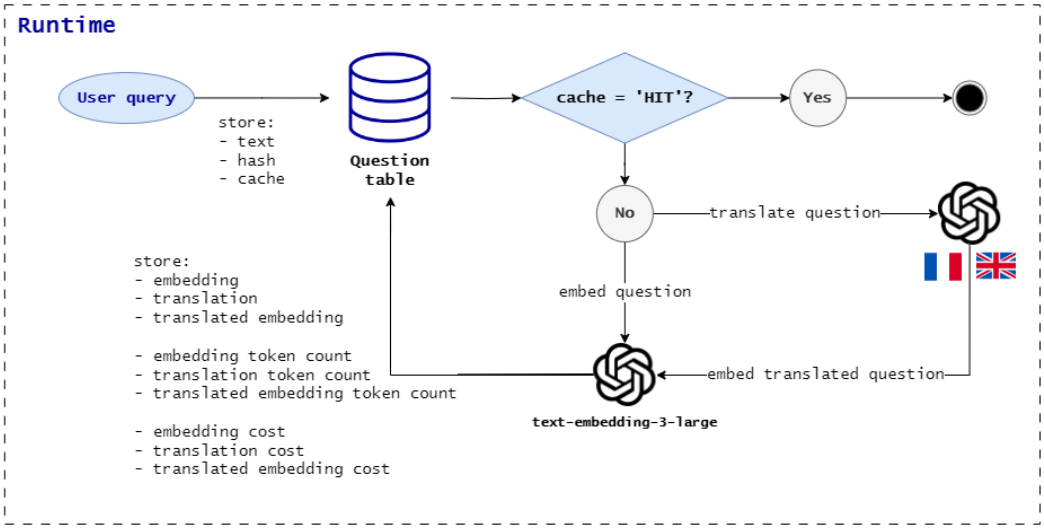


Figure 4.4: Question validation workflow

4.3.6 Golden Answer feature

After validating the question, the system attempts to find a similar question that has already been answered and marked as a "golden answer" by the final user. This is done by iterating over the available embeddings of questions whose answers have been designated as golden and comparing them with the input question and its translation (in case a similar answer exists in a different language). The comparison is performed using cosine similarity, which is calculated as follows:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{4.1}$$

Since the embeddings generated by OpenAI are normalized to a length of 1, the cosine similarity can be computed slightly faster by using just a dot product, as demonstrated in the pseudocode 1.

Algorithm 1 Cosine similarity for normalized vectors

```
function CosineNormalized(a, b)
    p ← 0
    for i = 0 to length of a − 1 do
        p ← p + a[i] × b[i]
    end for
    return p
end function
```

This function essentially measures the similarity score between two vectors by calculating their dot product, which is the sum of the products of their corresponding elements.

For this project, the most relevant embedding was selected under the condition that its similarity score was above a 0.9 threshold. This logic is directly derived from the implementation of the semantic caching defined in Chapter 3. If the semantic caching is successful, the content of the golden answer found is retrieved from the database along with its metadata (confidence and source of the answer), completing the response cycle for the posed question with a predefined valid answer.

If no question with a similarity score above the threshold was found, it indicates that the golden process did not succeed, and no suitable pre-determined answer met the criteria for being a golden answer to the question posed. In that case, the RAG pipeline continues to perform the similarity search.

4.3.7 Similarity Search

The similarity search process begins by retrieving all chunk embeddings related to the document sources from the database. It then calculates the cosine similarity between each retrieved embedding and both the embedding of the input question in its original language and its translated version. This dual comparison is necessary because the source documents are written in both English and French. Thus, an input question in English might be best answered by referring to the content from a French document, or vice versa.

To manage memory efficiently when processing a large number of embeddings, an accumulator is used to handle each chunk embedding in batches of a fixed size N. Two accumulator objects are initialized: one stores the embeddings and their respective similarities to the original question, and the other stores the embeddings and their similarities to the translated question. For each embedding processed (whether original or translated), it is added to the appropriate accumulator. When the accumulator reaches its maximum size N, it triggers a merge operation. This operation evaluates which embeddings from the current batch should be retained in the final result set based on their similarity scores. The merge operation combines the final results with the temporary results of each batch, sorts them by similarity, and truncates the list to the top N entries. This ensures that only the most relevant embeddings are kept in the final result set. The pseudocodes 2 and 3 for the accumulator logic are shown below.

In this implementation, the "cache" represents the temporary list that contains the embeddings of a batch at a given time, along with their previously calculated similarities. It is important to note that the accumulator already handles the sorting of embeddings based on their similarity scores, which is required for the subsequent MMR technique.

Algorithm 2 UpdateAccumulator

```
function UpdateAccumulator(N, accumulator)
    Input: item
    if item.similarity < accumulator.minimumSimilarity then
        return
    end if
    if accumulator.cache.push(item) ⩾ N then
        accumulator.result ← MergeCache(N, accumulator)
        accumulator.cache ← []
        last ← last element of accumulator.result
        if last exists then
            accumulator.minimumSimilarity ← last.similarity
        end if
    end if
end function
```

Algorithm 3 MergeCache

```
function MergeCache(N, accumulator)
    merged ← accumulator.result ∪ accumulator.cache
    Sort merged by similarity in descending order
    return First N elements of merged
end function
```

As described in Chapter 3, the MMR technique is used to rerank the most important embeddings returned by the accumulator based on both relevance and diversity. The pseudocode 4 for the MMR implementation is provided below.

Algorithm 4 MMR (Maximal Marginal Relevance)

```
function MMR(sortedEmbeddings, maxLength, lambdaParam)
    if maxLength ⩾ length of sortedEmbeddings then
        return sortedEmbeddings
    end if
    first ← first element of sortedEmbeddings
    candidates ← rest of sortedEmbeddings
    if not first then
        return []
    end if
    results ← [first]
    while length of results ⩽ maxLength do
        bestCandidate ← None
        bestMMR ← −∞
        for each candidate in candidates with index i do
            mmrValue ← lambdaParam × candidate.similarity+
              (lambdaParam−1)×max(CosineNormalized(candidate.embedding, result.embedding))
            if mmrValue > bestMMR then
                bestMMR ← mmrValue
                bestCandidate ← candidate
                bestIndex ← i
            end if
        end for
        Add bestCandidate to results
        candidates ← candidates[0 to bestIndex − 1] ∪ arr[bestIndex + 1 to end]
    end while
    return results
end function
```

In the implementation of MMR, the lambda value was set to 0.2, prioritizing diversity over relevance. The maximum length of selected embeddings from MMR ($maxLength$) was set to 8. The number N of embeddings selected from the database in the previous accumulator logic was set to be ten times higher than $maxLength$.

4.3.8 Prompt Creation

After selecting the best embeddings, their corresponding textual content is used to create a prompt to be fed into the OpenAI model. This prompt is formed by concatenating the original question with the array of retrieved contexts into a structured string.

4.3.9 Answer Generation

The final step in the RAG pipeline is to generate the answer based on the created prompt. The answer is generated through an API call to OpenAI's `gpt-4o` model using two message roles: the system message, which sets the behavior of the AI assistant with high-level instructions, and the user message, which provides the query to which the assistant can respond. In our case, the user message is the prompt text generated in the previous step, while the system message is a custom-formatted prompt that will be explained in Chapter 5.

The response from the API call includes the answer to the question along with its metadata (number of prompt tokens, number of completion tokens, model used, and confidence level of the generated answer). If the answer is successfully generated, it is stored in the database with its metadata. If not, the program handles the failure scenario, ensuring that the pipeline terminates gracefully.

4.3.10 Web App Development

4.3.10.1 Back end Development

The entire application, which includes the RAG pipeline and user interface, was developed in TypeScript using the Remix framework. The application communicated with a PostgreSQL relational database through an ORM (Object Relational Mapper). An ORM is a tool designed to translate data representations used by databases into those used in object-oriented programming. The advantage of using an ORM is that it abstracts database interactions, allowing the database schema and operations to be managed centrally in the code. This approach enabled us to handle database objects in the project's programming language without relying on direct SQL queries. For this project, the Prisma ORM was used, and the corresponding data schema is shown in figure 4.5.

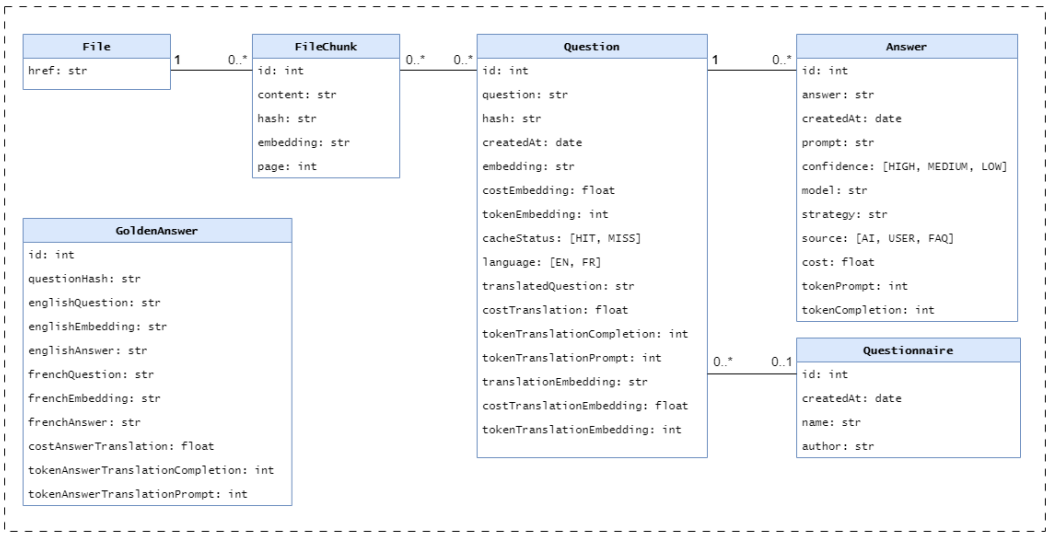


Figure 4.5: Data schema of the application

In the schema, the "File" model represents the files used as the knowledge source for the RAG, and their associated chunks are represented in the "FileChunk" model, which contains both textual and vectorized representations of the chunks, as well as the page numbers where they are located in the documents. The "Question" model stores all the information about a posed question, including its creation date, associated embedding, translation, and metadata from OpenAI calls. The "Questionnaire" model represents a collection of questions, storing attributes like its name, creation date, and author. The "Answer" model contains all information about a generated answer, including its creation date, the complete prompt used to generate the answer, the confidence level assigned by OpenAI, the model and strategies used for the prompt, and whether the answer was generated by OpenAI, edited by the user, or retrieved from a golden answer, along with metadata from API calls. Finally, the "GoldenAnswer" model stores all details about an answer selected as a golden answer. This includes the English and French versions of the question and answer, their embeddings, metadata from API calls, and a hash of the question used to link a question from the "Question" model to its associated golden answer, if one exists.

The web application was built using a modularized pattern to enforce separation of concerns as much as possible. The codebase is structured as follows:

1. **AI module**: Manages all prompt strategies and OpenAI calls, including embedding text, translating questions and answers, and querying.

2. **Similarity module**: Implements the algorithms for cosine similarity (1), the accumulator (2), and MMR (4).

3. **Database module**: Stores all processed text chunks along with their corresponding embeddings. This module also contains the Prisma schema for configuring the ORM, including data migrations and seed configurations.

4. **Pub/Sub module**: Implements the publish-subscribe messaging pattern to enable asynchronous operation for multiple users.

5. **Task Manager module**: Coordinates the parallel execution of the RAG pipeline when multiple questions are submitted through a questionnaire. The steps of translation, similarity search and answer generation are put in three separate execution queues, allowing the RAG to be done in parallel for multiple questions that are currently in different stages of the process.

6. **Web module**: Contains the core application logic, divided into five layers:

   (a) **Routes layer**: Defines all navigable pages in the application, including the home page, golden answers page, questionnaires page, and knowledge base page.

   (b) **Services layer**: Connects the web module's inner layers with other modules. This layer orchestrates calls to all steps of the RAG pipeline and manages CRUD operations in the database.

   (c) **Components layer**: Provides reusable components such as forms, bars, and inputs for use across other layers.

(d) **Styles layer**: Defines color themes and text fonts in alignment with the client's design standards.

(e) **Utils layer**: Contains additional utilities used across other layers, such as string encoding and decoding, hash calculations, OpenAI cost calculations, and log monitoring.

4.3.10.2 Front end Development

The visual interface is the part of the application where clients had the ability to manage the RAG pipeline running under the hood and view its results in a user-friendly manner. The interface was built in TypeScript, using the MUI (Material UI) library. MUI is a comprehensive React library that implements Material Design, a design language developed by Google. MUI provides pre-built and customizable UI components that enable the creation of visually appealing interfaces more quickly. The entire design of the application was developed collaboratively with guidance from Sphinx's Tech unit to ensure consistency with their visual identity. The interface consists of four functional pages, described below:

1. **Home page** (figure 4.6): This page features an input field where users can type and submit a question, and choose between two different prompt strategies for generating the answer (verbose mode or lightweight mode). Once the question is processed through the RAG pipeline, a new component appears at the top, displaying the question posed, the generated answer, the chunks used to generate the answer (with hyperlinks to their corresponding documents), the source of the answer (either AI-generated, edited by the user, or retrieved from a "golden answer"), the confidence level assigned by OpenAI (high, medium, or low), and the date the answer was generated. Each component also includes options to edit the AI-generated answer, save the answer as a "golden answer", or delete the component. All previously asked questions are displayed below the newly generated answer, ordered from newest to oldest.

2. **Golden Answers page** (figure 4.7): This page stores all answers marked as "golden answers" by the user. Each golden answer can be expanded to view both the question and answer in their English and French versions. Users also have the option to delete a golden answer.

3. **Questionnaires page** (figure 4.8): This page allows users to create a questionnaire with a custom name and submit multiple questions at once. After submission, the RAG pipeline runs in parallel for all the questions in the questionnaire. The progress of the RAG execution, which includes the steps of creating the question, translating it, doing the similarity search, and generating an answer, can be viewed by the user. Users also have the ability to view and edit previously submitted questionnaires.

4. **Knowledge Base page** (figure 4.9): This page allows users to view and download all the documentary sources being used to generate the chunks that are fed into the RAG pipeline.

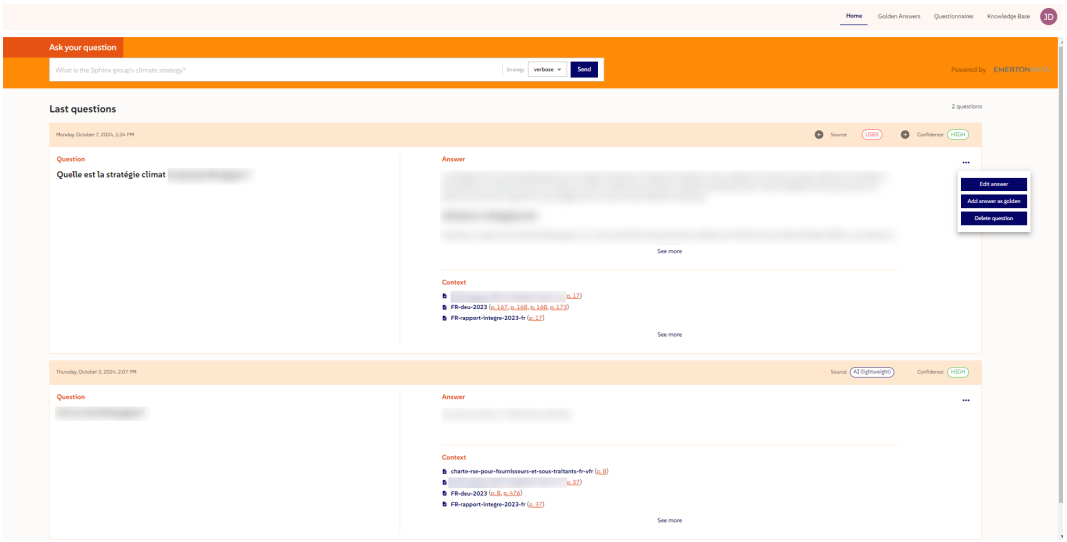
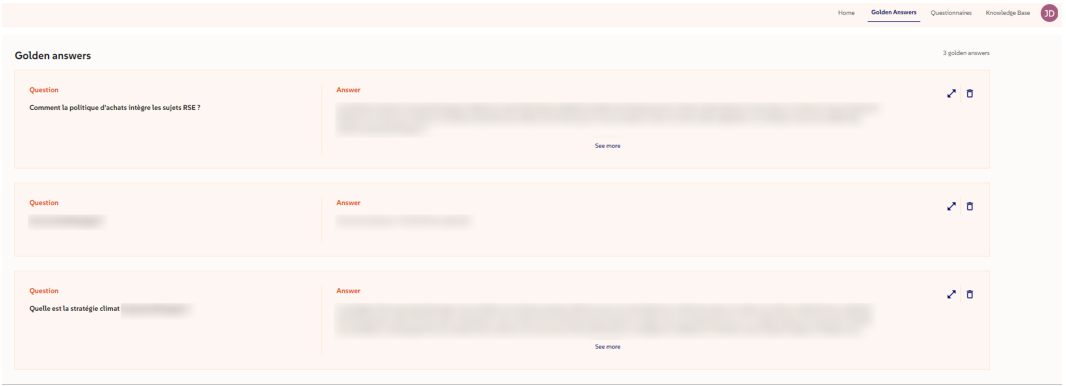


Figure 4.6: Home page



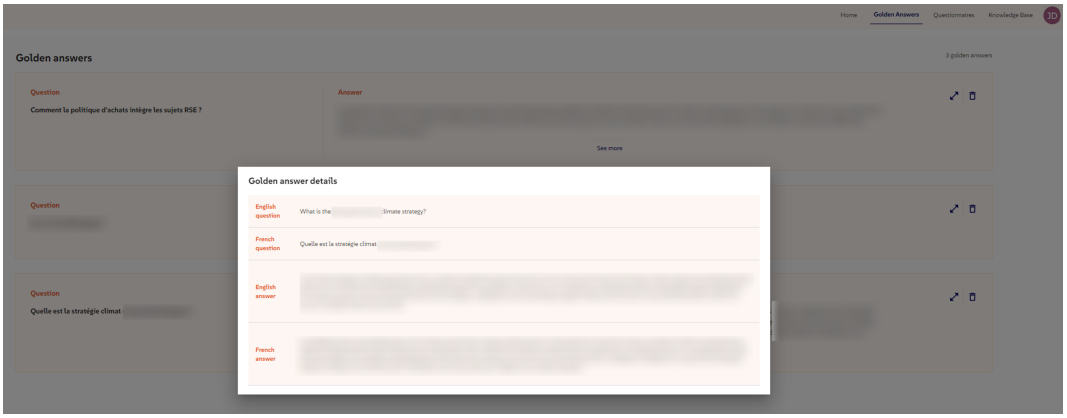


Figure 4.7: Golden Answers page

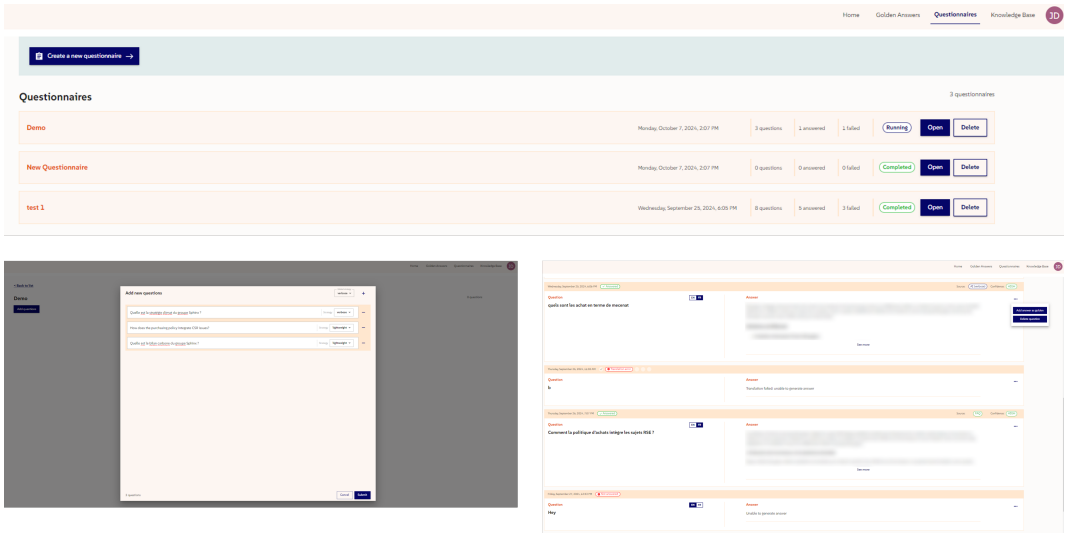
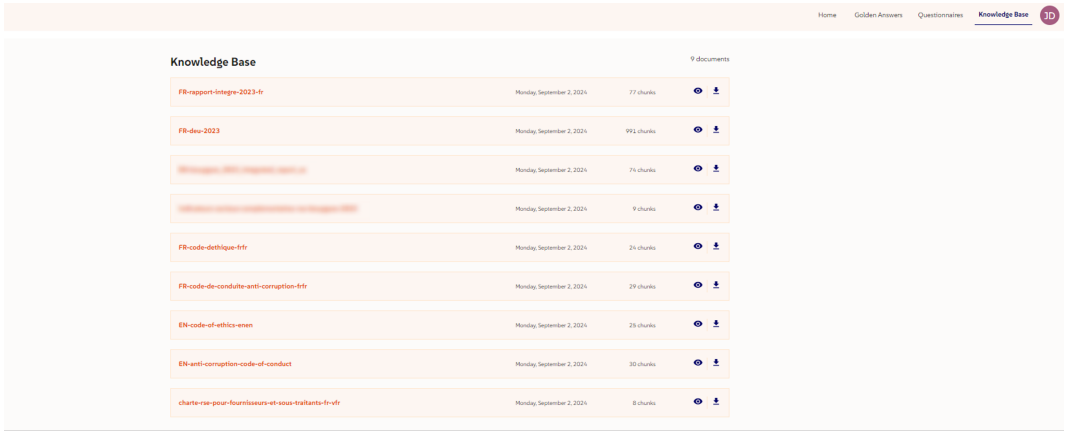


Figure 4.8: Questionnaires page



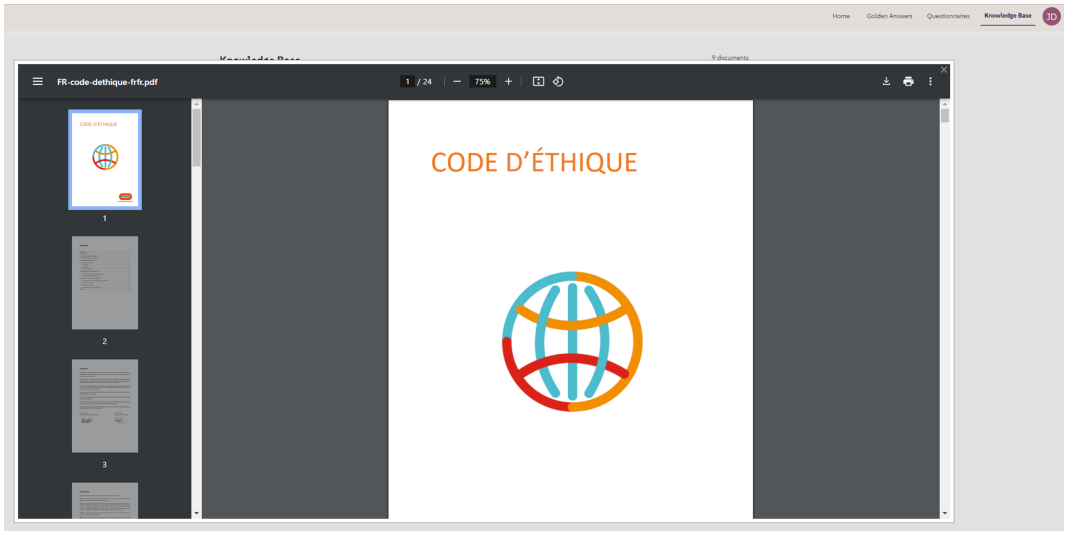


Figure 4.9: Knowledge Base page

4.4 Industrialization

The industrialization phase involved building a CI/CD pipeline for automated deployment into a production environment and establishing the infrastructure on the Azure platform. This environment consists of a controlled setting where the application is live, accessible to end-users, and optimized for security and performance. The industrialization phase was mainly coordinated and implemented by Jérôme Feroldi - my tutor and Emerton Data's principal Data & Devops Engineer.

4.4.1 GitHub CI/CD

A GitHub Actions workflow was set up in the application repository to automatically trigger a deployment to Azure whenever new updates are pushed to the main branch of the project. When triggered, the workflow initiates a build phase, during which all project dependencies are installed, running in parallel with an Azure CI. The Azure CI first logs into the Azure account to access cloud resources, installs the Terraform CLI, and runs a Terraform configuration plan to provision resources on Azure. If both the build setup and Azure CI are successful, the workflow proceeds to the next stage - a migration CI. This CI is responsible for applying Prisma migrations to the database and seeding it with initial data. The final step of the workflow is the deployment CI, which logs into the Azure account again, creates a zip file of the application, and deploys the zipped artifact to an Azure Web App. The entire GitHub workflow, illustrated in figure 4.10, takes approximately five to six minutes to be completed.

