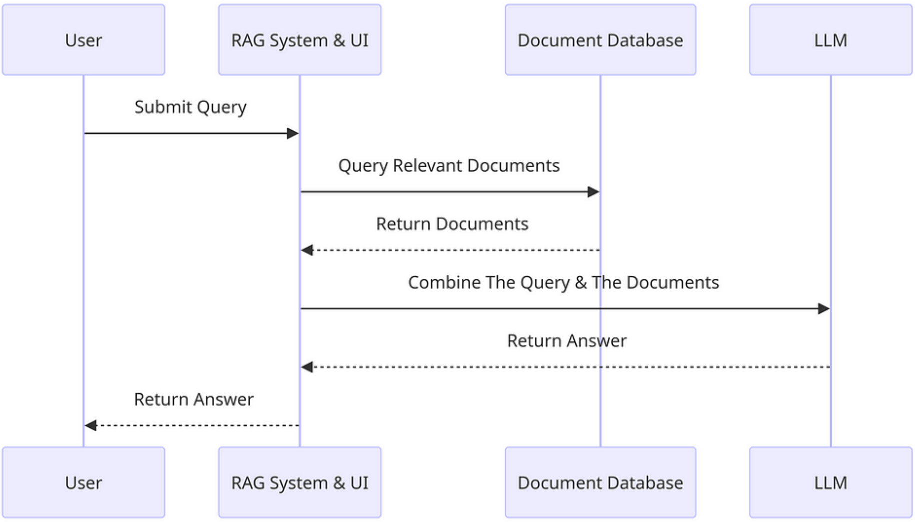
Retrieval Augmented Generation

1.1 Steps of a RAG System

1. **User Submits Query:** User inputs a query.
2. **RAG System Queries Relevant Documents:** RAG searches for relevant documents.
3. **Document Database Returns Documents:** Database returns the documents.
4. **Combine the Query & Documents:** RAG combines documents with the original query.
5. **LLM Returns Answer:** Combined query and documents are sent to an LLM.
6. **RAG System Returns Answer to User:** Answer is returned to the user.

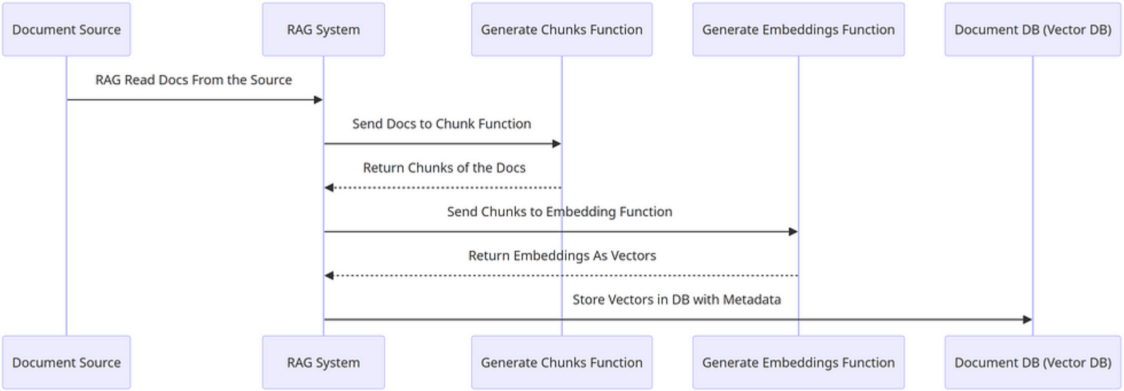


1.2 How to Query Documents

Inserting to DB:

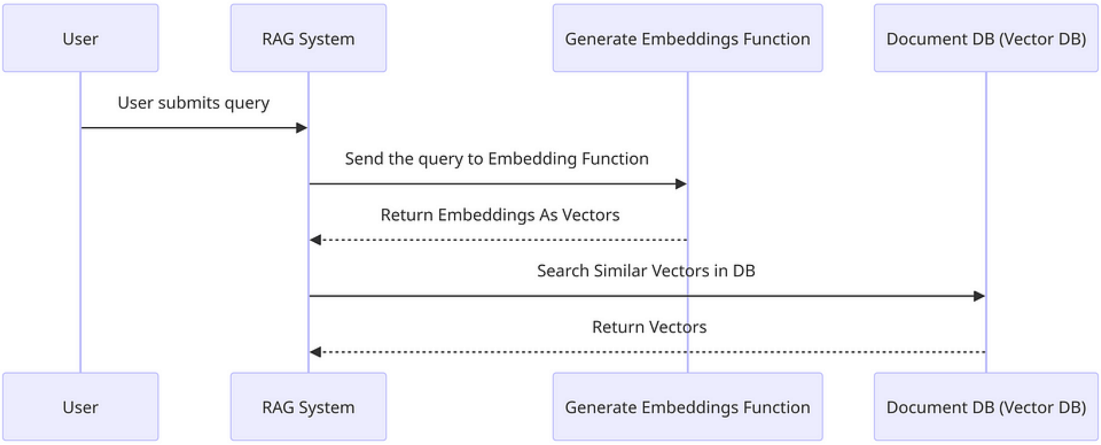
1. **Read Docs from Source:** RAG reads documents.
2. **Chunk Function:** Documents are broken into smaller chunks.
3. **Return Chunks:** Chunk function returns smaller chunks.
4. **Embedding Function:** Chunks are converted into embeddings (vectors).

5. **Store Vectors in DB with Metadata:** Vectors and metadata are stored in a vector database.


Document Source
RAG System
Generate Chunks Function
Generate Embeddings Function
Document DB (Vector DB)
RAG Read Docs From the Source
Send Docs to Chunk Function
Return Chunks of the Docs
Send Chunks to Embedding Function
Return Embeddings As Vectors
Store Vectors in DB with Metadata
Document Source
RAG System
Generate Chunks Function
Generate Embeddings Function
Document DB (Vector DB)

Retrieving from DB:

1. **User Submits Query:** User submits a query.
2. **Embedding Function:** Query is converted into an embedding vector.
3. **Search Similar Vectors in DB:** Query vector is used to search for similar vectors.
4. **DB Returns Vectors:** Database returns the most similar vectors.


User
RAG System
Generate Embeddings Function
Document DB (Vector DB)
User submits query
Send the query to Embedding Function
Return Embeddings As Vectors
Search Similar Vectors in DB
Return Vectors
User
RAG System
Generate Embeddings Function
Document DB (Vector DB)

Requirements

1. **AWS Services:**
   - S3: store document data.
   - SageMaker: Jupyter Notebooks and model deployment.
   - DynamoDB/Elasticsearch: document storage and retrieval.
   - IAM Roles: permissions and security.
2. **Claude LLM Access:** Claude API.
3. **Python Libraries:** boto3, requests.

Workflow

1. **Data Preparation:**
   - Store documents in S3.
2. **Document Ingestion:**
   - Use SageMaker notebooks to process documents.
   - Chunk documents and generate embedding.
   - