
CERN


# A CERN Knowledge Retrieval Chatbot

## (Harnessing the power of AI for efficient information retrieval)

Verena Kain, Florian Rehm, Sofia Vallecorsa, Giovanni Guerrieri and **Manuel Guijarro**

# Introduction

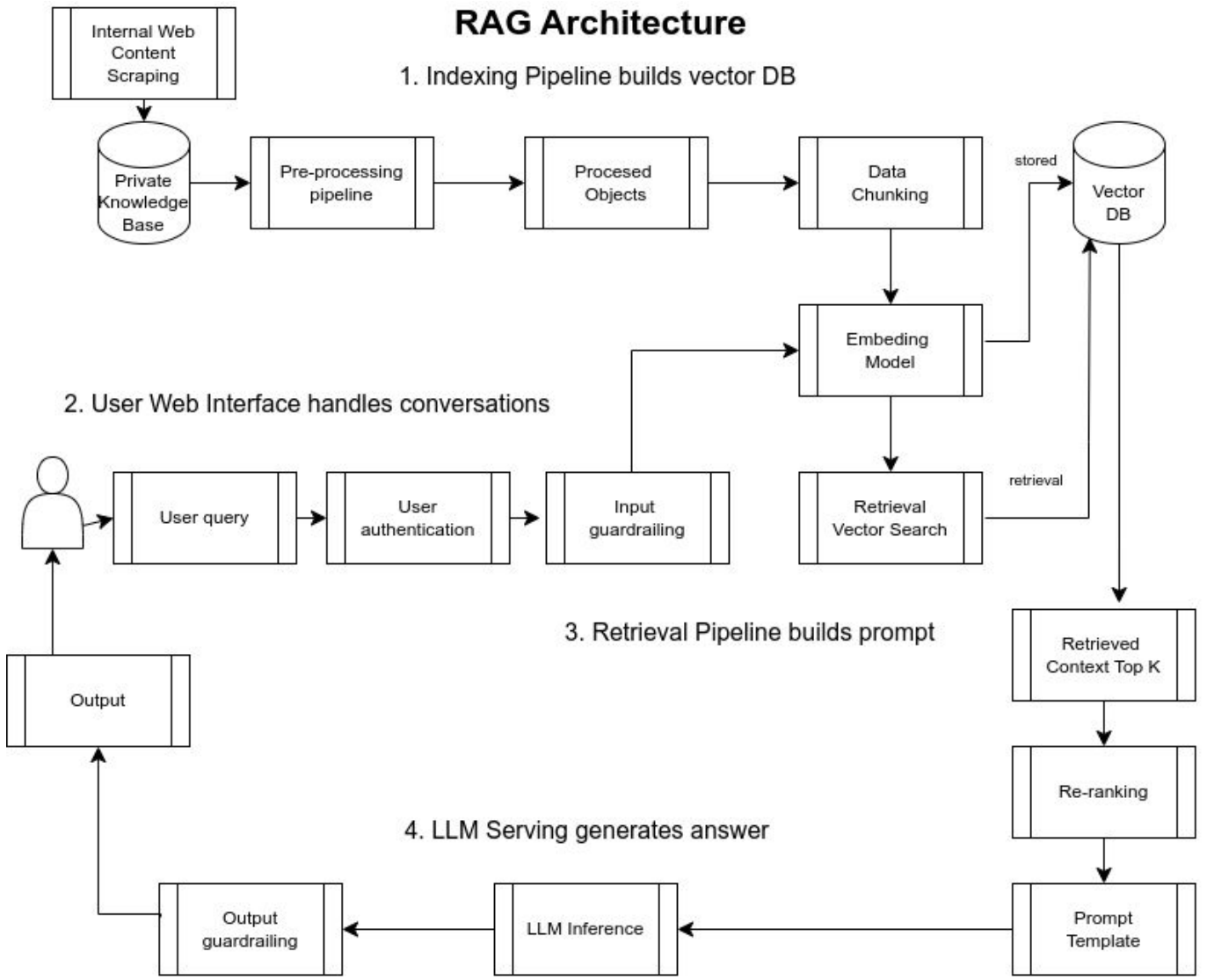
- Problem:
  - Finding relevant information within CERN's extensive internal documentation is complex and time-consuming.
- Solution:
  - AccGPT - an AI-powered chatbot that leverages Natural Language Processing (NLP) for knowledge retrieval.
- Goal:
  - Create a purpose-built chatbot using open-source Large Language Models (LLMs).
- Future Potential:
  - Assist in code development and other tasks.

# Motivation and Demand

- Growing demand for LLMs at CERN for various applications:
  - AI Chatbots
    - Information Retrieval: User Support, HR selection processes, …
    - Summarization and Minutes generation
  - Code Assistants (IDE integrated)
    - Code Generation
    - SQL Generator/Executor
  - Application use cases:
    - Zenodo Information Categorization
    - Transcription and Translation Services (TTaaS)
- Easy and efficient access to LLMs is crucial for widespread adoption of GenAI

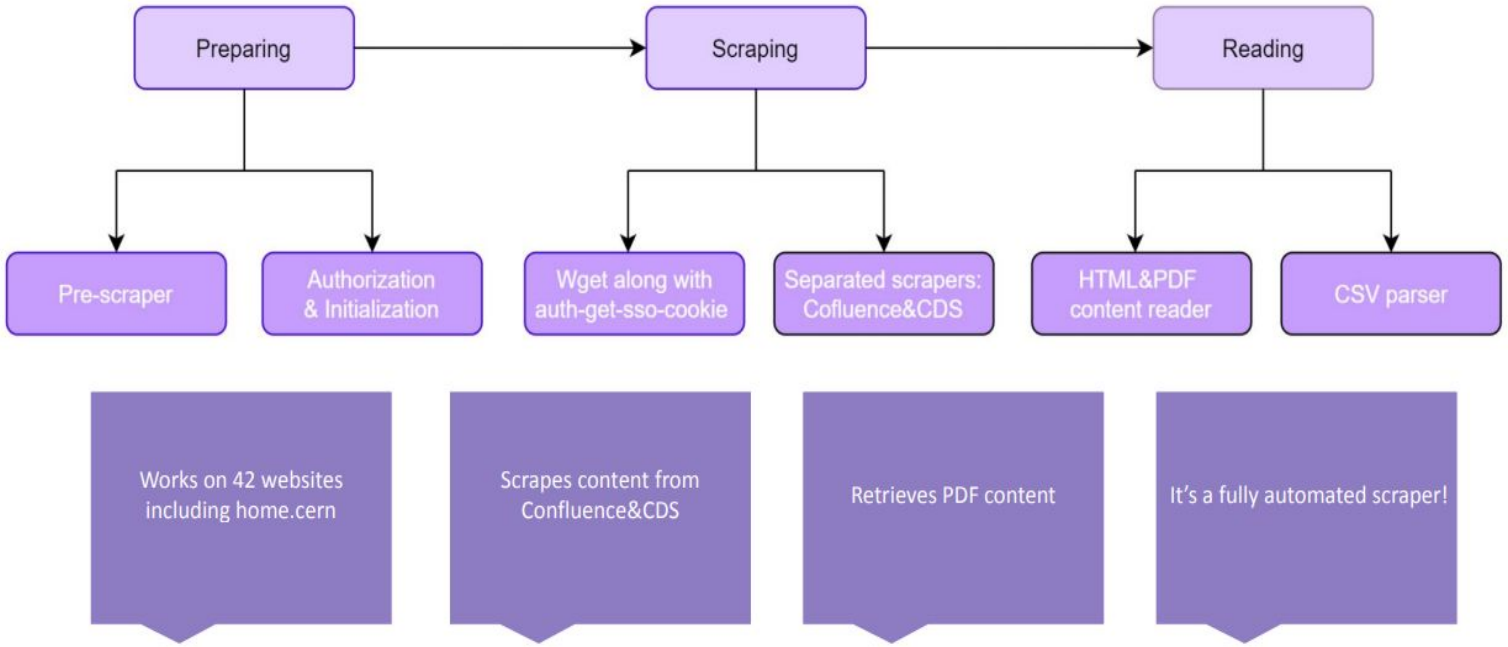
# Prototype and Architecture

- Prototype demonstrates successful knowledge retrieval using a RAG pipeline
- Key Components of RAG:
  - Indexing Pipeline
  - Retrieval Pipeline
  - LLM serving
  - User Web Interface



# Indexing Pipeline: Functions

- **Web Content Scraping**: Collects data from CERN's internal web content, including IT service documentation and parts of the public CERN website.
- **Preprocessing**: Cleans data, removes irrelevant information, and structures it in a consistent format (JSON).
- **Chunking**: Divides data into smaller, manageable units for the LLM.
- **Vectorization**: Embeds text chunks into vectors to represent their meaning.