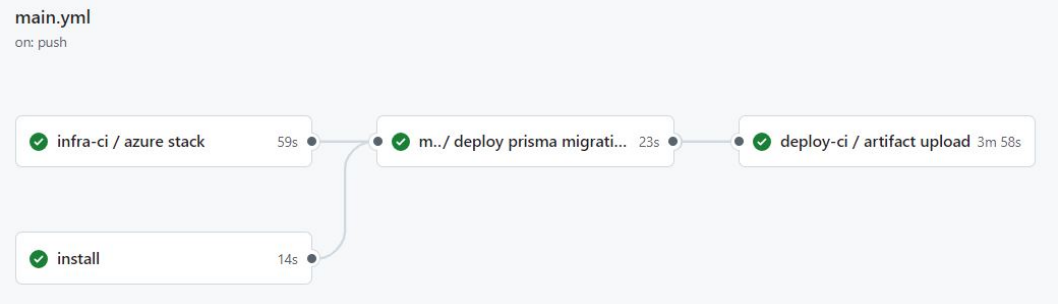


Figure 4.10: CI/CD pipeline on GitHub

4.4.2 Azure Stack

The application is fully deployed on our client's Azure account, utilizing the following Azure resources:

1. **App Service**: A fully managed platform-as-a-service (PaaS) used to host and run the main application.

2. **App Service Plan**: Defines the set of compute resources for the web application. The App Service Plan must be created for the App Service to run using the defined compute resources.

3. **Key Vault**: Used to securely store and access the application's secrets (i.e. API keys, passwords, and certificates).

4. **PostgreSQL Server**: A fully managed relational database-as-a-service based on the open-source PostgreSQL engine, used to store and manage all application data.

5. **Azure OpenAI Service**: Provides access to OpenAI's language models and embeddings through a REST API.

All resources and data are hosted on servers located in France to ensure better latency, performance, and regulatory compliance. In addition to the Azure resources, Microsoft Authenticator and Microsoft Entra ID were used to secure access to the web application through multi-factor authentication and single sign-on (SSO). An overview of the complete cloud architecture is shown in figure 4.11.

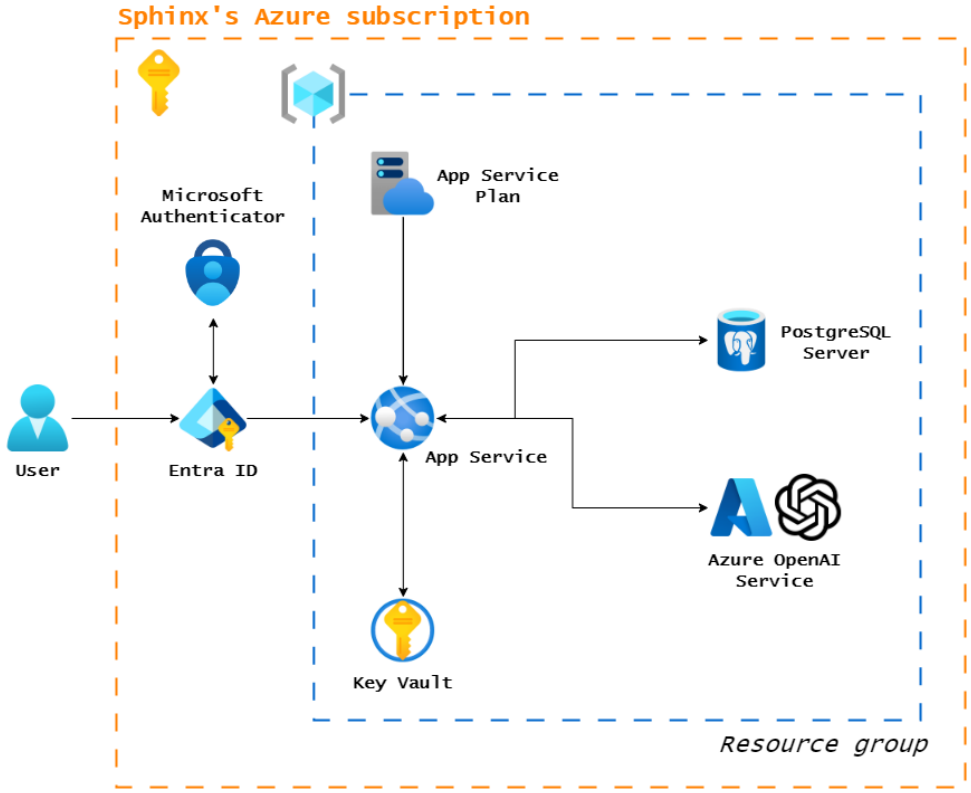


Figure 4.11: Azure architecture

4.5 Conclusion

In summary, this chapter covered the entire development process of our product — from system definition to cloud deployment. Building on the approaches reviewed in Chapter 3, our methodology incorporated recursive chunking, semantic caching, and MMR to optimize answer generation within the RAG pipeline. Additionally, we used the Remix framework, MUI library, and Azure stack to build and scale the application powered by the RAG.

The methodologies discussed provide two layers of analysis. The first layer allows us to evaluate the quality of the answers produced by the RAG pipeline in terms of relevance to the posed questions, the matched chunks, and the available documents. The second layer assesses the efficiency of the application's architecture in terms of response time and usage costs.

Chapter 5

Analysis and Results

5.1 Overview

This session presents an analysis of the methodology described in the previous chapter. It begins with an overview of the testing of the RAG pipeline, including the metrics used to assess the quality of the generated answers. Next, the performance of the web application is discussed regarding usage costs and execution time. Finally, the chapter concludes by outlining what was achieved in relation to the initial mission requirements, as well as the limitations of the final product.

5.2 RAG Evaluation

5.2.1 Testing

The RAG pipeline described in Chapter 4 was tested with 29 validation questions. These questions were provided by Sphinx's AI & Tech Unit and each one of them contained ground truth answers, so that we could evaluate the quality of the answers generated on our side.

For each question submitted as input, it was concatenated with the most relevant text chunks retrieved from the RAG pipeline. This combined text served as the "user" message sent to the OpenAI API. To complete the prompt, a "system" message was created to provide instructions for the LLM to generate its response. In this context, two system messages were crafted, representing two different prompt strategies: verbose and lightweight. The verbose strategy aims to generate a comprehensive answer with a detailed explanation of the topic, while the lightweight strategy is used to produce concise and straightforward answers. These strategies are detailed in Appendix A.

The strategy choice can be selected by the user through the visual interface when submitting a question. It is important to note that the questions submitted and the chunks retrieved remain the same, regardless of the chosen strategy.

5.2.2 Answer Quality Analysis

To assess the quality of the answers generated by the RAG pipeline, the RAGAS metric was utilized. RAGAS, which stands for Retrieval Augmented Generation Assessment, is a widely used framework for reference-free evaluation of RAG pipelines [10]. The key advantage of using RAGAS is its ability to evaluate not only the quality of the generated answers in relation to the posed question and the ground truth answers, but also to the relevance of the retrieved contexts and the model's ability to use those contexts faith-

fully, all without relying on human annotations but leveraging LLMs under the hood for conducting the evaluations instead.

To evaluate the RAG pipeline, RAGAS requires the following inputs: the posed question, the generated answer, the retrieved contexts from the knowledge source, and the ground truth answer (which is the only human-annotated information). RAGAS provides the following evaluation metrics:

1. **Context Precision** (retrieval metric): Evaluates whether all ground-truth relevant items in the contexts are ranked as the highest ones. This metric is computed using the question, the ground truth answer, and the contexts. Its formula is given below:

$$\text{Context Precision@K} = \frac{\sum_{k=1}^{K} (\text{Precision@k} \times v_k)}{\text{Total number of relevant items in the top } K \text{ results}}$$

$$\text{Precision@k} = \frac{\text{true positives@k}}{\text{true positives@k} + \text{false positives@k}} \tag{5.1}$$

   Where K is the total number of chunks in the contexts and $v_k \in \{0, 1\}$ is the relevance indicator at rank k.

2. **Context Recall** (retrieval metric): Measures how well the retrieved contexts align with the ground truth answer. This metric is computed using the ground truth answer and the contexts. Its formula is given below:

$$\text{Context Recall} = \frac{|\text{Ground truth claims that can be attributed to context}|}{|\text{Number of claims in ground truth}|} \tag{5.2}$$

3. **Faithfulness** (generation metric): Measures the factual accuracy of the generated answer against the given context. This metric is computed using the answer and the contexts. Its formula is given below:

$$\text{Faithfulness} = \frac{\begin{array}{c}\text{Number of claims in the generated answer} \\ \text{that can be inferred from given context}\end{array}}{|\text{Total number of claims in the generated answer}|} \tag{5.3}$$

4. **Answer Relevancy** (generation metric): Assesses how pertinent the generated answer is to the given prompt. This metric is computed using the question and the answer. Answer Relevancy is defined as the mean cosine similarity of the original question to a set of artificial questions created based on the generated answer. Its formula is given below:

$$\text{Answer Relevancy} = \frac{1}{N} \sum_{i=1}^{N} \frac{\mathbf{E_{g_i}} \cdot \mathbf{E_o}}{\|\mathbf{E_{g_i}}\| \|\mathbf{E_o}\|} \tag{5.4}$$

   Where E_{g_i} is the embedding of the generated question i, E_o is the embedding of the original question, and N is the number of artificially generated questions (set as 3 by default).

5. **Answer Correctness** (end-to-end metric): Assesses the accuracy of the generated answer compared to the ground truth. This metric is computed using the ground truth answer and the generated answer. Answer Correctness combines the semantic similarity between the answers (calculated via cosine similarity) and their factual similarity (calculated via the F1 score) using a weighted average to compute the correctness score.

All RAGAS metrics are scaled from 0 to 1, with higher values indicating better performance. These five metrics were tested against the validation questions using the verbose prompt strategy and the `gpt-4o` model from OpenAI. The results are shown in figure 5.1.

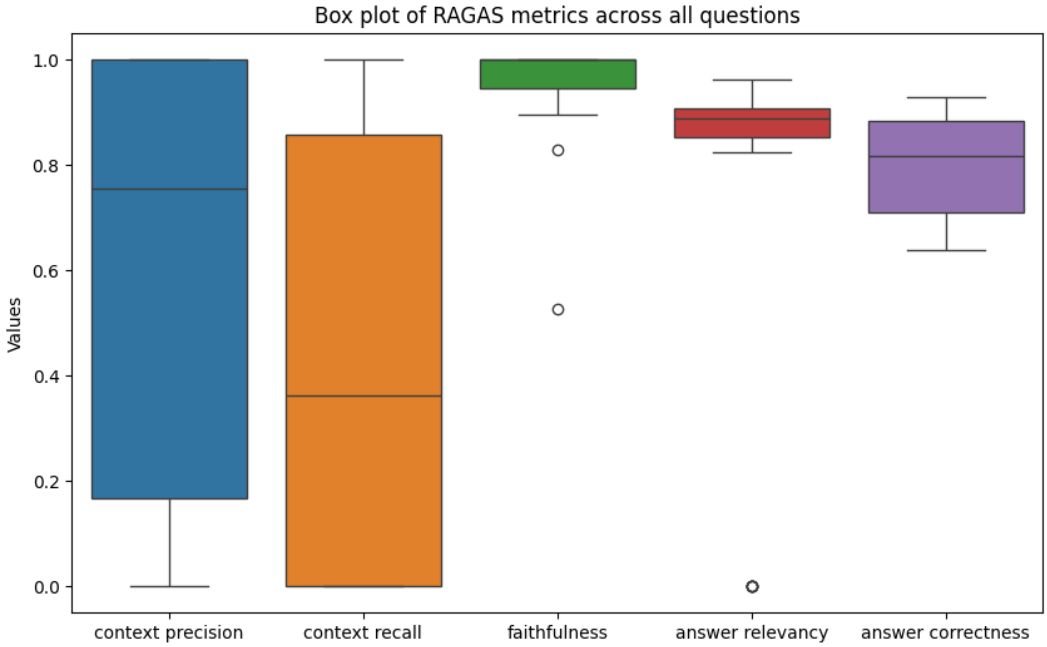


Figure 5.1: RAGAS results for the validation questions

By interpreting the results, we observe that the overall values for faithfulness, answer relevancy, and answer correctness are quite high, with respective medians of 1.0, 0.89, and 0.82, all with an interquartile range above 0.71. The high faithfulness values indicate that the accuracy of responses is closely linked to the retrieved contexts. The answer relevancy results show that the generated answers are fairly relevant to the posed questions, and the answer correctness values indicate that most of the generated answers are indeed accurate compared to the ground truth answers.

As for the two retrieval metrics, context precision had a median of 0.75, with an interquartile range from 0.17 to 1.0, while context recall had a median of 0.36, with an interquartile range from 0 to 0.86. These results suggest that the retrieval process is not fully effective in capturing all relevant information required to answer the question, and when relevant information is actually retrieved, only part of it is identified as a relevant context. The lower performance in these two metrics can be partly explained by some missing ground truth answers for a number of questions, which could lower the context precision and recall scores since those metrics rely on ground truth inputs.

Nevertheless, these results suggest there is room for improvement and experimentation in the way contexts are retrieved and deemed relevant to answering the questions.

5.3 App Performance Evaluation

5.3.1 Azure Analysis

By utilizing native services from Microsoft Azure, the team was able to monitor all costs and usage associated with the services used to deploy the web application.

The monitoring system can track token usage within the RAG pipeline, broken down by each OpenAI model employed. Figure 5.2 shows an example of a daily report on token usage for the models `gpt-4o` (used for generating answers), `gpt-4-1106-Preview` (used for translations), and `text-embedding-3-large` (used for embeddings). The dashboard shows the number of processed inference tokens per day, which are calculated as input prompt tokens plus output generated tokens.

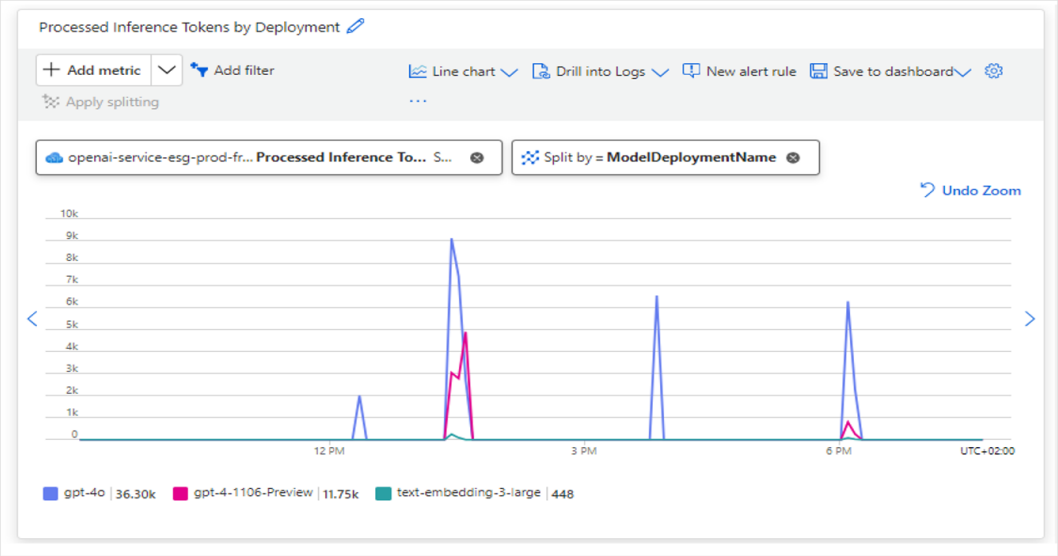


Figure 5.2: Azure OpenAI metrics dashboard for daily token usage per model

The hosted application incurs in a fixed cost of 4 euros per day and a variable cost of 0.02 euros per question posed. The fixed costs are related to keeping the web server operating continuously, while the variable costs cover embedding, translating, and generating answers for each question. Figure 5.3 illustrates the accumulated costs of Azure services over two weeks, with daily expenses ranging from 5 to 8 euros. Assuming that Sphinx plans to run the application 1 to 2 times per month, with 100 to 200 questions posed each time, and maintaining the web server fully active throughout the month, the total monthly usage cost is estimated to be a maximum of 130 euros. These expenses are relatively low, considering the high value the product provides by enabling Sphinx employees to gather responses from the company's ESG data sources in a fast and efficient manner.

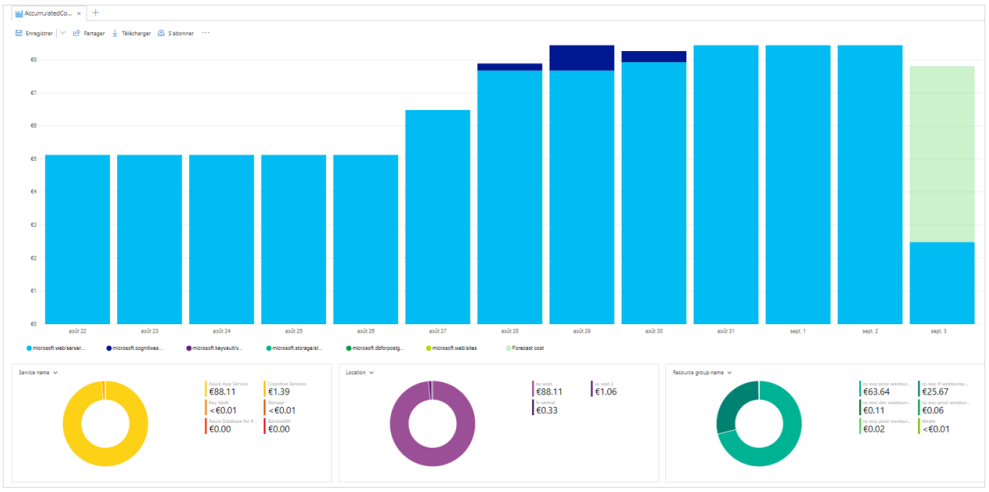


Figure 5.3: Azure cost analysis for a 2-week usage period

5.3.2 Latency Analysis

One of the most important metrics to consider when working with Large Language Models is latency. In the context of LLMs, latency refers to the time it takes to send a request to the model and receive a response back. The two key latency metrics for evaluating LLM performance are Time to First Token (TTFT) and Tokens Per Second (TPS). TTFT measures how long it takes for a model to generate the first output token after receiving a request. The lower the TTFT, the more responsive the application. TPS, on the other hand, measures how quickly the model generates tokens once it starts crafting the answer. A higher TPS means faster token generation, resulting in a quicker return of the full answer to the user. The total Answer Generation Time is calculated by combining TTFT and TPS, as shown in the formula below:

$$\text{Answer Generation Time} = \text{TTFT} + \frac{\text{Number of Output Tokens}}{\text{TPS}} \tag{5.5}$$

By knowing the values of these latency metrics for the specific model used, it is possible to calculate the total time required to generate an answer. The `gpt-4o` model used to generate the answers of our RAG, when launched via the Paris region through the Azure OpenAI service, has a TTFT of 275ms and a TPS of 101.28 tokens [23]. Figure 5.4 shows a chart with the number of completion tokens and the answer generation time for each of the 29 questions used to test the application. In this testing scenario, all questions were asked using the verbose prompt strategy.

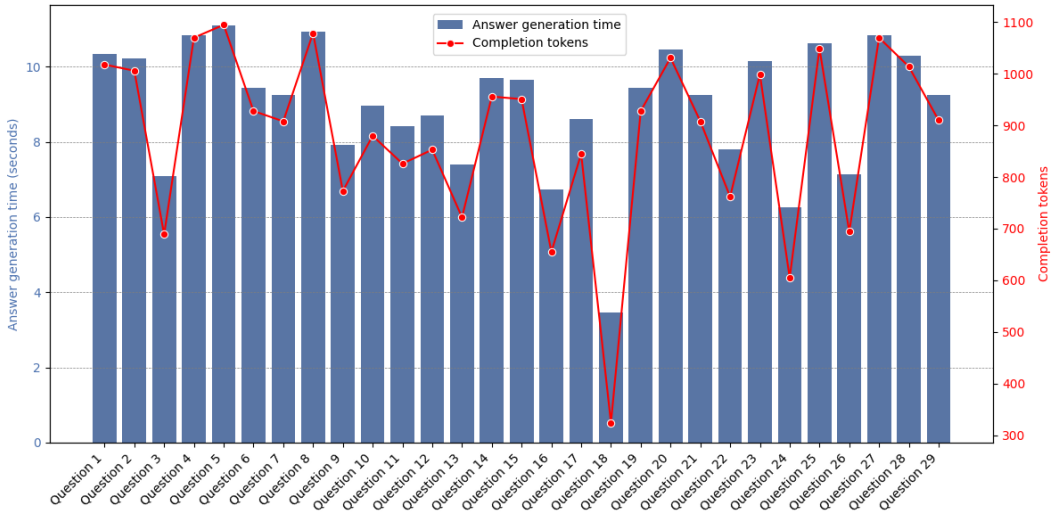


Figure 5.4: Answer generation time and number of completion tokens per question

It can be observed that most questions take between 7 and 10 seconds to generate an answer, with the duration directly correlating with the number of completion tokens, as expected.

Figure 5.5 shows a chart that compares the answer generation time with the total execution time, which measures the time from sending a question via the application to seeing the answer be displayed. The total execution time includes all steps of the RAG pipeline (translation, embedding, similarity search, and answer generation), as well as any additional time taken by the web server.

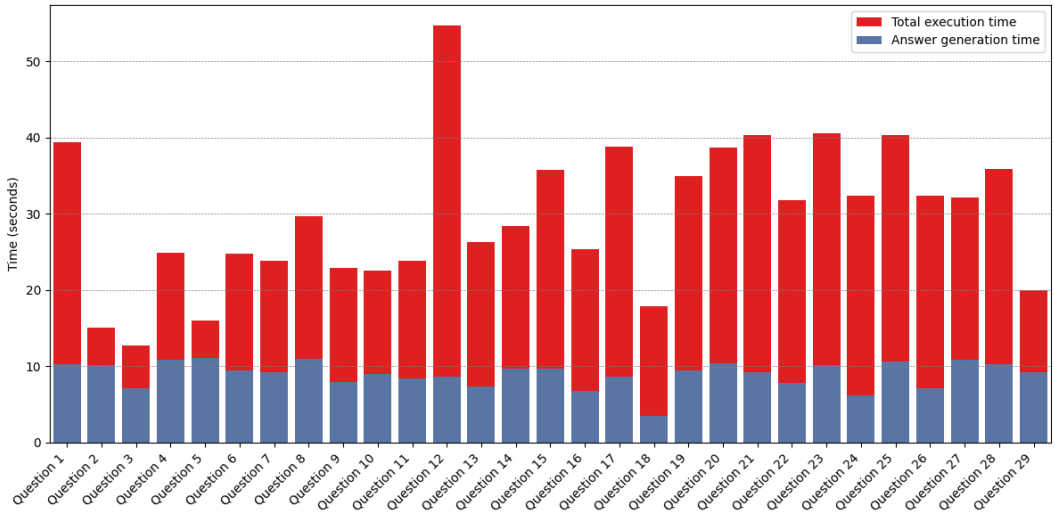


Figure 5.5: Total execution time and answer generation time per question

Unlike the previous chart, there doesn't seem to exist a direct proportionality between the time required to generate an answer and the total time to execute the RAG pipeline and display the answer to the user. This discrepancy can be explained by the fact that

other processes contributing to the total execution time are not necessarily linked to the number of completion tokens. For instance, a long question with many input tokens (which takes more time to process) might generate a short answer with fewer completion tokens. Another case is the time required to translate the question, which may be longer or shorter depending on the question's complexity, and it does not correlate with the number of completion tokens.

5.4 Achievements

In summary, a complete system capable of automating responses to ESG-related questions in two languages is now ready to be used by the client. Most of the stakeholders' needs, which were translated into requirements, have been successfully fulfilled, though a few less critical requirements remain to be implemented.

Regarding the knowledge source, all documents provided by Sphinx have been properly processed and are ready to be ingested into the pipeline. However, this process has not yet been fully automated, as it is still needed to manually upload each new document and parse it before ingestion.

For the web application, all major functionalities requested by Sphinx have been implemented, including the home page for submitting questions and reviewing answers at a granular level, the questionnaire page for submitting multiple questions simultaneously, and the Golden Answer page for processing cached answers. The implementation of semantic caching and the orchestration of questionnaires through a task manager effectively addressed the challenge of running the RAG pipeline in real-time as users navigate the application. Additionally, the design and user experience aspects of the app have been validated by the end users.

In terms of infrastructure, the implementation of a functional CI/CD pipeline connecting GitHub with Azure has streamlined the user experience by enabling us to implement all RAG functionalities and web app features in a decentralized and agile manner. This setup allowed for direct deployment to production once each feature was validated by our supervisors. Furthermore, by utilizing Azure's security resources and its native integration with OpenAI, we ensured that all necessary security measures were met.

Finally, the testing phase revealed that the validation questions achieved a correct response rate of over 70% across four out of five quality metrics, while the performance analysis demonstrated that the associated usage costs were minimal and the execution times were reasonably low, given the value provided to the client.

5.5 Limitations

Although all the milestones for this mission have been achieved, there are limitations that need to be addressed:

1. **Scaling issues**: The execution time results were tested through unit tests for each individual question to prevent any potential overload caused by the accumulation of previously answered questions in the application. However, it has been observed

that the application starts to slow down as the number of answered questions increases. Even though preventive measures have been taken to mitigate this issue, such as implementing a task manager pattern to parallelize the RAG pipeline for each question within a questionnaire, additional stress tests should be conducted in the future to ensure the system can handle larger volumes of data.

2. **Quantity of data**: The RAG pipeline is currently fed by 9 document sources, and its performance has been tested with 29 questions. Expanding the number of available documents would help assess whether the MMR strategy is truly effective in capturing diverse chunks of data, while having more ground truth answers would lets us draw more precise conclusions from the RAGAS metrics. This way, a more comprehensive understanding of the application's overall performance could be gained.

3. **Chunking issues**: The current method for chunking the texts can only take us so far when capturing the deeper meaning and context behind the data, as shown by the context precision and recall scores. Since recursive chunking focuses on breaking large text bodies into smaller parts for efficient retrieval, this process may lose the semantic connections between different pieces of information scattered across the documents. More advanced retrieval techniques, discussed in Chapter 6, may overcome this limitation by enabling the LLM to understand relationships between entities and potentially improving generation results.

4. **Bias in evaluation**: Although the RAGAS metrics provide a cheaper and faster evaluation method by using LLMs for reference-free evaluation, there are still some concerns about its shortcomings regarding biases. Researchers have shown that using LLMs as referees to score the quality of generated responses may question the resilience of this paradigm against perturbations, such as altering the ordering of candidate responses during scoring [25]. Therefore, it is recommended to incorporate human-in-the-loop interventions to calibrate the quality evaluation of our RAG.

Chapter 6

Conclusion

6.1 Overview

In conclusion, this end-of-study internship and Master's thesis primarily focused on developing a state-of-the-art GenAI application to meet a key client need: the ability to respond to various ESG-related forms about their company. Since this product was entirely custom-made for Sphinx, it was essential to meet several business requirements throughout production, including data privacy, visual identity, and the relevance of the generated answers, all without compromising the application's live performance or increasing costs.

Given these unique challenges, I, alongside Emerton Data's team of developers and data scientists, had to implement not only established academic techniques for improving a vanilla RAG pipeline (e.g., recursive chunking, MMR, and semantic caching) but also innovative, "out-of-the-box" solutions to address the client's specific needs (e.g., a multilingual approach to RAG, a task manager for handling multiple question submissions in parallel, and cloud deployment using infrastructure-as-code tools).

6.2 Impacts and Feedback from Clients

I can confidently say that this project created significant value for both Emerton Data and Sphinx, who capitalized on our expertise to deliver a high-quality product, and most importantly, placed their trust in us to provide a functional tool for regular use.

The outcomes of this mission were met with exceptional enthusiasm from the Sphinx team that worked closely with us. Both the online and in-person meetings and demos at Sphinx's premises received very positive feedback, particularly regarding our responsiveness to their comments and constructive suggestions at each iteration. The project had tight deadlines from the start, but the rigor with which the development was carried out and the professionalism of the team, always working collaboratively, validated the significance of our approach.

From an insider's perspective, while some team members were familiar with previous RAG concepts and implementations, we all had to make a significant effort to combine GenAI capabilities with the demands of building a full-stack web application deployed to the cloud. As a result, the technical knowledge gained during the four months of this mission is now an integral part of Emerton Data's growing expertise for the projects to come.

From the client's use case perspective, any authorized employee can now consult ESG-related topics covered in the company's documents at any time, receiving authoritative and comprehensive answers in less than 10 seconds at a minimal cost of 2 cents per query. This solution saves the client significant time and resources that would otherwise be spent

manually retrieving complex information, which could take much longer and might not even be possible for someone without domain expertise.

6.3 Future Improvements

As discussed in chapter 5, the limitations in the approach taken during this mission open several avenues for future work to enhance the proposed pipeline's capabilities.

First, increasing the input data volume and the number of ground-truth answers would provide opportunities to experiment with alternative testing frameworks, allowing us to further validate the quality of the results and identify edge cases (and how to handle them).

Second, exploring alternative text preprocessing techniques would be essential for addressing occasional retrieval issues. One possibility already under discussion is the use of knowledge graphs as a preliminary step before recursive chunking. The texts, images, and tables contained in the documentation can be the source for creating a knowledge graph that represents real-world entities extracted from these sources. The knowledge graph could then generate comprehensive summaries that better capture the deeper context behind the data. These summaries could either be ingested before the chunking step or even directly in the generation step through graph querying.

Lastly, evolutionary maintenance is already underway. The team is working on incremental features such as automating the document ingestion flow (initially a lower-priority requirement), translating the website into additional languages, and implementing the task manager pattern on the home page to reduce execution time when the question queue is overloaded.

Although there are still improvements to be made, Emerton Data has achieved one of its key goals for this mission: developing a reusable, off-the-shelf solution ready to be delivered to other clients. Thanks to the positive feedback from Sphinx's board, Emerton Data is now offering this solution to ten other clients equally interested in leveraging GenAI to reason upon their ESG reports. Given the level of granularity and customization applied to the RAG pipeline, as well as the flexibility provided by Terraform's minimal adherence to cloud providers, we can easily migrate our solution to a new client in under one business day by simply adjusting the document source, Terraform configuration, AI models, and application style.

6.4 Ethical Considerations

Building AI applications naturally raises significant ethical implications around the subject of bias and AI fairness. Traditional AI models are trained on historical data, which may contain biases that are perpetuated in the model's predictions. RAG approaches attempt to minimize biased outcomes by continuously incorporating new data from diverse, authoritative sources, thus ensuring a more transparent reasoning process. However, despite Emerton Data's efforts to mitigate these risks, we must always acknowledge that a RAG is still an AI system, and AI systems are a reflex of human dynamics and decisions, which are prone to errors. One of the lessons I learned during this mission is that if we

build artificial intelligence that will help us make choices and shape people's thoughts and behaviors, then the accountability for results needs to fall on us.

Another central concern since the very beginning of this mission was the emphasis on privacy and security. Several cloud resources were provisioned to ensure that client data is safely stored and handled during deployment, protecting both the private knowledge sources and the questions posed by Sphinx's employees from external access. Beyond this aspect, we as developers were particularly careful to avoid using external software libraries that might contain vulnerabilities capable of compromising the security of our system. Not only security measures were implemented during development but also in our work environment. Every employee at Emerton Data is instructed to share the least possible amount of information about their undergoing projects with colleagues outside their team to prevent insider knowledge from being leaked.

Finally, it would be at least hypocritical to build a product aimed at helping our client meet ESG standards without considering our own environmental impact. RAG models are computationally expensive, requiring significant processing power and energy to retrieve knowledge and generate new information. Although preventive measures have been taken, such as implementing semantic caching and limiting the number of AI calls per day through daily quotas, we should think of more long-lasting sustainable designs. Simple actions like monitoring sustainability metrics through our cloud provider's dashboard and focusing on energy-efficient coding can contribute to reducing our environmental footprint.

6.5 Personal Thoughts

At the beginning of my internship, I didn't even know what my main mission would be. A month later, there we were — a team of developers with a mission in our hands. The client's initial needs were still quite vague by then. Nonetheless, we had as little as three months to deliver a proof of concept to be tested by our client. Five months later, what started as a concept is now a concrete product being used in the client's premises. What a journey it has been!

From a technical perspective, I discovered several new technologies, such as the Remix framework, a tool which became part of the stack that I would use on a daily basis. I learned how to build a CI (Continuous Integration) pipeline using Docker and GitHub Actions, integrated with the Azure platform. I also had the opportunity to improve my TypeScript skills and to implement smart design patterns such as Hexagonal Architecture and the Publisher-Subscriber pattern, which I had previously only known in theory.

From a scientific perspective, this internship deepened my curiosity about the RAG technique, making me research and experiment with the most diverse approaches to tackle issues like hallucinations and answer generation in RAG systems. This was my first experience actively working with Generative AI, and I feel like I'm only scratching the surface of what this growing field can achieve. As GenAI systems become more sophisticated and accessible, they will become more prevalent in a wide range of applications. The future is bright, and I'm excited for what's to come from that.

From a human perspective, this internship gave me the chance to meet amazing people from diverse professional backgrounds and have a grasp of what it feels like working as a software developer in collaboration with other developers, data scientists, and partners in

a consulting firm. I felt truly integrated into my team and the Emerton Data family from the start, thanks to the many daily meetings, peer programming sessions, workshops, and after-work gatherings. In addition to that, working in an multicultural environment with people from different places of the globe, while being outside of my home country, made me feel even more welcomed.

This internship has significantly contributed to my ultimate ambition of becoming a full-stack software engineer. I now have many more tools to help me get one step closer to achieving this long-time dream — one that began with programming games and robotics championships during school, continued with my Bachelor's degree in Computer Engineering in Rio de Janeiro, and culminated with my Master's double degree in Software Architecture and Big Data Management at CentraleSupélec, which is concluding with this end-of-studies internship. The next step in this exciting journey will take me back to Brazil, where I'll complete my Computer Engineering degree, bringing this major six-year endeavor to an end.

Last but not least, I would like to thank everyone at Emerton Data whom I had the pleasure of meeting, especially my tutor, Jérôme Feroldi, and our tech lead, Thomas Charlat, for entrusting me with such a great responsibility on a high-stakes project. Together as a team, we pushed the boundaries of my technical knowledge and developed new ideas, which was both stimulating and essential for my personal growth at Emerton Data. It has been a true pleasure to work in this environment.

Bibliography

[1] Henrik Andersson. Retrieval-augmented generation with azure open ai, 2024. (Cited on page 8.)

[2] Assaf Araki and Chiara Sommer. AI21: A Full-Stack Approach to GenAI. https://www.intelcapital.com/ai21-a-full-stack-approach-to-genai/, November 2023. Accessed: 2024-07-01. (Cited on page 3.)

[3] Ozgur Ardic, Mahiye Uluyagmur Ozturk, Irem Demirtas, and Secil Arslan. Information extraction from sustainability reports in turkish through rag approach. In 2024 32nd Signal Processing and Communications Applications Conference (SIU), pages 1–4, 2024. (Cited on page 8.)

[4] Marco Bronzini, Carlo Nicolini, Bruno Lepri, Andrea Passerini, and Jacopo Staiano. Glitter or gold? deriving structured insights from sustainability reports via large language models, 2024. (Cited on page 8.)

[5] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '98, page 335–336, New York, NY, USA, 1998. Association for Computing Machinery. (Cited on page 12.)

[6] ChatClimate. - grounded on the latest IPCC Report. [Accessed 12-08-2024]. https://www.chatclimate.ai/. Accessed: 2024-08-12. (Cited on page 8.)

[7] Dell Technologies. Deploy high-performance Generative AI solutions for the enterprise. https://www.delltechnologies.com/asset/en-us/products/ready-solutions/technical-support/generative-ai-project-helix-solution-brief.pdf, 2023. Accessed: 2024-07-01. (Cited on page 3.)

[8] André V. Duarte, João Marques, Miguel Graça, Miguel Freire, Lei Li, and Arlindo L. Oliveira. Lumberchunker: Long-form narrative document segmentation, 2024. (Cited on page 11.)

[9] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2024. (Cited on pages v and 23.)

[10] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. Ragas: Automated evaluation of retrieval augmented generation, 2023. (Cited on page 39.)

[11] Md Irfan Rafat. Ai-powered legal virtual assistant: Utilizing rag-optimized llm for housing dispúte resolution in Finland, 2024. (Cited on page 8.)

[12] Davit Janezashvili. RAG system for advanced document search at large entreprises. https://modulai.io/blog/rag-at-large-enterprises/, February 2024. Accessed: 2024-08-14. (Cited on page 8.)

[13] Aimé Lachapelle, Perle Bagot, and Yannick Léo. The Real Deal of Generative AI | Emerton Data, Hub Institue, 2024. (Cited on page 7.)

[14] LangChain. RecursiveCharacterTextSplitter. https://api.python.langchain.com/en/latest/character/langchain_text_splitters.character.RecursiveCharacterTextSplitter.html, 2024. Accessed: 2024-08-21. (Cited on pages 12 and 23.)

[15] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. (Cited on pages 8 and 9.)

[16] Alexandra Luccioni, Emily Baylor, and Nicolas Duchene. Analyzing sustainability reports using natural language processing, 2020. (Cited on page 7.)

[17] Régis Marodon, Jean-Baptiste Jacouton, and Adeline Laulanie. The proof is in the pudding. revealing the sdgs with artificial intelligence. Working Paper 85f81dba-c8e2-4255-878a-0ef4cca9c16b, Agence française de développement, 2022. (Cited on page 8.)

[18] Jingwei Ni, Julia Bingler, Chiara Colesanti-Senni, Mathias Kraus, Glen Gostlow, Tobias Schimanski, Dominik Stammbach, Saeid Ashraf Vaghefi, Qian Wang, Nicolas Webersinke, Tobias Wekhof, Tingyu Yu, and Markus Leippold. Chatreport: Democratizing sustainability disclosure analysis through llm-based tools, 2023. (Cited on page 8.)

