LLM Hallucination

- LLM Large Language Models are susceptible to ‘hallucination’ where they generate incorrect output data, for example a router configuration with errors
- General LLM models are very good at generating natural text, but they can fall short when technical knowledge of a topic or up to date information is required
- You may be able to include data such as a running configuration as part of an input prompt, but this is not very scalable

Ways to Reduce Hallucination

- **Build a new model from scratch:** Make an LLM with a relevant dataset. This is very expensive due to the expertise, time and compute resources required
- **Fine tuning:** Load additional relevant data to an existing LLM and use techniques such as backpropagation to tune it. This is not as expensive as building a new model from scratch but is still very costly and time consuming and carries the risk of the model forgetting previously learned information
- Both options are impractical for keeping the data up-to-date

Retrieval-Augmented Generation (RAG)

- Retrieval-Augmented Generation (RAG) enhances the accuracy and currentness of an existing LLM by looking up an external database.
- It can be used by nearly any LLM.
- It is relatively easy to implement with public tools and knowledge bases available.
- Internal knowledge bases can also be used which remain private.

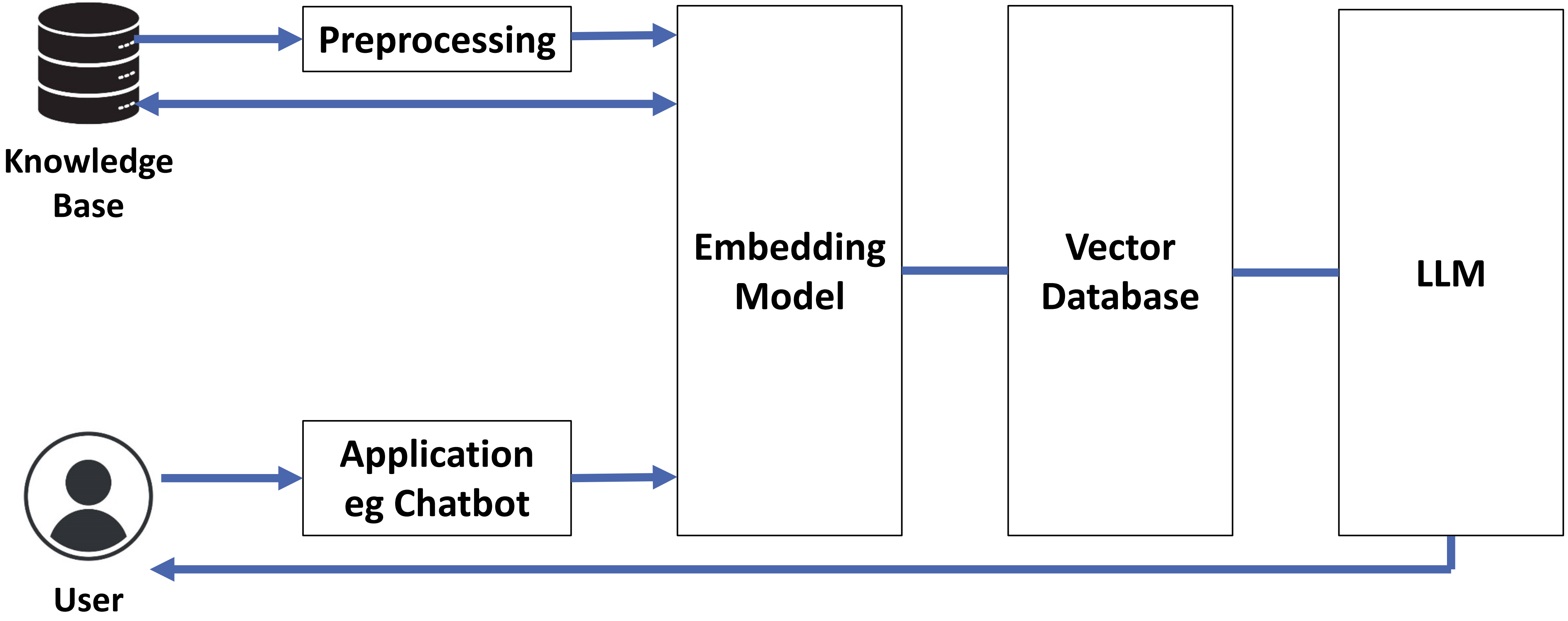
How RAG Works – Creating the Database

- During preprocessing the knowledge base is split into tokens and chunks and converted to a machine readable numeric ‘vector’ format.
- An embedding model creates a vector database optimized for search and retrieval.
- In the background the embedding model continuously updates the vector database as the knowledge base is updated.

How RAG Works – User Queries

- When users enter a query the Embedding Model converts it into numeric format which is compared to the vector database.
- Matches are retrieved and sent to the LLM.
- The LLM combines the retrieved entries with its own response to create the output for the user.

How RAG Works (Cont.)



Preprocessing

Application eg Chatbot

Embedding Model

Vector Database

LLM

RAG usage in Network Operations

- Configuration generation
- Troubleshooting: Knowledge base can contain relevant articles, also previous incident reports and their solutions
- Up-to-date documentation generation
- Predictive maintenance: Previous maintenance schedules, anomalies, environmental data