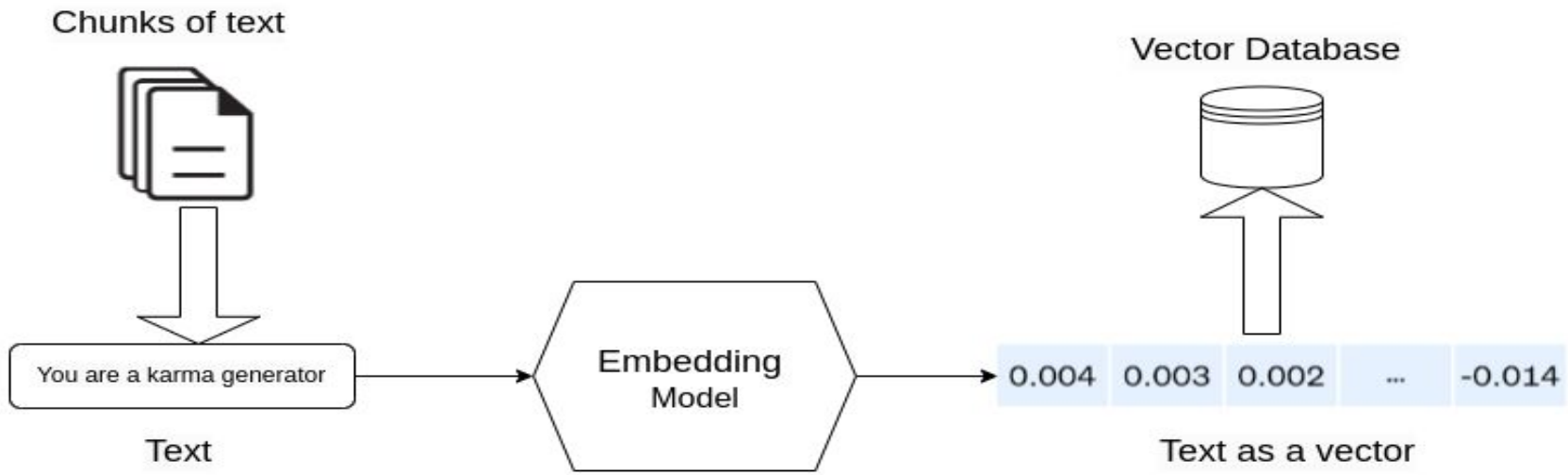
# Scraping and Sources

- Data Sources:
  - Internal web pages (HTML, PDF)
  - Confluence spaces (using a specialized scraper with Python Selenium and API interaction)
  - CERN Document Server (CDS) - scientific articles (CDS XML API)
- Future:
  - Expansion to include more internal CERN domains and data formats
  - Multimodal: Images, audio, video

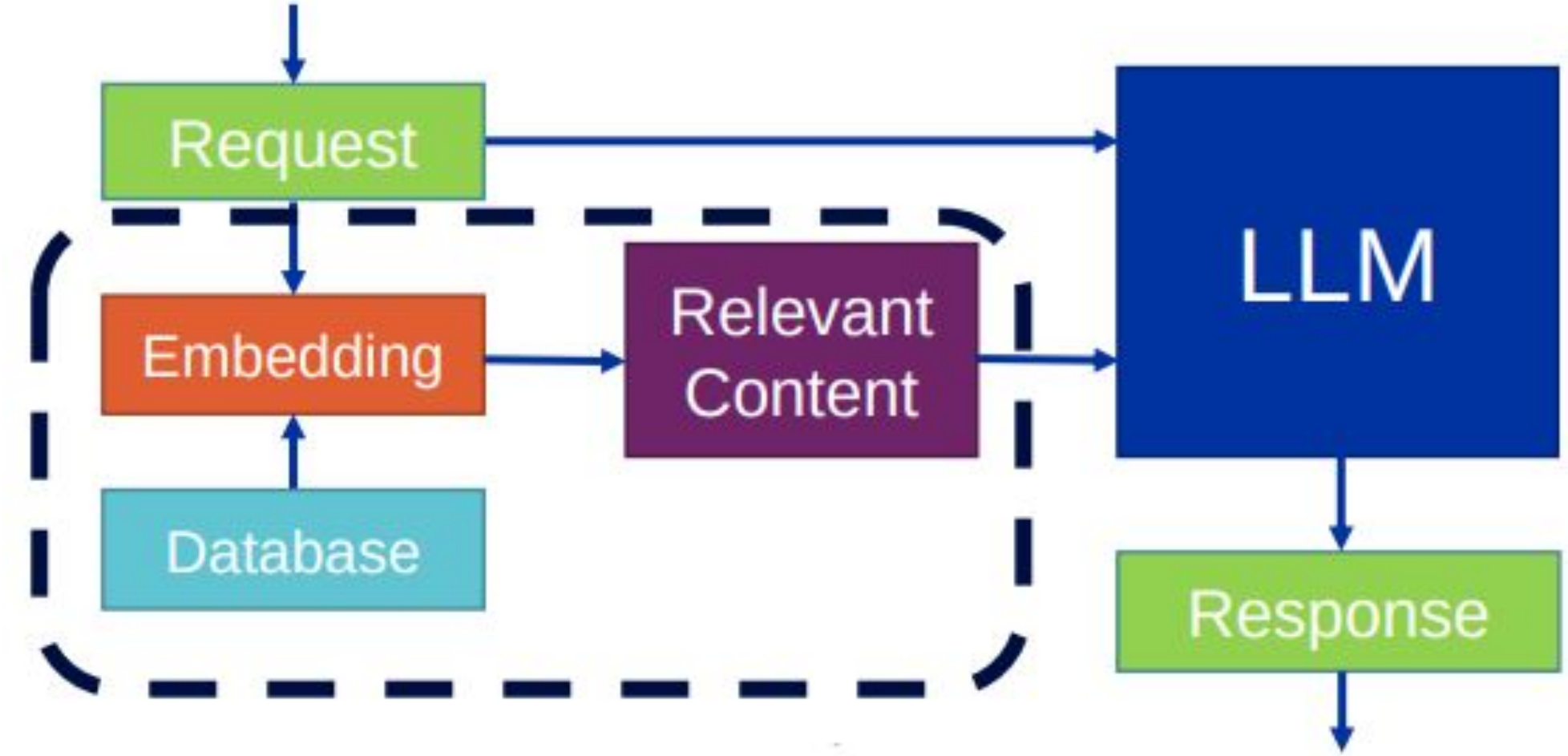
| Name of column | Content of column |
|---|---|
| Index | Individual page numbers increasing by 1 (good practise) |
| Name | Value of tag <title>, which is unique for each web page or title of PDF file |
| Text | Retrieved content for web page or PDF file |
| Url | URL address of web page or PDF file |

# Chunking and Embeddings

- Chunking is crucial due to the limited input context length of LLMs.
- Current method: Rule-based division by paragraphs or sentences, with plans to explore advanced techniques.
- Embedding model: **e5-large-v2**, chosen for its efficiency, effectiveness, and multilingual capabilities.
- Vector database: **ChromaDB**, used for storing and retrieving embeddings and associated metadata.