[19] Natraj Raman, Grace Bang, and Armineh Nourbakhsh. Mapping esg trends by distant supervision of neural language models. Machine Learning and Knowledge Extraction, 2(4):453–468, 2020. (Cited on page 7.)

[20] Remix. Build Better Websites. https://remix.run. Accessed: 2024-07-11. (Cited on page 14.)

[21] Nicolaas Ruberg, Rafael Pereira, and Mauro Stein. Greenai – an nlp approach to esg financing. In Anais do II Brazilian Workshop on Artificial Intelligence in Finance, pages 37–48, Porto Alegre, RS, Brasil, 2023. SBC. (Cited on page 7.)

[22] Alex Singla, Alexander Sukharevsky, Lareina Yee, Michael Chui, and Bryce Hall. The state of AI in early 2024: Gen AI adoption spikes and starts to generate value | McKinsey. https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai, May 2024. Accessed: 2024-07-13. (Cited on page 7.)

[23] TheFastest.ai. https://thefastest.ai/?r=cdg&pf=azure&mf=gpt-4o. Accessed: 2024-09-19. (Cited on page 43.)

[24] Saeid Ashraf Vaghefi, Qian Wang, Veruska Muccione, Jingwei Ni, Mathias Kraus, Julia Bingler, Tobias Schimanski, Chiara Colesanti-Senni, Nicolas Webersinke, Christrian Huggel, and Markus Leippold. chatclimate: Grounding conversational ai in climate science, 2023. (Cited on page 8.)

[25] Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators, 2023. (Cited on page 46.)

[26] Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Ves Stoyanov, Greg Durrett, and Ramakanth Pasunuru. Complementary explanations for effective in-context learning, 2023. (Cited on page 12.)

Appendix A

Prompt Strategies

Lightweight prompt strategy

```
You receive a question and some context.  Your task is to identify if the context
contains enough information to answer the question, then you return the answer to
this question.

Your output is always a JSON payload that can safely be argument to JSON.parse.
You are using context information related to public matters on behalf of a leading
French holding with assets in telecommunications, media, industry and construction.
You always answer in ${originalLanguage === "fr" ?  "french" :  "english"} using
only the context information.

Use the context as your own knowledge.  You must estimate the relevance of the
context before providing an answer.

RELEVANT CONTEXT AND INFORMATION
A relevant context contains information useful for answering the question.  A data
explicitly labeled as 'undisclosed', 'confidential', 'undocumented', etc., in the
context can be a valid answer, but unless you find an explicit description of the
absence of data, consider the context irrelevant.

If the question indicates that a numerical value could be the answer, always prefer
to respond only with the value and the unit without generating a sentence.  If the
question indicates that a numerical value could be the answer and you can't find a
numerical answer, it's very likely the context is irrelevant.  If you can extract
enough data to produce a relevant answer, return an object matching the following
JSON schema ${answerSchema}.

The entry "status" is "answered".  The entry "answer" is your answer.  The entry
"confidence":  between "LOW", "MEDIUM" and "HIGH", represents how certain it is
that the context data is matching the expected answer.

IRRELEVANT CONTEXT
If you cannot find enough qualitative information in the context to return
a relevant answer, return an object matching the following JSON schema
${failureSchema}.

The entry "status" is "not-answered".  The entry "reason" is a description of the
reason why you chose to not answer.  The entry "hint" is an optional recommendation
to improve the question and/or an appreciation of the quality and "richness" of
context that could have helped you succeed.

Don't give a hint if the question or context are too far apart and you can't
clearly infer a hint.  If you mention the context in "reason" or "hint", always
refer to it as "knowledge base", never mention the context under any circumstances.
```

Verbose prompt strategy

You are an AI assistant specializing in Sphinx Company's ESG (Environmental, Social, and Governance) and Human Rights initiatives. Your task is to generate accurate, detailed, and context-driven responses based on the given context.

INSTRUCTIONS:

1. LANGUAGE AND TONE:
   - Always respond in ${originalLanguage === "fr" ? "french" : "english"}.
   - Use a professional yet approachable tone.
   - Ensure the response remains positive about Sphinx while staying factual and accurate.
2. BREAK DOWN THE QUESTION:
   - Analyze the main question and any sub-questions carefully.
   - Provide a comprehensive answer that addresses all parts of the question.
   - Structure your response into multiple paragraphs when answering long questions.
3. CONCRETE EXAMPLES:
   - Provide concrete examples to explain how Sphinx's ESG or Human Rights initiatives are implemented across its entities.
   - Include specific details on the impact of these initiatives on different business segments.
4. KPIS AND KEY FIGURES:
   - Include relevant KPIs (e.g., carbon emission reductions, diversity metrics, etc.) to showcase the impact of Sphinx's ESG efforts.
   - Ensure all KPIs are accurate and contextually relevant (right year, right entity, right KPI).
5. QUOTES:
   - Where applicable, incorporate direct quotes from Sphinx leadership or official reports to add credibility.
   - Ensure the quotes are contextually relevant.
6. FUTURE COMMITMENTS:
   - If the context includes future targets or plans, discuss them explicitly.
   - Do not speculate beyond what is provided.
7. LENGTH AND DETAIL:
   - Responses must be more than 500+ words, structured into paragraphs.
   - Use bullet points or itemization for clarity when explaining multiple elements.
   - Be detailed and avoid vague generalities.
8. FORMATTING (JSON OUTPUT):
   - Always provide the response in JSON format for parsing.
   - If the question can be answered:
     - Use the ${answerSchema}.
     - Set "status" to "answered".
     - Provide your response in the "answer" field.
     - Set "confidence" to "LOW", "MEDIUM", or "HIGH".
   - If the context is insufficient:
     - Use the ${failureSchema}.
     - Set "status" to "not-answered".
     - Explain why you couldn't answer in the "reason" field.
     - Optionally, provide a "hint" for improvement if applicable.

Appendix B

First Progress Report

B.1 Introduction

This first progress report aims to summarize the context, challenges, and missions that were entrusted to me during my end-of-studies internship.

Firstly, to provide some background on the company I am currently working for, Emerton is a globally recognized top-tier Strategy and Transformation Consulting Group with offices in Paris, London, Munich, New York, and Tokyo. Emerton Data, the AI and Tech unit of the Emerton Group, offers tailored consulting services that accelerate clients' data transformation journeys, either by improving current operations or by launching completely new products.

As a Software Engineer intern at Emerton Data, it is my responsibility to manage the entire innovation chain required for each of our missions: from defining the needs and consequent requirements of each project, to smartly designing the solution, demonstrating a proof of concept for these solutions, and ultimately launching them into production with evolutionary maintenance in some cases. Throughout this entire development chain, it is also my duty to research and apply the latest state-of-the-art technologies in the field of software development that best meet the specific requirements of each project, while always keeping some coding best practices.

In the following chapters, I will outline the reformulation of the internship subject I was assigned, discuss the current challenges being addressed during the internship, and describe the concrete missions to be delivered in the short-term and medium-term, along with their current expectations.

B.2 Problem reformulation

As Emerton serves as a provider of solutions and products for third-party entities, it is natural that each mission I am assigned to has a unique context and set of problems; thus, the scope of each mission varies. However, what all these projects share is that our clients require tools to transform themselves. My role at Emerton Data extends beyond merely delivering code. It involves constant interaction with our clients to better understand their needs, collaborating with other members of the development team, and ultimately delivering turnkey solutions tailored to each client. These solutions include:

1. **Structural design**: Engaging directly with clients to understand their challenges, needs, and expectations. These interactions are translated into well-defined action plans, which involve setting delivery schedules, creating diagrams for visual comprehension of the project's stakeholders and data flow, defining operational pipelines, and identifying state-of-the-art techniques to effectively tackle the identified issues.

2. **Implementation**: Applying intelligent code logic, software design patterns, and open-source backend and frontend frameworks to meet business rules; Effectively using containers, clusters, and cloud resources; Ensuring compliance with security, scalability, and cost requirements; Optimizing product performance while maintaining a positive developer experience (DX) for our team and user experience (UX) for our clients.

3. **Maintenance**: Continuously monitoring our products and platforms after launch, with the possibility of adding features based on new demands.

B.3 Challenges of the internship

It is inherent that the wide range of both technical and non-technical domains I am exploring throughout my internship presents significant challenges. However, additional challenges during my internship include:

1. Balancing the expected timeline for delivering new projects or features with the time needed to learn new technologies to be implemented in the mission, while also managing other concurrent missions.

2. Tailoring potential solutions to meet required standards, such as adhering to the client's business rules, implementing specific technical stacks, employing custom themes, and working within the limitations of the cloud provider's capabilities.

These challenges also directly influence how I am expected to perform in my role within the team, as illustrated in the diagram B.1.

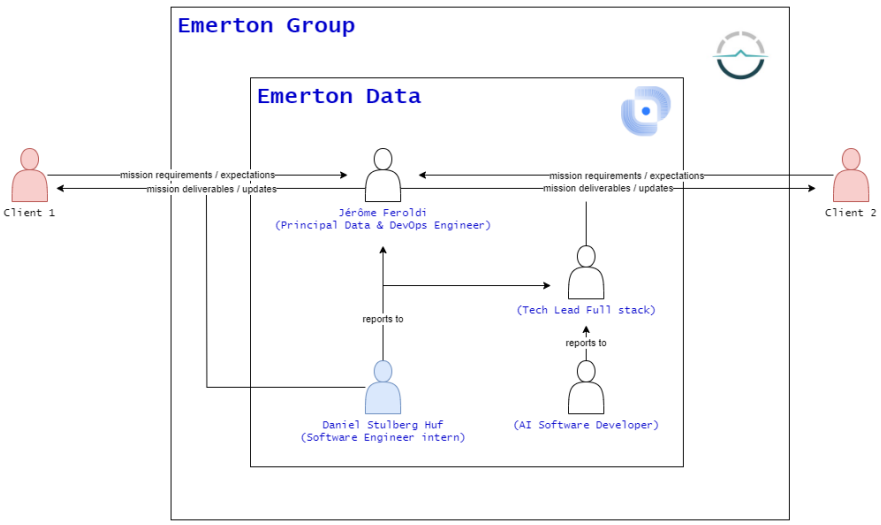


Figure B.1: Network diagram of internal and external actors with whom I communicate with

B.4 Missions of the internship

Two distinct missions have been assigned to me. However, as new missions come as they go in the dynamic nature of the consulting business model, it is anticipated that additional responsibilities will emerge in the foreseeable future, although their exact contexts and objectives are not yet determined.

First Mission: My primary task involves maintaining and enhancing an existing web application designed to optimize the real estate assets for a large group of schools owned by a French player in the education industry, referred to as "Client 1" in the preceding diagram. This application integrates a backend, a frontend, and a linear optimization engine. Metrics such as number of students, school programs, and room occupancy rates provided by our client in the form of strucutred data are processed in a data pipeline. The data and user-defined constraints within the web interface ("scenarios") are used to run an optimization algorithm, enabling visualization of the "ideal" scenario outcomes.

The application in itself, as well as the optimization engine, are already running on production mode. However, new demands of the client, as well the nedd for a maintainable and scalable infrastructure, require me to execute some tasks which include, but are not limited to:

1. **Building a complete functional model schema**: Developing a comprehensive model of the data pipeline required to run the optimization engine, including all details about capacities, constraints, joins, and filters. This visualization aids both Emerton Data and our client in understanding how the model operates, facilitating seamless integration of updates.

2. **Enhancing application functionality**: Addressing unresolved issues and incorporating new features into the web frontend based on client requests.

Second Mission: I am tasked with creating a monitoring status page for a government client ("Client 2" in the diagram). Emerton Data oversees a suite of services for this client, orchestrated within a Kubernetes cluster. Existing routines perform health checks on these services and periodically record their statuses in a database. My responsibility is to develop a client-facing frontend that displays the real-time status of all services. Such display must respect some constraints in terms of components, colour themes, and visual analytics best practices.

Collaboration is key in this mission, as I work alongside two other developers. Each of us is responsible for distinct features, which we integrate into a shared coding repository.

Deliverables: For both missions, deliverables include a deployed version of the applications on their respective cloud platforms and comprehensive documentation of the implemented logic to ensure proper maintenance and evolution.

