Implementing LLM with RAG in Elasticsearch

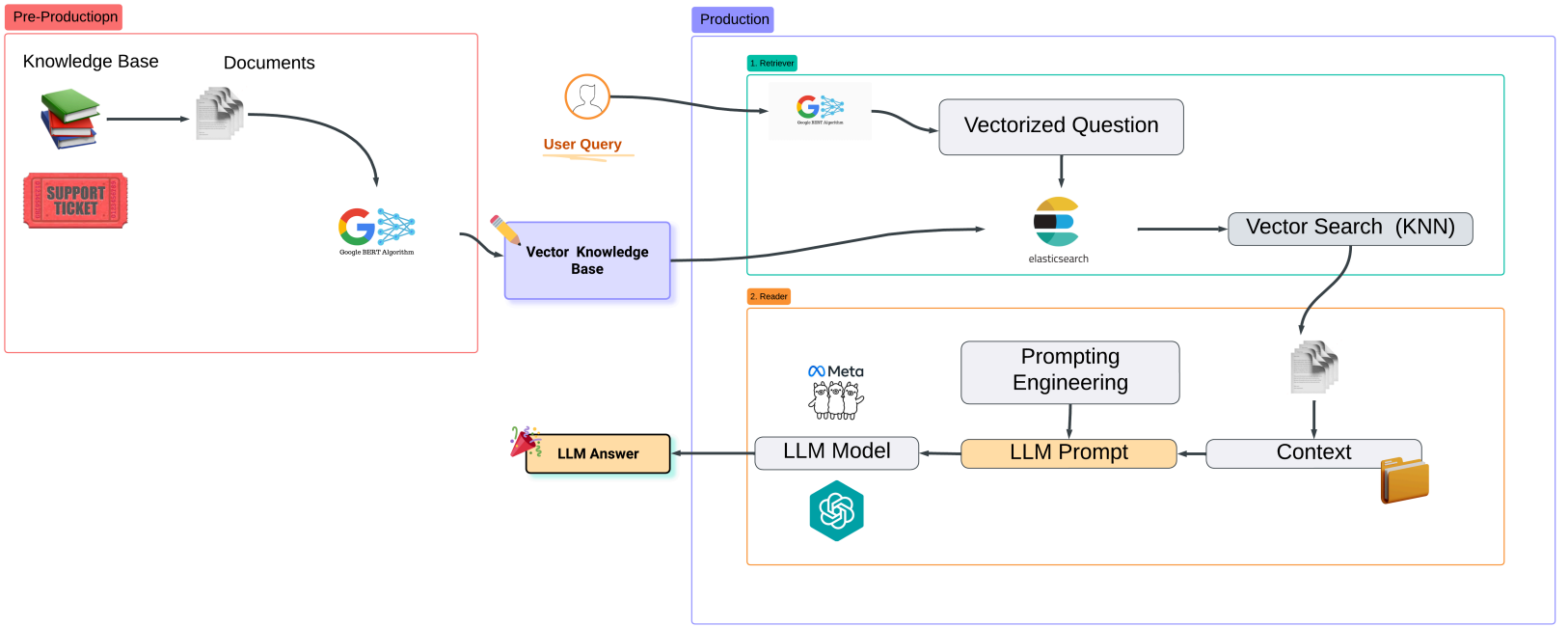
Proof of Concept Prototype for Odoo Ticket Querying

Managed Services

What is RAG (Retrieval-Augmented Generation)?

- **Combination of Retrieval and Generation**:
  - **Retrieval**: Searches for relevant information from a dataset.
  - **Generation**: Produces coherent responses using retrieved information.
- **Components**:
  - **Retriever**: Finds relevant documents.
  - **Generator**: Uses documents to generate a response.

RAG Proof of Concept



What are the Use Cases for RAG?

There are many different use cases for RAG. The most common ones are:

- **Question and Answer Chatbots**: Automate customer support and website lead follow-up to answer questions and resolve issues quickly.
- **Search Augmentation**: Search engines that augment search results with LLM-generated answers can better answer informational queries.
- **Knowledge Engine**: Ask questions on your data (e.g., HR, compliance documents, tickets). Company data can be used as context for LLMs and allow employees to get answers.

Why LLMs and RAG are an Opportunity

- **Leverage Historical Data**:
  - Utilize existing ticket data for context-aware, accurate responses.
- **Boost Efficiency**:
  - Resolve tickets faster with quick access to relevant information.
- **Proactive Issue Resolution**:
  - Identify and address recurring problems.
- **Improve Reporting**:
  - Generate detailed, insightful reports on ticket status and performance.

LLMs and RAG are not Magic

- **Prediction-Based**:
  - LLMs predict the next word based on context, requiring substantial computational power.
- **Data-Driven**:
  - Performance hinges on the quality and volume of training data.

Investment Requirements

- **Infrastructure**:
  - To achieve real-time responses, we may need to invest in robust computing infrastructure or cloud services.
- **Token Management**:
  - Efficient token handling is crucial for managing costs and performance.

Proof of Concept Prototype for Odoo Ticket Querying

Proof of Concept: LLM with RAG in Elasticsearch

- **Objective**: Enhance ticket querying in Odoo using LLM with RAG.
- **Components**:
  - **Elasticsearch**: Store and retrieve ticket data.
  - **Ollama Models**: Generate responses based on retrieved data.
- **Scope**: Implement a prototype to query ticket information efficiently.

Why Ollama Models?

- **Privacy**: Runs LLMs locally, ensuring data privacy.
- **Control**: Full control over deployment and model usage.
- **Customization**: Easily customizable to fit specific business needs.

Conclusion and Next Steps

- **Conclusion**: Implementing LLM with RAG in Elasticsearch offers significant benefits.
- **Next Steps**:
  - Develop and test using better data and better prompting techniques.
  - Gather feedback and refine the solution.
  - Plan for real implementation.