# Core of Retrieval Pipeline

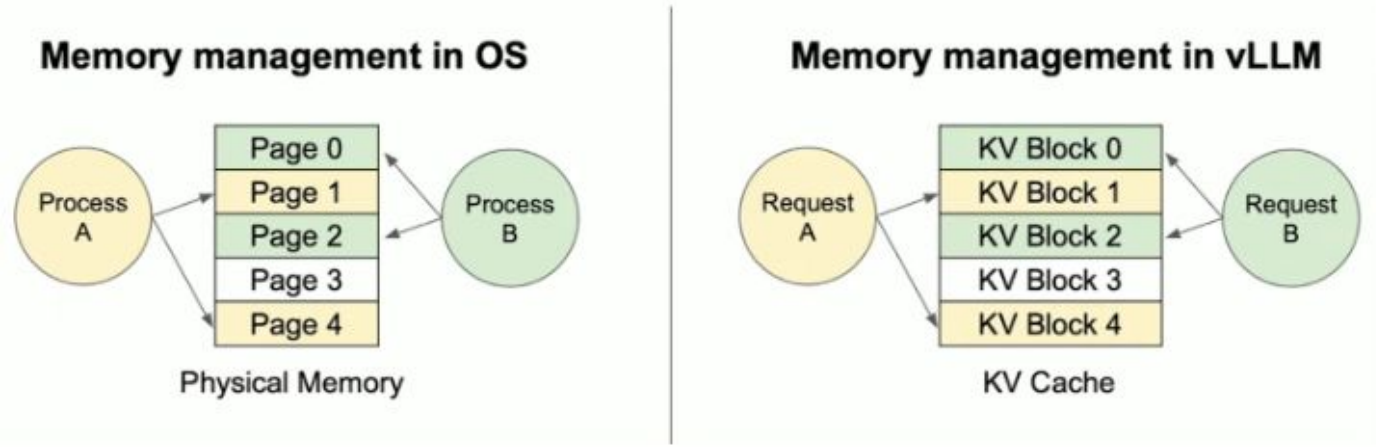


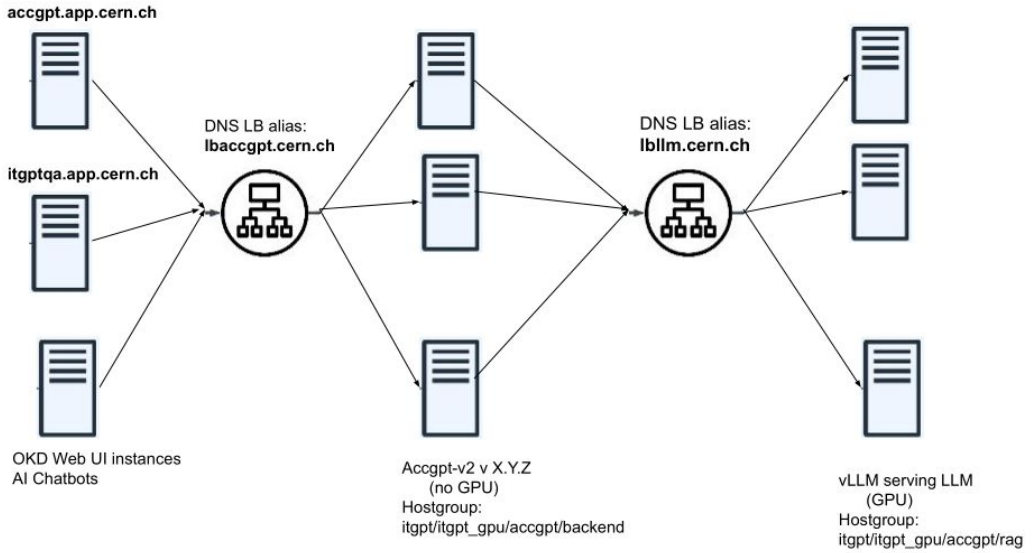
# Retrieval Pipeline

- **Purpose**: Retrieve relevant knowledge from the vector database to support the LLM in answering user questions.
- **Process**:
  - Converts the user's question into a vector using the same embedding model as used for creating the embedding database.
  - Performs a similarity search between the question vector and vectors in the database.
  - Retrieves the most relevant text chunks.
- **Enhancements for Accuracy**:
  - Additional embedding search using keywords extracted from the question by an LLM.
  - Implementation of a re-ranker model for more precise filtering and confidence scores.

# LLM Serving

- **Functionality**: Provides an API for interacting with the LLM, sending prompts and receiving responses.
- **Implementation**: vLLM library, featuring PagedAttention for efficient memory management.
- **Benefits of vLLM**:
  - Reduced memory overhead
  - High throughput and performance
  - Compatibility with Hugging Face models
  - Support for multi-modal workloads
- **Current LLM**: Meta-Llama-3.1-8B-Instruct, chosen due to limitations in available GPU resources.