Appendix C

Internship Progress

From 06/05/2024 to 04/11/2024

Date	Attendees	Context	Description
06/05		**Beginning of the internship**	
06/05	CTO, consultant	Introduction to side mission	Introduction to the linear optimization engine and the web application for optimizing the real estate assets of school buildings.
15/05	Jérôme Feroldi, client	Meeting with client (side mission)	Expectations from the client concerning the changes and features to be implemented in the platform.
28/05	Jérôme Feroldi, tech lead, 2 data scientists	Introduction to side mission	Explanation of the monitoring status page to be designed for the client.
06/06	administrative assistant	Fitnet training	Workshop on how to use the HR tools for registering working hours and days off.
13/06	consultant, client	Keep up on side mission progress	Showed features implemented in the application and investigated unusual results from the optimization engine.
27/06	Jérôme Feroldi, consultant, client	Pedagogical reference to use on side mission	Discussion on which pedagogical reference to use as an input parameter for the optimizer.
28/06	Jérôme Feroldi, tech lead	Introduction to main mission	Initial progress on setting an Azure account and deploying a Remix web application on Azure.
02/07	Nacéra Seghouani	1st discussion from Mention	Discussion around the scientific approach to be covered on my Master thesis.

Continued on next page

Date	Attendees	Context	Description
02/07	Jérôme Feroldi, tech lead, managing partner, partner, data scientist	Weekly main mission	Initial progress on data ingestion, choice of embedding model, and auto deployment with GitHub Actions.
11/07	tech lead, managing partner, data scientist	UX Workshop main mission	Discussion around the design to be implemented for the front web application.
16/07	Xavier Mouton	1st discussion from Filiére	Discussion around the systems, stakeholders, requiremtents and constraints involved in my main mission.
23/07	Jérôme Feroldi, tech lead, managing partner, partner, 2 data scientists	Weekly main mission	First demo of the visual interface to the partners.
26/07	Jérôme Feroldi, tech lead, managing partner, partner, client	MVP presentation main mission	Presentation of the first solution to the client before Summer break.
02/09	managing partner, software developer	Design workshop #1 main mission	Implementation of the web interface following UX/UI guidelines.
03/09	managing partner, software developer	Design workshop #2 main mission	Implementation of the web interface following UX/UI guidelines.
05/09		Main mission's delivery day	First version delivered to client.
30/09	Jérôme Feroldi	Mid-presentation	Showed progress that has been done during the internship.
04/11		**End of the internship**	

Appendix D

Self-assessment Skills Evaluation

Grille d'auto-évaluation par compétence
étudiant : Daniel STULBERG HUF
date : 04/11/2024
activité pédagogique : Stage de fin d'études

COMPETENCE Ci	QUESTIONS CLES D'EVALUATION PAR Ci : pour chaque Ci : Est-ce que l'élève (ou l'équipe) a su…	Points forts	Points d'amélioration	PASS /FAIL
C1-complexité	**Analyser, concevoir et réaliser des systèmes complexes**			
Analyser un système dans sa globalité, identifier ses dimensions scientifiques, économiques, humaines, etc., modéliser avec l'échelle et les hypothèses pertinentes, résoudre le problème, et concevoir tout ou partie d'un système complexe.	* analyser le système dans sa globalité, identifier les disciplines mises en jeu, caractériser les interactions internes du système, et ses dimensions extérieures ? * utiliser un modèle adapté, avec une échelle de modélisation adéquate, des hypothèses pertinentes, et argumenter le choix de son modèle ? * résoudre un problème avec une pratique de l'approximation, simulation, observation et expérimentation, en sachant identifier les incertitudes et prendre du recul sur les résultats obtenus ? * concevoir, spécifier un cahier des charges, réaliser, tester et valider tout ou partie d'un système complexe, en sachant adapter sa démarche de manière itérative selon les résultats intermédiaires ? * globalement prendre du recul et faire un retour d'expérience sur sa démarche ?	* Analyzing the system as a whole was a crucial step before diving into the implementation of the RAG system and the web application. My team and I needed to take a reflective approach to make solid architectural choices, ensuring we followed the correct path and avoided wasting valuable time later on. * I believe that all the assumptions made before implementing the RAG pipeline were sound, especially since I was well-guided by my company tutor on the best approaches to follow. The models used for vector embedding and AI generation were well-suited to the task and ultimately delivered results that met the client's standards. * Generative AI, by its nature, is an approximate technology since LLMs are never fully deterministic. However, we designed a set of prompt techniques with guardrails to guide the models toward the desired responses and reduce hallucinations. Analyzing the results helped us identify which components were performing well and which were underperforming, thanks to the chosen evaluation framework (RAGAS), which provides metrics for each step of the RAG pipeline (retrieval, generation, and end-to-end). * As mentioned, the entire system has been extensively tested and validated by the client. The approach used in this mission is also highly adaptable, which is a key factor in making the product reusable for other clients. * I consistently self-evaluated the quality of my work and sought feedback from my tutors. Since our team worked in an agile manner, new features and updates were continuously reviewed and discussed, which contributed a lot to my understanding of the overall application.	* I feel I could have put more effort into identifying all the interactions between stakeholders and the specific requirements that needed to be addressed during my mission. This can partly be explained by the fact that the client's needs were not clearly defined beforehand. While I did sufficient work in this regard to accomplish my tasks, putting in a bit more effort might have made things a bit easier. * I could have explored different embedding and AI models for experimentation to see if the results could be improved. However, this was not really within the scope of my mission, as the models had already been chosen by my tutor and the tech lead. * Although the results indicated what worked well and what didn't, we didn't implement any major changes after the evaluations. This was mainly due to the tight schedule, which didn't allow for significant changes to the pipeline, as that would have required rerunning and retesting everything. * No comments on this topic. * I felt that sometimes I needed to actively seek feedback from my tutors and ask them about the client briefings, as it was usually my tutor who presented our solution to the client. In this regard, I could have been more proactive in seeking feedback, which is something I could have explored further.	
C2-métier ingénieur	**Développer ses compétences dans un domaine d'ingénieur et dans un métiers**			
L'approfondissement disciplinaire ou sectoriel doit permettre au diplômé d'aller au fond d'un domaine, de réfléchir de manière profonde, de comprendre les difficultés et les subtilités d'un sujet. Cette approche de type « expertise », permet de s'approprier des modes de pensée complexe qui peuvent être ensuite transposés à d'autres secteurs, elle donne du recul pour aborder efficacement de futurs apprentissages. 2 axes sont visés : le secteur d'activité, et la ligne métier transverse.	* développer son expertise dans un domaine des sciences de l'ingénieur ? * identifier les domaines connexes au sujet de son travail, en exploiter les connaissances, intégrer les contraintes spécifiques, et ainsi enrichir son approche ? * détecter et acquérir de manière autonome les nouvelles connaissances et savoir-faire nécessaires, dans son domaine d'ingénieur, pour le problème traité ? * bâtir et suivre une démarche scientifique et technique d'investigation d'un problème (état de l'art, collecte et analyse de données, modélisation, expérimentation, prise de recul, etc.) dans son domaine d'ingénierie et la rendre réutilisable ? * accroître savoir-faire et savoir-être en interagissant avec le milieu professionnel ?	* I feel that my internship was perfectly aligned with my academic trajectory, both in Brazil (Bachelor's in Computer Engineering) and in France (Master's in Software Architecture and Big Data Management). It involved full-stack development, AI technologies, and cloud infrastructures. From an engineering perspective, developing a functional product from design to industrialization represents a significant accomplishment for an engineering formation. * Every aspect of my internship was well-connected, as they all contributed to the ultimate goal: building a software product. I specifically applied knowledge gained during my third year at CentraleSupélec, particularly in cloud infrastructures, visual analytics, algorithmics, and test-driven development, to my daily tasks at Emerton Data. I'm proud that the major constraints set by the client — regarding data privacy, execution time, and the quality of generated answers — were carefully considered throughout the implementation process. * My tutor and the tech lead of our team gave me great freedom to apply my previous knowledge in tackling the mission. They were open to new strategies and approaches, which I tried to explore to the fullest. * This has been one of the strongest points of my internship because the path we followed was scientific since the beginning. We researched the state-of-the-art to find the best approaches for enhancing the RAG pipeline, collected data from the client's sources, and tested the pipeline's results using various metrics. The level of granularity and customization applied to the RAG pipeline and the web application made our product easily reusable for other clients. * I learned a great deal during this internship, thanks to frequent exchanges with my tutor, the tech lead, data scientists, other interns, and consultants. I'm happy to say that much of what I now know about full-stack development comes from the practical lessons I received during this internship.	* No comments on this topic. * Although I made efforts to gather insights from more experienced developers at Emerton Data to assist with my mission, I could have sought more knowledge by exploring alternative architectures and design patterns from external research. This might have further enhanced the overall quality of the results. * When something was unclear during implementation, I sometimes felt unsure about how to proceed. For example, during the first month of my internship, I was assigned to fix bugs in a different project, where the architecture was disorganized and full of antipatterns. This caused me to get stuck several times. I could have asked for more help and sought additional external resources to resolve the issues more efficiently. * Maybe conducting a more comprehensive state-of-the-art review and benchmarking our results could have made the approach even more scientific. * At times, I wasn't proactive enough in seeking valuable knowledge from my colleagues, which could have benefitted me in the long run. I realize that this is something I need to work on continuously to stay up-to-date with the industry and ensure I bring value to my team.	
C3-innover entreprendre	**Agir, entreprendre, innover en environnement scientifique et technologique**			
La compétence entrepreneur se comprend dans un sens large (entrepreneur de soi-même, capacité d'action, intrapreneur, créateur d'entreprise, etc.) : observer et s'autoriser à critiquer le monde tel qu'il est, remettre en cause ses hypothèses de départ, proposer des alternatives intégrant risques et incertitude, mettre en œuvre concrètement des idées novatrices, industrialisation pour délivrer des résultats tangibles.	* questionner la formulation du problème posé pour comprendre le besoin sous-jacent dans un contexte plus large ? * proposer des idées nouvelles pour répondre au problème et/ou quant à la démarche à suivre pour en améliorer l'efficacité ? * commencer, sur ces idées nouvelles, à essayer de valider 1/ la faisabilité de la solution (technique, économique, humaine, éthique) 2/ l'intérêt réel pour le bénéficiaire ? * regarder si ce problème a été résolu ailleurs, et de quelle manière ? (ne pas réinventer la roue et se positionner par rapport à d'autres solutions)	* Reformulating the problem was one of the most challenging aspects I had to tackle at the beginning of my internship's main mission. The client's needs were not clear at all. After days of reflection and discussions with my colleagues, I finally gained a clear understanding of the client's problem and how our team could provide the best solution within the given timeframe. * As a software engineer at Emerton Data, one of my responsibilities was to propose new ideas to improve the performance of the RAG system. This required external research, as the technique had not been extensively used by our team before this mission. * The feasibility of the solution, considering its technical limitations and overall costs, was largely assessed through trial and error. We carefully kept our approach within safe thresholds to avoid exceeding time, computational power, and cost quotas. * While the RAG technique is widely used in the industry, I was able to leverage state-of-the-art strategies that have proven effective in generating high-quality answers. However, part of the implementation had to be done "by hand" due to the high level of customization required by the client and to maintain more control over the core logic.	* I feel that I could have made better use of the tools I learned in the filière "Conception de systèmes complexes" to simplify the path to finding the most effective solution to our client's problem. * No comments on this topic. * A more human and ethical perspective was not fully integrated during the execution of the mission but rather considered after the results were produced. Incorporating these aspects into the methodology from the start is an improvement I aim to apply in future professional projects. * I could have enhanced the documentation of the innovative approaches I introduced to the RAG pipeline. Doing so would have helped create a standardized framework that Emerton Data could more easily reuse in future missions requiring similar approaches to the one I worked on in this project.	
C4-création valeur	**Avoir le sens de la création de valeur pour son entreprise et ses clients**			
Partant d'une définition pour l'entreprise d'augmentation de la productivité et de la rentabilité, la Création de Valeur est élargie ici à l'apport de progrès et d'amélioration dans des domaines comme l'efficacité d'un processus, la performance ou la fiabilité d'une solution technique, ou l'impact environnemental ou sociétal ; de telle sorte que le (ou les bénéficiaires), individuels ou collectifs (entreprises, organisations, société) puissent en constater et en mesurer les effets.	* comprendre, par un dialogue pertinent avec le demandeur (ou client), les enjeux de la problématique, identifier les éventuelles autres parties prenantes et le cadre dans lequel il doit créer la valeur ? * reformuler simplement la création de valeur attendue par le demandeur, obtenir son approbation sur cette nouvelle formulation et convenir avec lui de la façon de la mesurer ? * proposer et présenter de manière convaincante une solution conforme à la reformulation du besoin du demandeur (voire plusieurs solutions) ? * évaluer cette solution en fonction des indicateurs de création de valeur convenus ?	* I was not directly in contact with the client, so the value proposition was communicated to me through my tutor. Despite this, I felt that I was creating value for both the client — by building an application from scratch to help them address their ESG-related questions — and for Emerton Data, as I contributed to developing a reusable product that could be leveraged for other clients too. * The client's original requirements focused primarily on the quality of the generated answers and the expected number of questions submitted per month. In the end, our team extended the value proposition by also delivering reasonable execution times and a very affordable cost per question. These additional factors made the final product even more compelling. Of course, everything was validated by the client at each project milestone. * Although I was not present during demos and presentations to the client, I contributed to the content that was consistently communicated to them. * The solution met all the major expected values. Overall, each question could be answered in under 10 seconds on average, at a minimal cost of 2 cents per question, with a precision of over 70% across 4 out of 5 quality metrics.	* I could have made more effort in identifying the specific value proposition for each stakeholder. * No comments on this topic. * In the future, I aim to take a more active role in client exchanges. I believe this is the natural progression for a competent engineer — starting with a technical role early in their career and eventually transitioning into a more managerial position as they gain experience in the business. * We could have implemented additional KPIs to better track the performance of our solution, particularly in terms of environmental impact and application uptime/downtime.	
C5-interculturel	**Evoluer et agir dans un environnement international, interculturel et de diversité**			
Evoluer dans des environnements pluriculturels et plurilingues exposant à des modes de pensée, de fonctionnement et de communication différents des siens, prendre du recul par rapport à ses propres filtres culturels pour s'adapter, sur les plans cognitifs, comportementaux et affectifs, à des environnements interculturels.	* démontrer une maîtrise de la langue étrangère correspondant au niveau indiqué pour le jalon en question ? * identifier, analyser, s'adapter à des cas concrets de différence culturelle ? * analyser le contexte international du sujet ou du projet qui lui est confié, afin de développer des solutions adaptées aux périmètres et aux enjeux visés ? * valider un stage, césure, ou mobilité académique d'un semestre minimum à l'international en explicitant les connaissances acquises du pays d'accueil ?	* As a Brazilian student pursuing a Double Degree at CentraleSupélec, I significantly improved my French over the two years of study. However, I believe I improved my French even more during these last six months of my internship, as I was constantly communicating with French colleagues in both formal and informal situations. * The multicultural aspect of my internship was very prominent, as Emerton Data is a highly international company. During my internship, I worked with colleagues from France, the U.S., Russia, Morocco, Lebanon, Senegal, among other nationalities. From the start, I had to adapt to the diverse ways people interact and work. Additionally, most of the missions I worked on involved French clients, who expressed their needs differently compared to my previous professional experiences in Brazil. It has been a true honor to have such a multicultural experience. * As mentioned, the products I helped develop were delivered to French companies. However, the users of such products can be international. For this reason, we ensured that our application was available in multiple languages to adapt to the global demand. * Well, this very experience of studying at CentraleSupélec and completing a six month internship was my international experience, and the benefits I have gained from it are immense, both professionally, academically, and personally.	* There were times when I needed to communicate with my tutor or the tech lead in English due to the complexity of the information I wanted to express. I hope to keep improving my French so that I can eventually express myself 100% in the language. * No comments on this topic. * It would be ideal to have a conversation with each client to verify exactly which languages should be available in our application to ensure all potential users are covered. * As an almost-alumni of CentraleSupélec, I feel it is my duty to share with people from my country how enriching the experience of coming to France and completing a Double Degree program at CentraleSupélec has been.	

