**🧠 EUROGATE Hackathon Challenge: “Build the smartest AI agent!”**

**Task:**

Develops an intelligent AI agent that is able to answer complex user queries based on a Retrieval-Augmented Generation (RAG) approach and Large Language Models (LLMs) – based on provided documents (e.g. B. PDFs, texts, technical documentation). Various documents are provided for the following use cases - select one of the use cases:

* The **Infobox** is Eurogate's internal employee portal. It provides general information, news, and general documents.
* **driveMybox.io** is a digital platform for booking and executing truck container transports. In addition to general information (e.g., company presentations) and FAQs, it provides database extracts and external order documents that can be used for further information on the status of the transport or to request a price quote via the Quotation API.
* **Technical Services** provides engineering services and is responsible for facility maintenance. It provides articles from the internal wiki, training materials, software manuals, and facility documents.

**🎯 Goal:**

The AI agent should understand questions, extract relevant information from data sources and provide precise, context-related answers to support users in their daily work depending on the use case.

**🛠️ What your agent should be able to do:**

* Process questions (text or speech)
* Generate funded answers using an LLM (suitable for the respective use case)
* Identify sources or references

**🔧 Technology Notes:**

Frameworks and LLMs should be used, e.g., LlamaIndex ( <https://www.llamaindex.ai>) and OpenAI (an API key can be provided for this). Python is preferred as the programming language.

**💡 How you implement this is up to you:**

Whether via a chatbot interface, a web app, a CLI tool, or something completely different – the main thing is that the agent is intuitive to use and provides understandable answers.

**🌟 Bonus points are available for:**

* An elegant, intuitive user interface
* Complex requests (e.g. order processing as PDF)
* Multimodal responses (e.g. B. combined text, table or visualization answers)
* Transparent reasons for the answer (e.g. B. by citing sources or highlighting in the document)
* Speech input and speech output (speech-to-text + text-to-speech)

**🔩 Materials**

Depending on the use case, the following materials are available.

* FAQs
* excerpts
* API descriptions
* Presentations
* Order documents