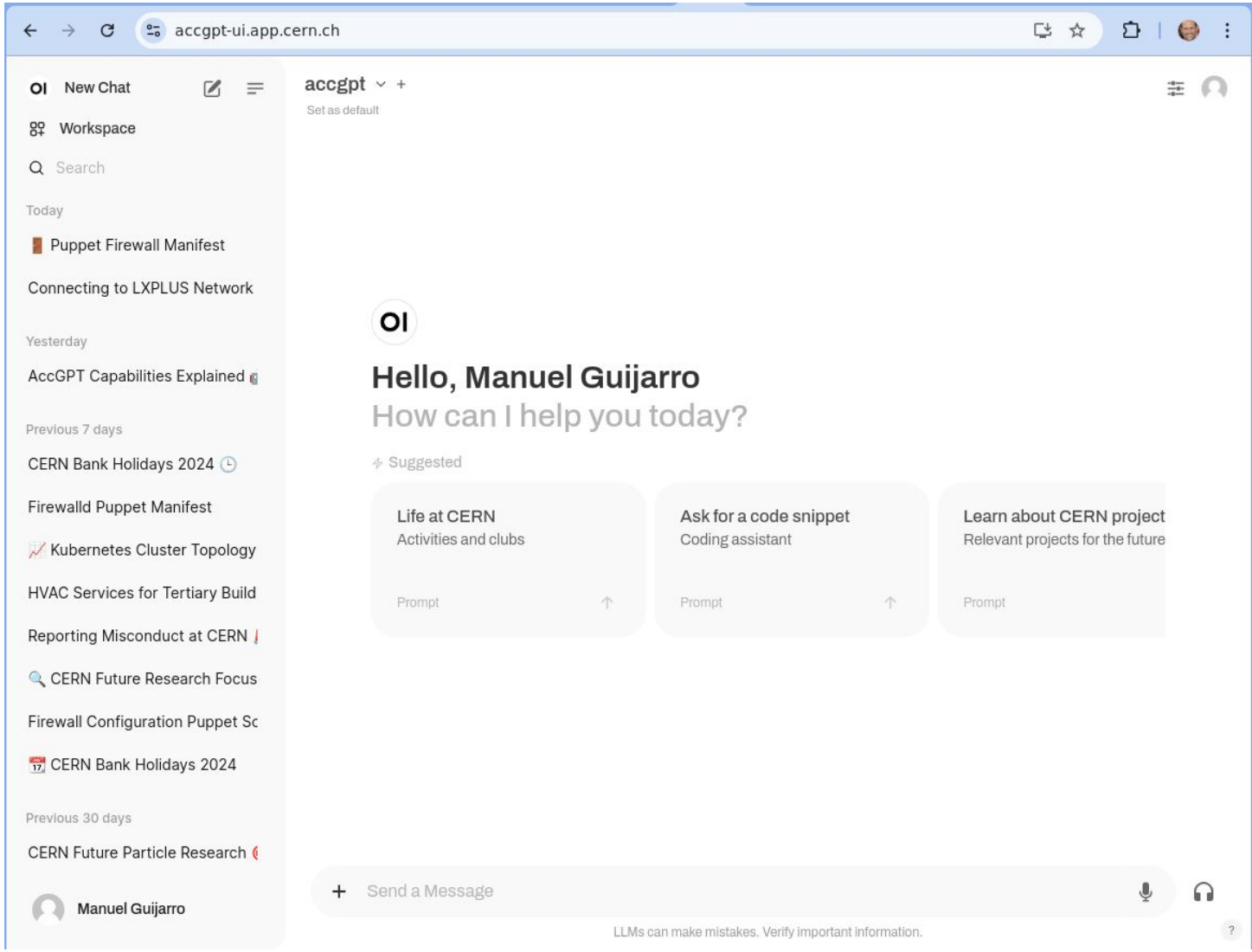



# Challenges with Commercial LLMs

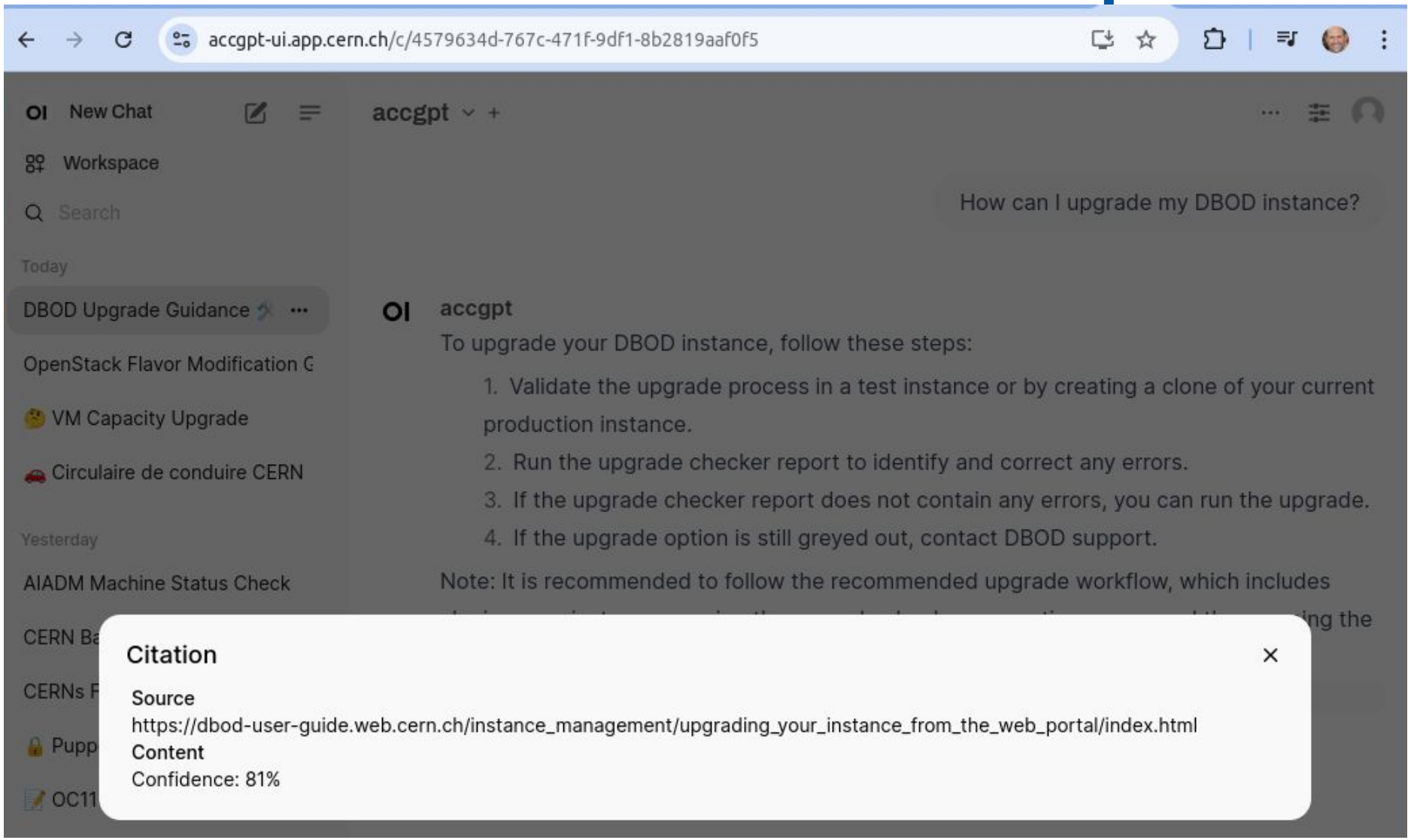
- CERN's Cloud Policy: Compliance with data privacy and security requirements poses challenges.
- User Preferences: CERN users prefer on-site, self-hosted models for continuous operation, confidentiality, and control over data.
- Fine-tuning Concerns: Reluctance to fine-tune commercial models due to ownership and exit strategy issues.
- Cost Considerations:
  - Evaluating costs, performance, and reliability is key when choosing between self-hosting LLMs or commercial providers like OpenAI.
  - For high-volume usage, self-hosted LLMs can become more cost-effective:
    - 720 questions = 20,105 input tokens + 1,170 output tokens
    - Cost: 1 hour of an A100 GPU vs $8.5 OpenAI subscription

# User Web Interface

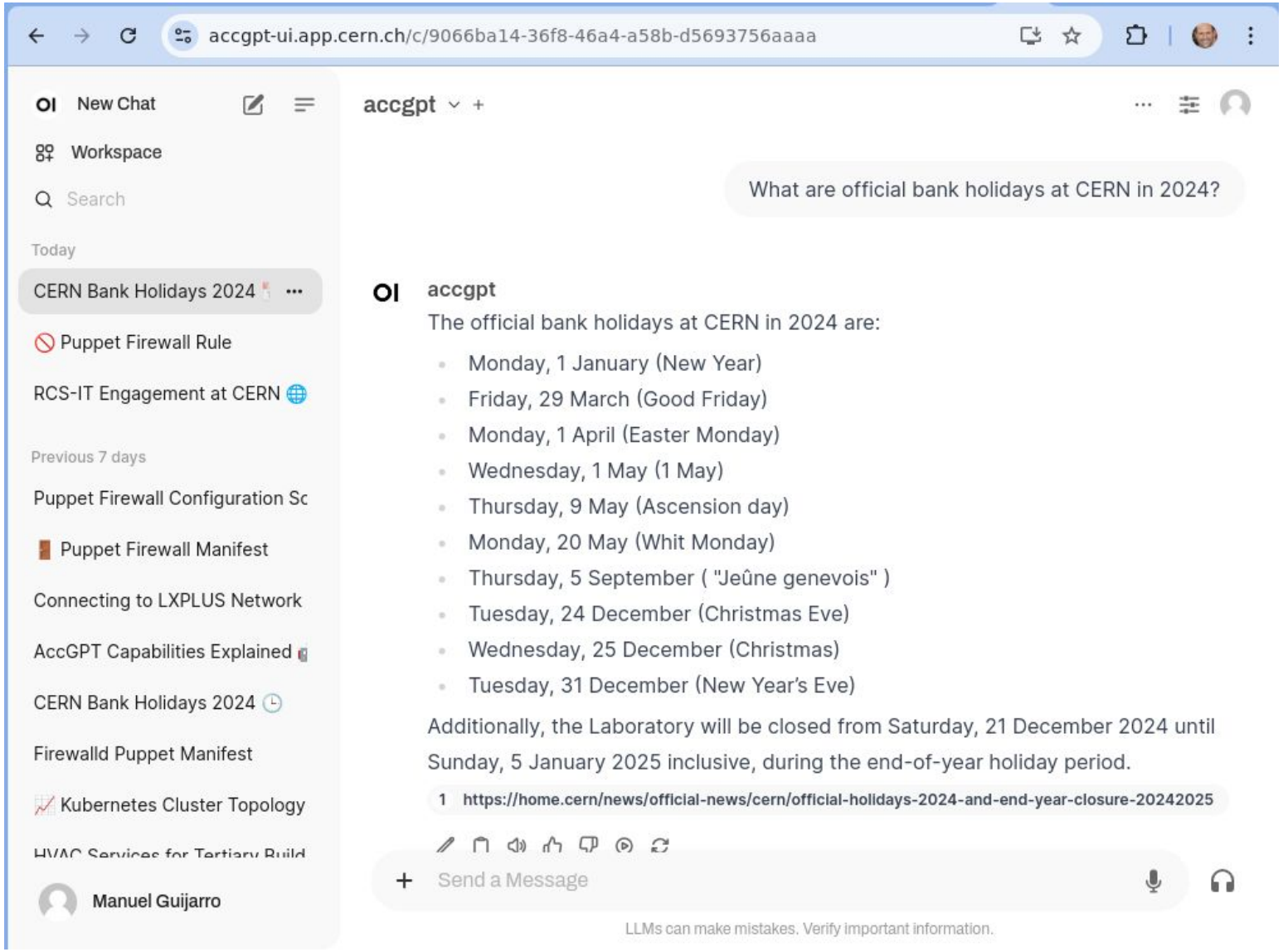
- **Platform**: OpenWebUI, an open-source solution chosen for its customizability, features, and community support.
- **Benefits of OpenWebUI**:
  - Integration with commercial and self-hosted LLMs
  - Cross-device compatibility
  - Scalability
- **Deployment**: OpenShift instance for scalability, security, and efficiency.