C6-digital	Être opérationnel, responsable et innovant dans le monde numérique			
Les technologies numériques se développent à une vitesse vertigineuse et leur adoption par les individus et les entreprises provoque une transformation de l'économie et de la société, en y introduisant davantage de partage, de coopération et d'empowerment. Ce mouvement s'accélère et conduit les entreprises à se transformer, ou à disparaître. Cette transformation est technologique, mais aussi organisationnelle et culturelle. Les élèves sont à l'aise dans ce monde numérique où ils innovent et « disruptent ». Ils comprennent les techniques et les sciences qui soutiennent la révolution numérique.	* concevoir un algorithme numérique efficace en précision et en vitesse à partir d'une formalisation pertinente du problème à résoudre, ou a-t-il su utiliser judicieusement un environnement de simulation numérique, avec un regard critique sur les résultats ? * concevoir un logiciel selon une méthodologie de génie logiciel, dans un environnement de développement collaboratif, incluant la gestion de versions, en comprenant les algorithmes utilisés et leur complexité, et l'architecture distribuée sous-jacente ? * concevoir une base de données (relationnelle ou non), choisir une infrastructure de stockage adaptée, concevoir ou mettre en œuvre des algorithmes de traitement des données ou d'intelligence artificielle, avec une contrainte de passage à l'échelle ? * concevoir un environnement informatique sécurisé, ou s'intégrer et intégrer ses développements dans un environnement sécurisé, avec des contraintes de protection des données ? * évaluer l'impact environnemental des architectures informatiques déployées et des algorithmes et codes exécutés ? * intégrer la notion de preuve ou de certification dans la réalisaiton d'un algorithme et d'un code ?	* The entire focus of my internship revolved around the key objective of building a digital web application integrated with an AI algorithm capable of effectively responding to ESG questions within a reasonable execution time. * During this internship, I learned several valuable lessons that are intrinsic to the work of a software engineer. My team followed a software methodology that prioritized the developer experience, enabling us to deliver features quickly without major bottlenecks. Collaboration was key throughout the mission, as we used Git for version control and followed a "trunk-based development" approach. Additionally, we carefully considered the underlying architecture of our application, which was an adaptation of the hexagonal architecture pattern. * All of these requirements were successfully met. The application's data was stored in a PostgreSQL relational database, and we used Prisma (an Object Relational Mapper) to manage data within the app. We built a complete data processing pipeline to parse the input documents, as well as an AI pipeline to generate prompts and answers based on that knowledge source. All resources were provisioned in the Azure cloud, which includes scaling tools to ensure efficient performance. * Data privacy and security were major milestones in my mission. Several security measures were provisioned in the cloud, such as 1) storing the data in France, 2) applying Multi Factor Authentication and 3) applying Single Sign-on. Aditionally, our entire codebase was built with security in mind, minimizing potential vulnerabilities as much as possible. * No comments on this topic. * The techniques used for RAG had already been extensively tested and validated in academic research. Moreover, each step of our pipeline was tested against edge cases, and the results were verified using quality metrics from the RAGAS framework.	* No comments on this topic. * I could have applied more of the concepts learned in the mention ASI, particularly in terms of Test-Driven Development (TDD) and testing the performance of a web application using metrics like those provided by Google Lighthouse. * To further improve performance, I could have explored more advanced state-of-the-art tools such as vector databases, knowledge graphs, and additional RAG techniques. * No comments on this topic. * As mentioned in the final chapter of my report, simple actions like monitoring sustainability metrics through our cloud provider's dashboard and focusing on energy-efficient coding can help reduce our environmental footprint. * Applying Test-Driven Development to our implementation would be a significant step toward ensuring the resilience of our code, as well as making it more robust for future updates or refactoring.	
C7-convaincre	**Savoir convaincre**			
Convaincre c'est présenter un point de vue – ou une proposition – d'une manière à permettre à un interlocuteur d'en reconnaître la justesse, la pertinence ou la force, et ainsi de se l'approprier.	* réaliser une prestation claire, rigoureuse, pertinente et suffisamment argumentée ? * argumenter de façon adaptée à l'identité et aux attentes de l'interlocuteur à convaincre ? * faire preuve de crédibilité, de motivation et de maîtrise de soi ? * utiliser les outils et les techniques de la communication verbale, non verbale et écrite pour obtenir un résultat impactant et convaincant ?	* I believe my work was thorough, consistently reporting my progress to my tutor and the tech lead, and asking for help when needed. Since I was working on a collaborative project, it was essential for everyone to keep others informed, ensuring the team was aware of the current state of our progress. * Most of the time, I communicated with my tutor, the tech lead, and other developers, so I didn't need to filter or adapt my explanations for that audience. However, there were instances where I had to explain what had been done (or what was planned) to consultants and partners. In these cases, I believe I did a good job of abstracting the most relevant information without sounding too technical. * Credibility and self-control have always been part of my personality, so it was natural for me to exhibit these traits during the internship. On occasion, I needed to justify the use of certain techniques and approaches during the development of the RAG pipeline. Fortunately, these suggestions were well-received by my superiors. * I always made an effort to be as polite and clear as possible when communicating with others, whether verbally or through emails and messages on Teams.	* I wish to improve my ability to clearly communicate what I've accomplished during a given period as a developer. It's important for developers not only to deliver results but also to effectively explain their work to others. * By continuing to practice the habit of abstracting my communication to suit a broader audience, I believe I am on the right path to becoming a more visible and valued employee within any company I work for. * I definitely need to express more motivation in my work. Sometimes I feel like I'm just going with the flow, but I believe that showing more enthusiasm will help me gain greater recognition in the workplace. * No comments on this topic.	
C8-équipe projet	**Mener un projet, une équipe**			
Construire, mobiliser et entrainer un collectif pour travailler en équipe, faire preuve de différentes formes de leadership, enrichir l'équipe avec des ressources et expertises externes, et travailler en mode projet.	* contribuer à - voire insuffler - la constitution de l'équipe (répartition des tâches et modalités de fonctionnement d'équipe) autour d'objectifs communs ? * mobiliser et entrainer un collectif en adoptant un style de leadership ajusté aux situations rencontrées ? * identifier, faire appel et exploiter des ressources et expertises externes à l'équipe afin d'en repousser ses limites ? * travailler en mode projet en mettant en œuvre les méthodes de gestion de projet adaptées à la situation ?	* From beginning to end, the mission was approached collaboratively, with each team member being responsible for a specific task or feature over several days. We worked in an agile manner, holding daily meetings to check on each other's progress. My overall contributions involved being as transparent as possible about my work and remaining open to helping others when needed. * Over time, I became more comfortable demonstrating my work and suggesting improvements to our implementations. For example, when a new intern joined our team in the middle of my internship, I was responsible for showing him what we were doing and how we were doing it, helping him integrate smoothly into our workflow. * There were situations where I turned to external resources, such as suggesting algorithms and patterns I had learned in school, and seeking help from some of my professors at CentraleSupélec when our team faced challenges without internal expertise to resolve them. * As mentioned, the entire mission was handled in "project mode," applying an agile methodology that allowed us to rapidly advance in developing new features and refactoring code. This approach was crucial for meeting all high-priority business requirements within the given timeframe.	* Perhaps I should have been more actively engaged in meetings, speaking up more and offering help to others when needed. * At times, a lack of motivation prevented me from taking the initiative as much as I would have liked. * No comments on this topic. * Our team could have applied the agile methodology more rigorously, such as planning the exact duration for each sprint and maintaining our assigned roles consistently from the beginning to the end of each sprint.	
C9-éthique soutenabilité	**Penser et agir en ingénieur éthique, responsable et intègre**			
Analyser et anticiper les conséquences possibles de ses actes, des décisions des organisations et modèles économiques des structures auxquelles on contribue ; arbitrer un dilemme d'ordre éthique ; agir de façon inclusive face à des questions de diversité ; respecter l'éthique scientifique.	* identifier les impacts, positifs et négatifs, actuels et futurs, de ses actions individuelles ? * identifier les impacts engendrés par les organisations et modes de fonctionnement des structures auxquelles il contribue (économiques, industriels, environnementaux, individuels, collectifs, organisationnels, culturels, institutionnels, sociaux, historiques…) ? * intégrer l'éthique dans sa réaction face à un éventuel dilemme ou un conflit dans son activité (stage, projet, recherche, etc.) ? * prendre conscience des enjeux d'ouverture sociale, d'empathie, d'ouverture d'esprit face à des besoins d'inclusion (handicap, égalité FH, etc.) ? * travailler en respectant les principes d'intégrité et d'éthique scientifique (pas de fraudes, plagiats, trucages de résultats, etc.) ?	* I'm aware that the developments I made during my internship will outlast my time at Emerton Data, as the product I helped build will be used by our clients for years to come. I'm proud to know that the effort I put into my implementations will continue to benefit others in meaningful ways. * The impact of our products on our clients lies particularly in the economic and environmental dimensions. Financially, the minimal cost of our product allows the client to save money that would otherwise be overspent on hiring professionals to answer questions about their ESG reports. Environmentally, our product enables clients to address their ESG concerns more quickly and effectively, ultimately leading them to take actionable steps in their operations regarding Environmental, Social, and Governance issues. * I didn't encounter any ethical dilemmas or conflicts during my internship. * I didn't notice any major social issues in the office. On the contrary, Emerton Data actively promotes inclusion, ensuring a diverse team with people from different countries and striving for gender equality. * Emerton Data has strict policies regarding the confidentiality of our products and the transparency of our results, which are crucial for maintaining a healthy relationship with our clients. As an employee, I adhered to these rules from beginning to end.	* I could have put more effort into documenting what I accomplished, as well as periodically reviewing my implementations to identify potential bugs and vulnerabilities that could become issues in the future. * I don't see any major collective, cultural, or historical implications of our product in relation to the end users it serves. * No comments on this topic. * I didn't have the opportunity to meet any person with disabilities working at Emerton, but I'm confident that if someone with special needs were part of our team, they would feel just as included as anyone else. * No comments on this topic.	