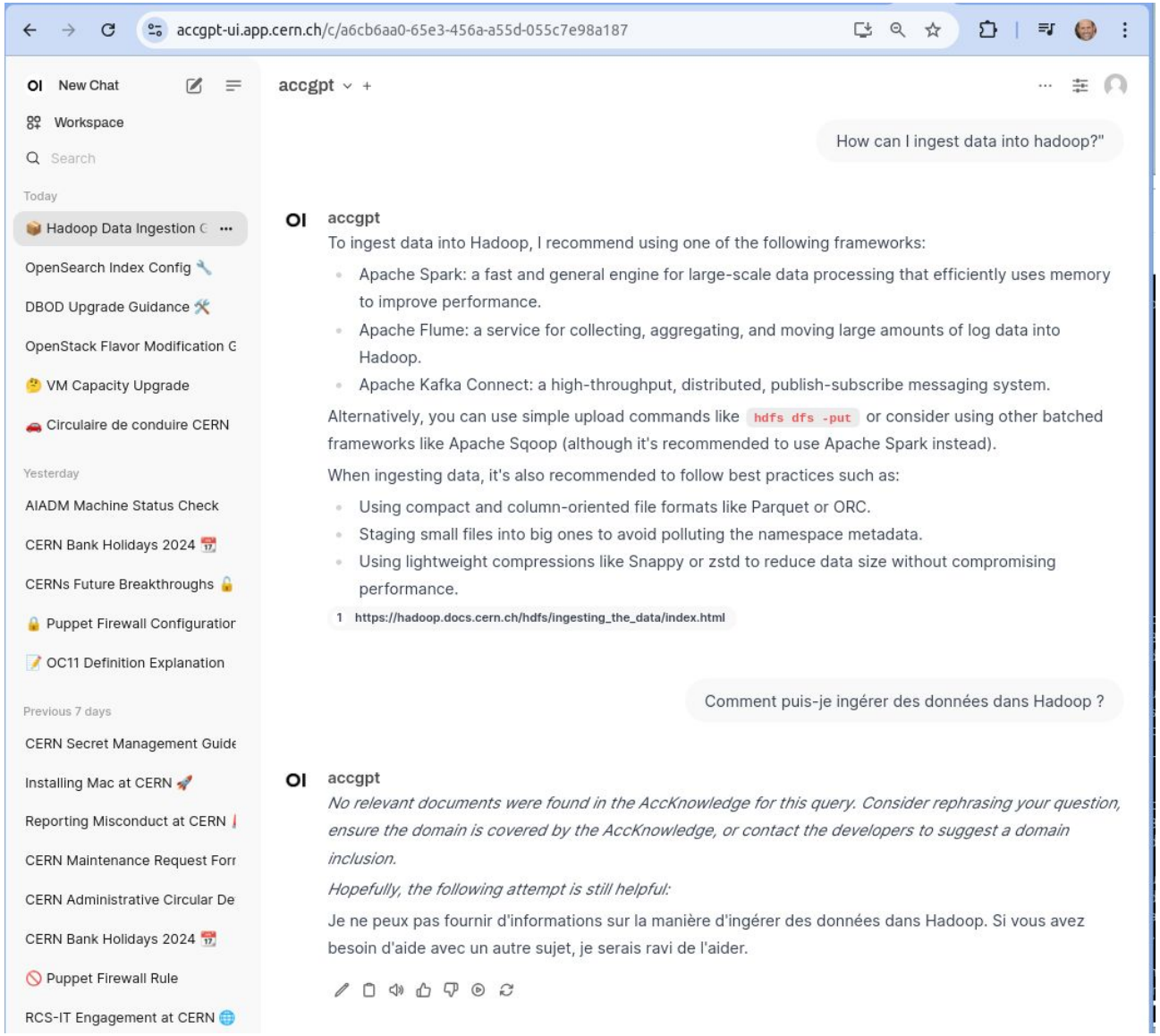
# Screenshot: DBOD example



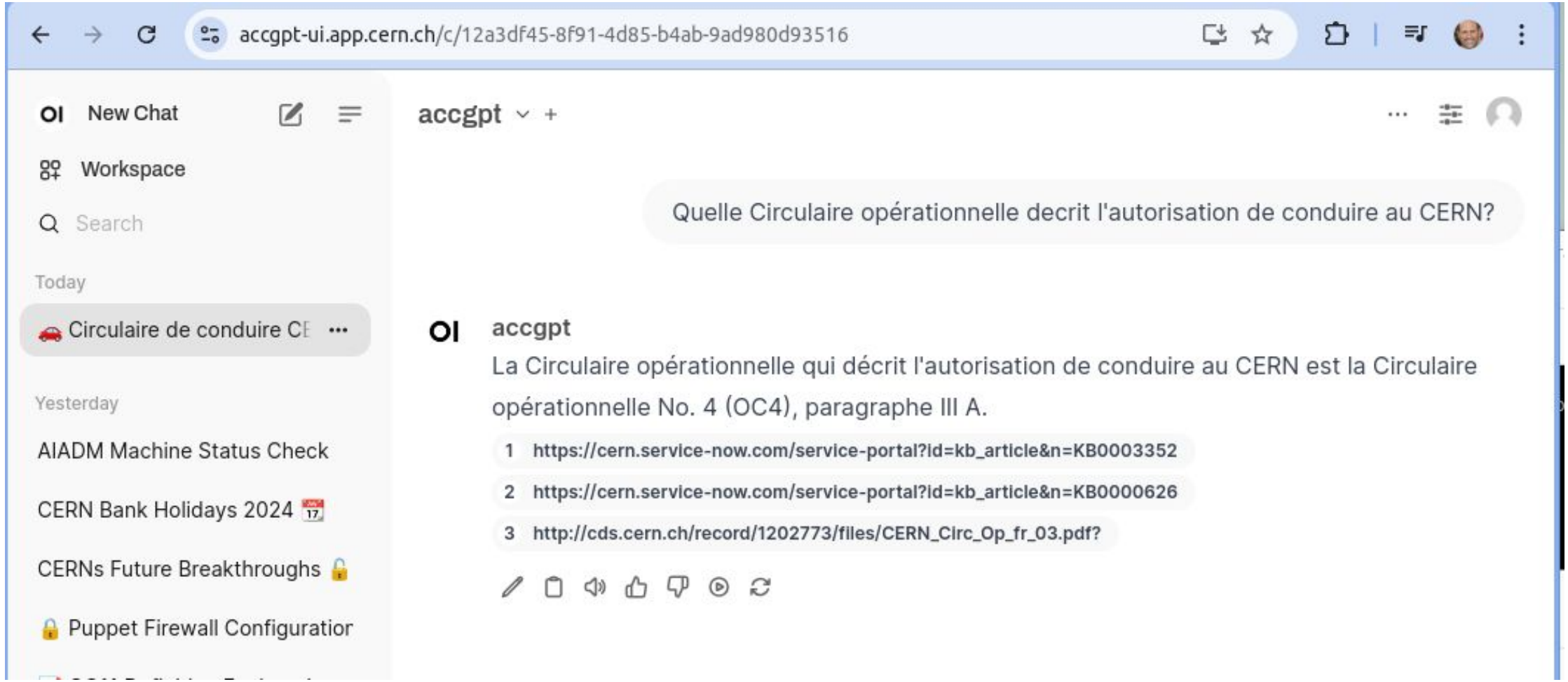
# Screenshot: Add context !!!



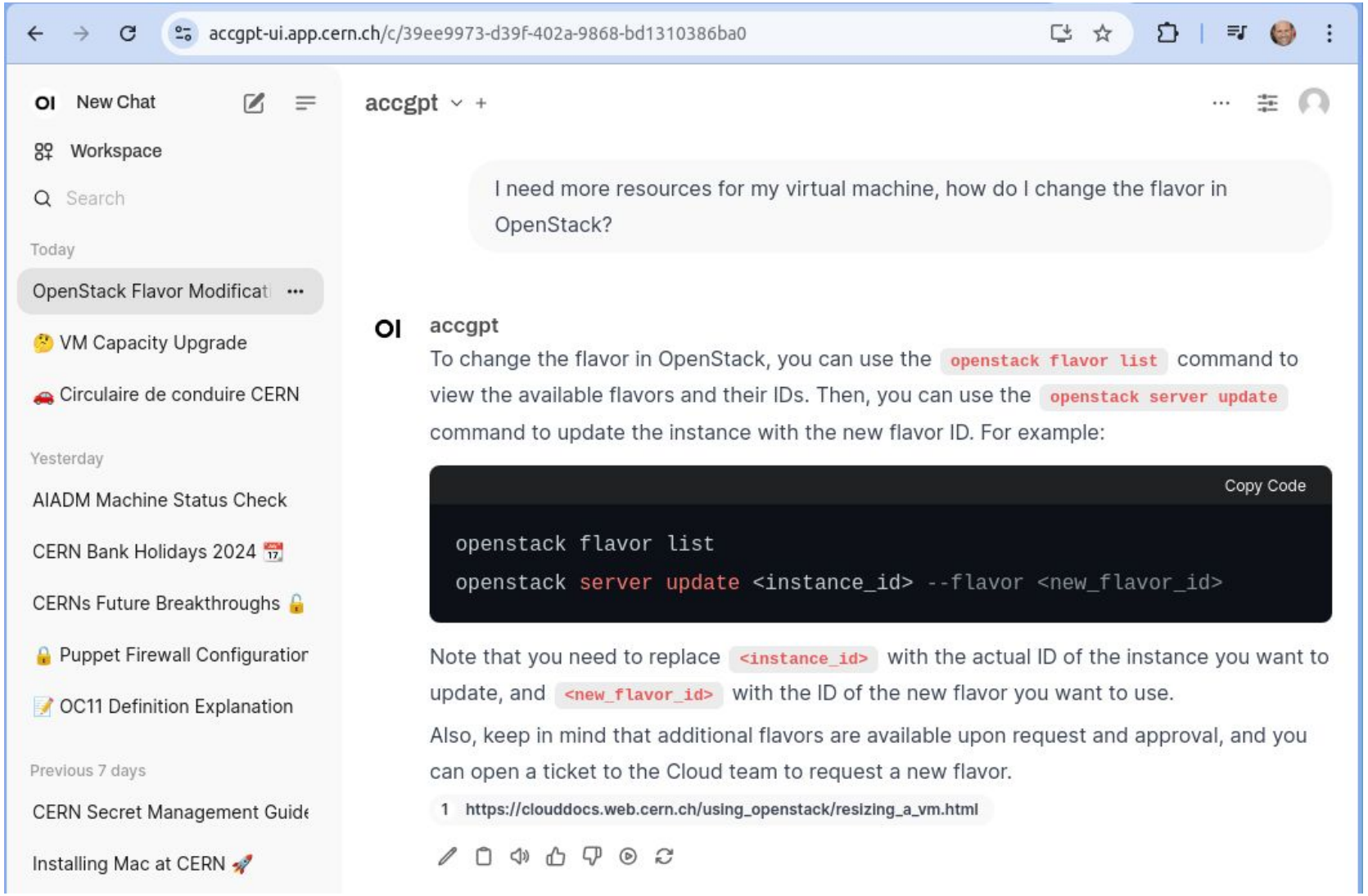
# Screenshot: No real French Support



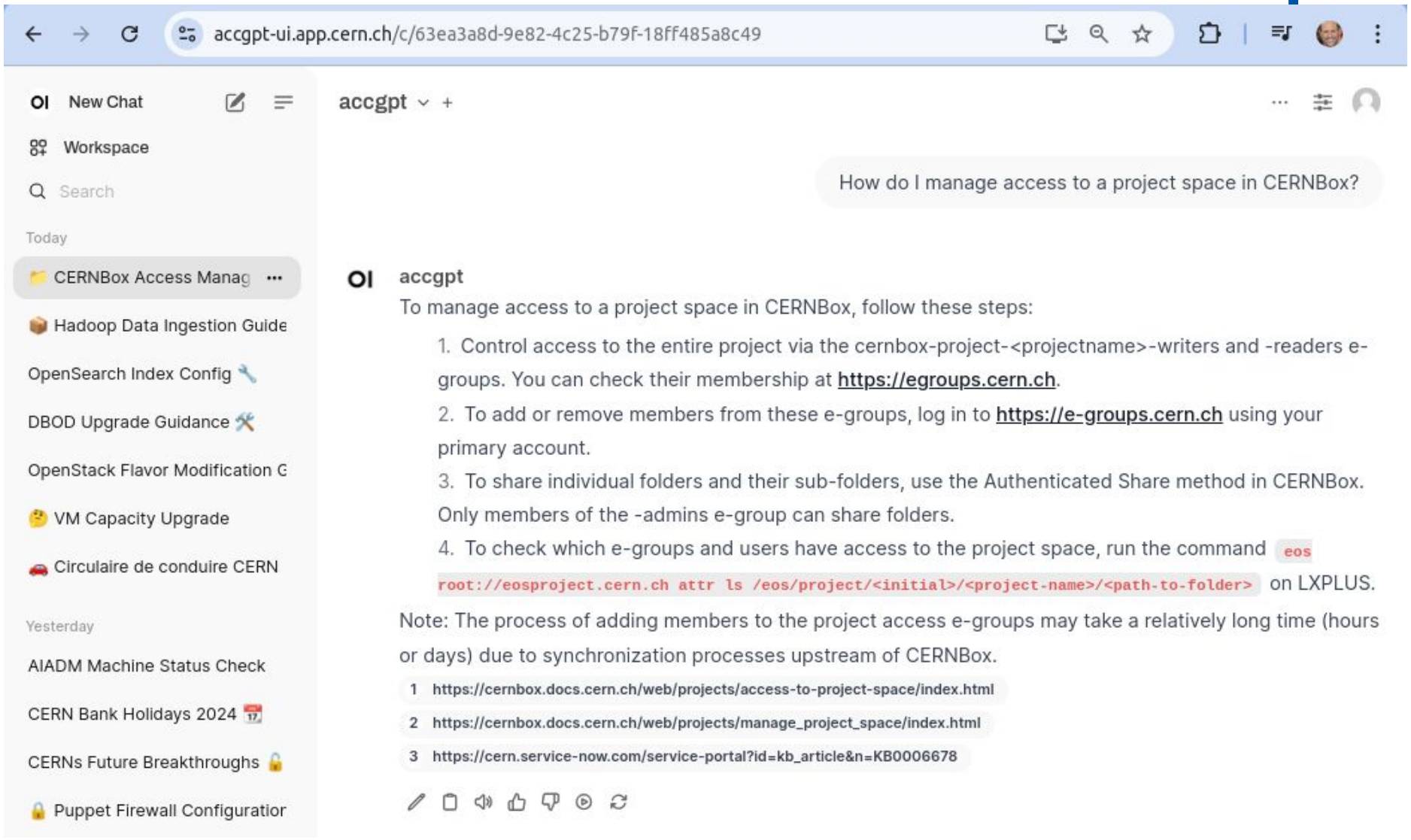
# Screenshot: …only if French doc



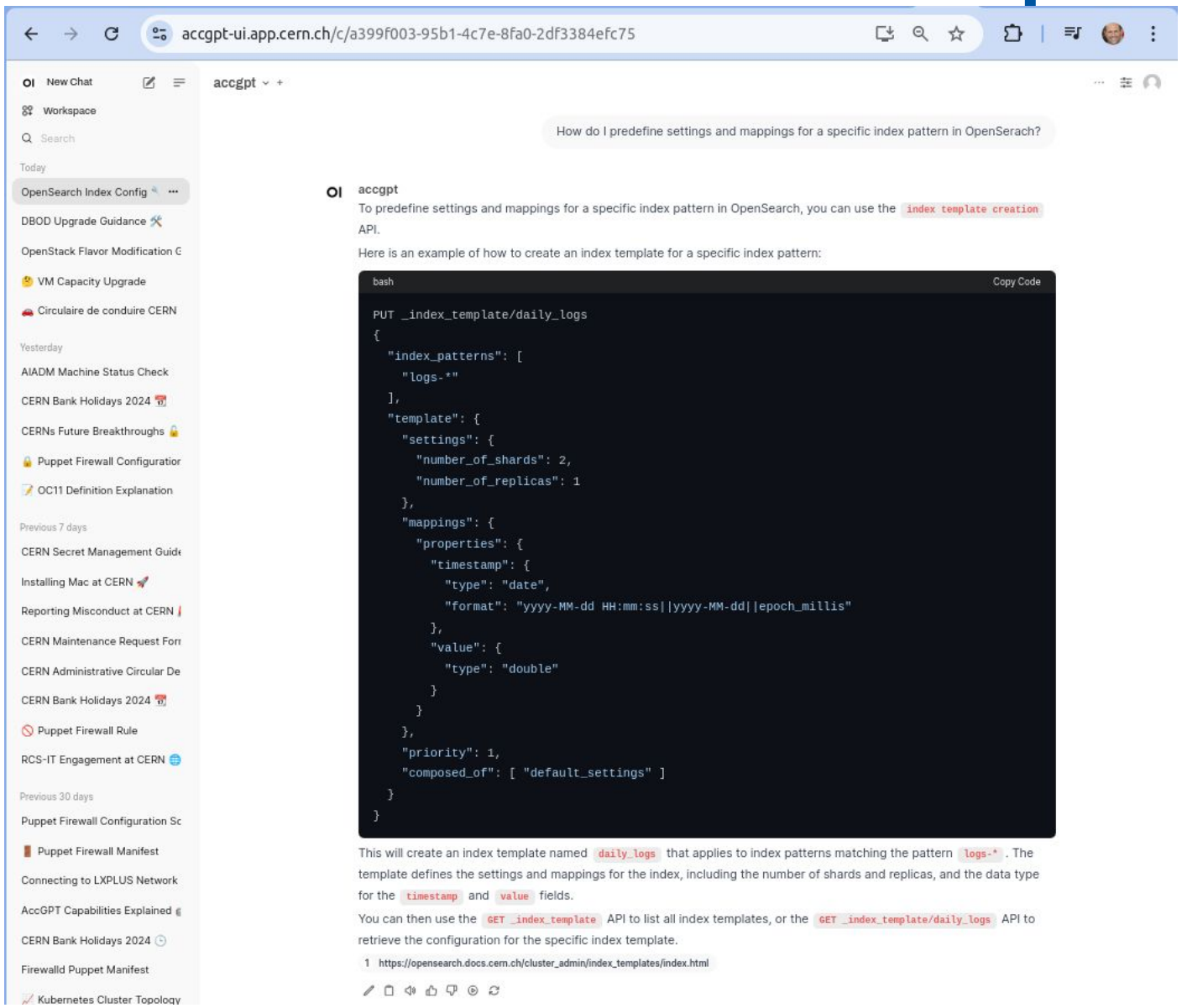
# Screenshot: OpenStack example



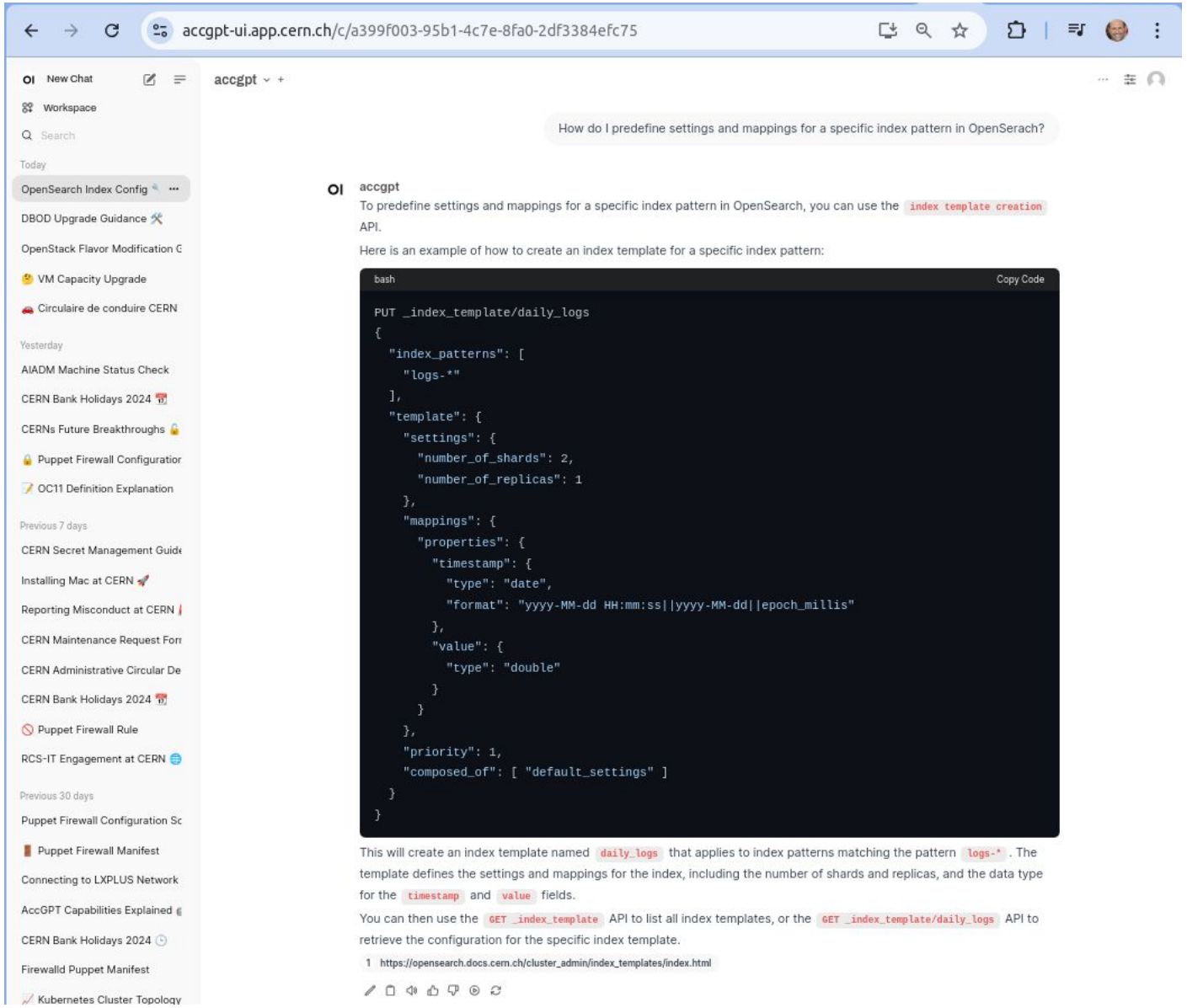
# Screenshot: CERNBox example



# Screenshot: DBOD example

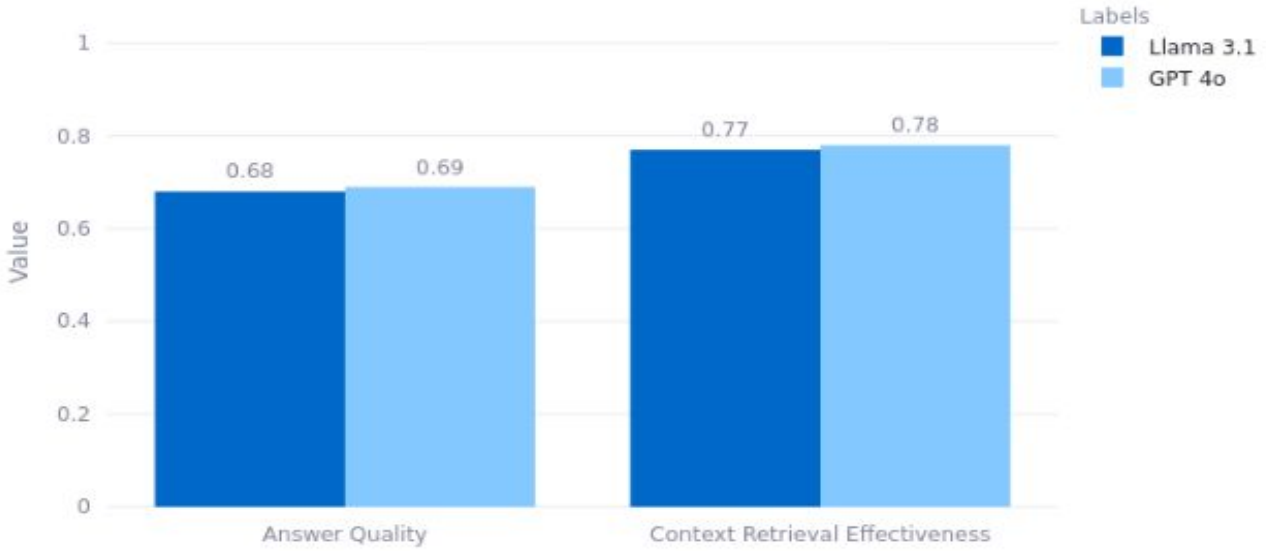


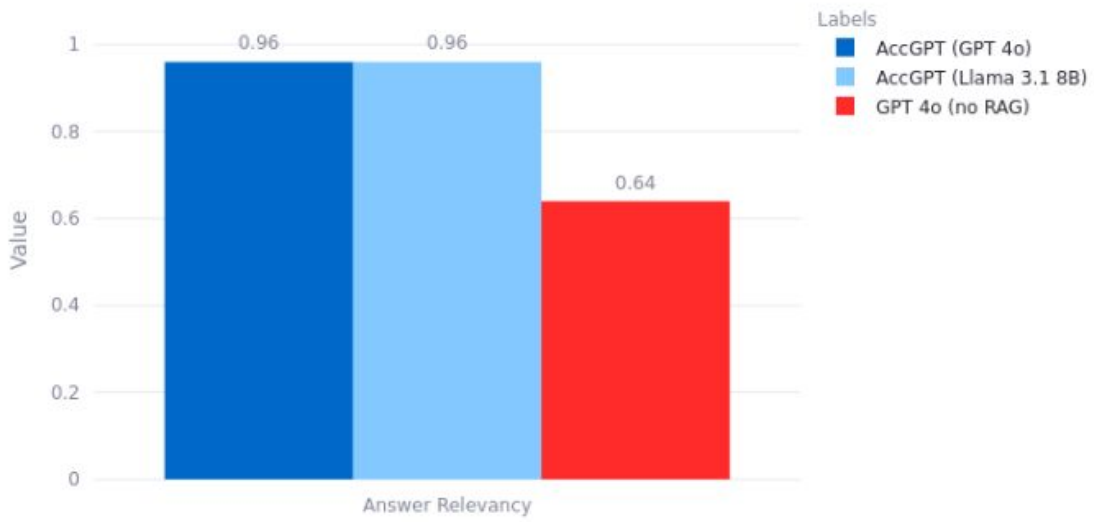
# Screenshot: OpenSearch example



# Evaluation and Performance

- **Metrics**:
  - Answer Quality (relevancy, correctness, faithfulness).
  - Context Retrieval Effectiveness (recall, precision, URL coverage, semantic similarity).
- **Evaluation Framework**: Ragas open-source framework.
- **Comparison**: AccGPT with Llama 3.1 vs. OpenAI GPT 4o (as a judge and a baseline).
- **Retrieved URL Coverage**: 92% accuracy in retrieving the correct URLs.
- **Context Retrieval Effectiveness**: 77-78%, demonstrating the accuracy of the RAG system.

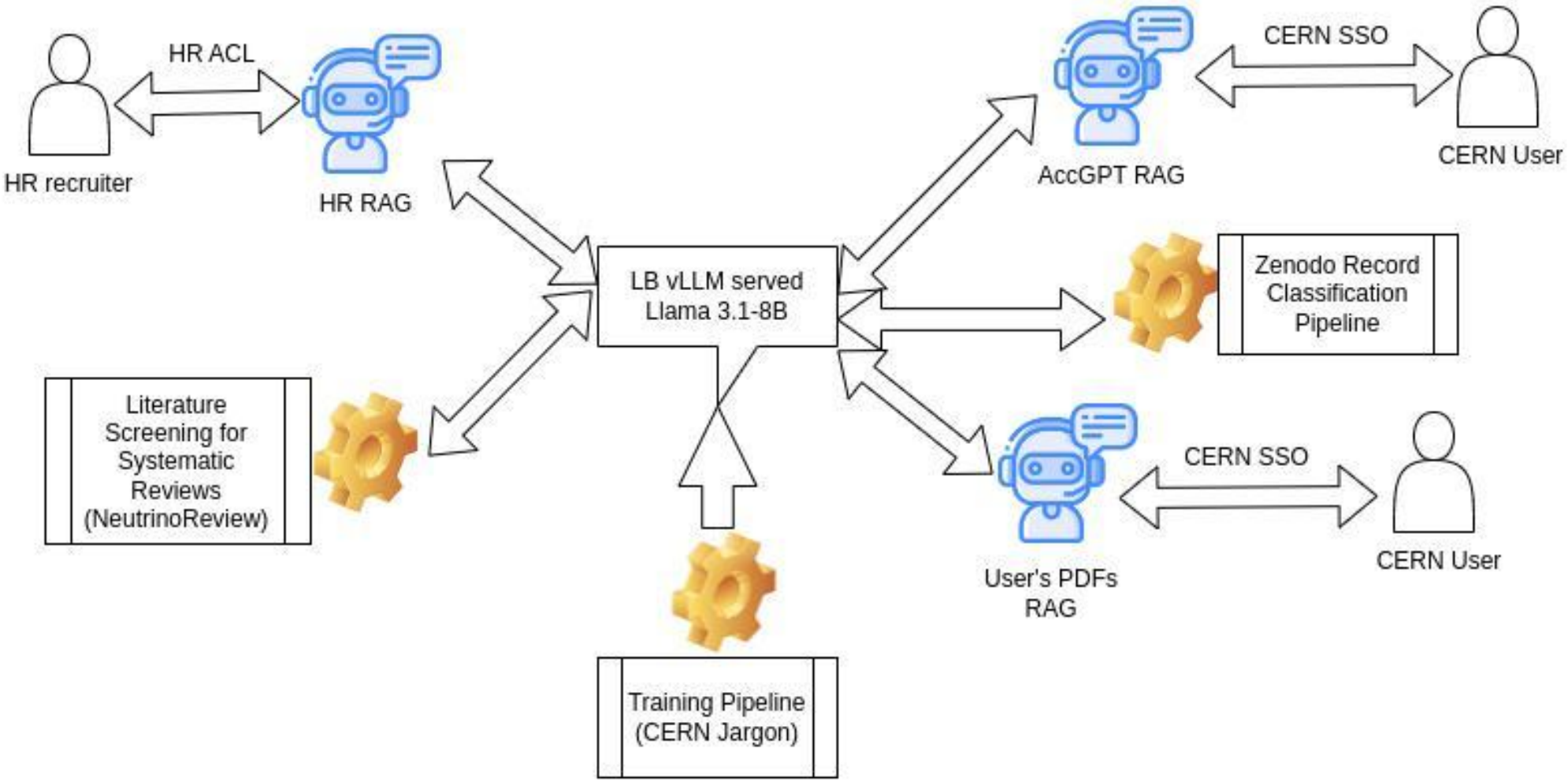


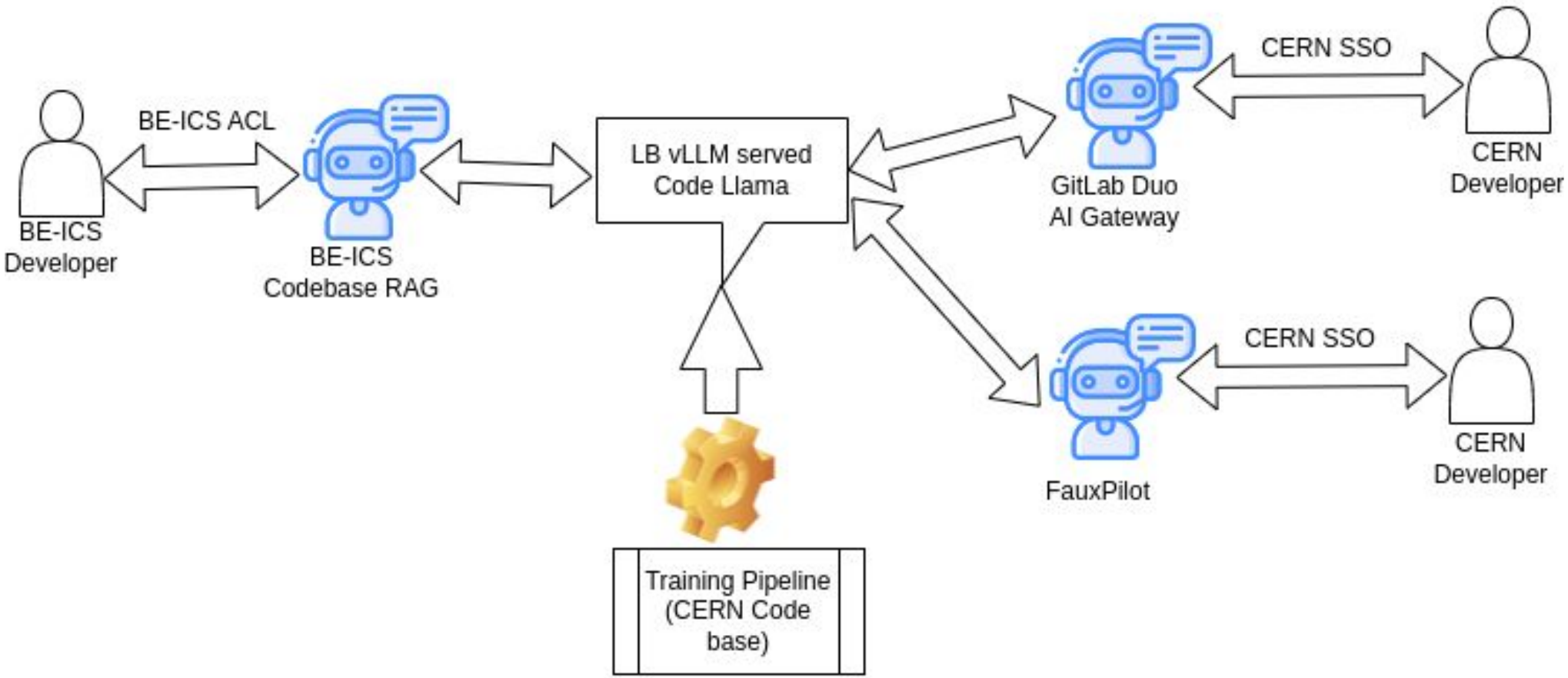
# Lessons Learned

- **Internal Web Content Quality**: The importance of well-structured and accurate internal web content for effective RAG performance.
  - Knowledge Base growth can trigger wrong answers: e.g. things documented in several places (...but reference URL provided).
- **Open Source Community:** Embracing open-source solutions (like OpenWebUI) can save time and effort
  - Facilitates enormously AI Chatbots building
  - Can be used as single access point to several LLMs
  - Can be used to build RAG pipelines. OpenSearch RAG-Tool as well
- **Access Control Lists (ACLs)**: The need for mechanisms to handle varying ACLs for different content within CERN's internal network
- **Right-sizing LLMs**: Selecting appropriate LLMs based on specific needs and optimizing their utilization to manage costs

# Information Retrieval Use cases



# Code Assistant Use cases



# Conclusion

- AccGPT demostrates the potential of **RAG-based LLM chatbots** for **efficient internal knowledge retrieval** in HEP sites and other organizations
- Documentation practices can be greatly improved, leading to increased efficiency and accuracy: It **identifies which internal documentation needs to be corrected**
- The success of LLM applications depends on:
  - A solid strategy
  - **Well-organized data**
  - Careful consideration of storage, compute options, and techniques (**moving target)**
- A Generative AI service is really needed at HEP sites:
  - To provide **access to LLMs**, and **tooling for AI application building**
  - To extend **the Service for Machine Learning lifecycle**
    - To also be usable for LLMs
    - To be based on OpenSource solutions (eg. https://mlflow.org/)
  - To provide a central **ML model repository**
  - To study the benefits and options for:
    - **Fine tuning LLMs on static knowledge**
    - **Per-Purpose AI Chatbots** rather than trying to compete with ChatGPT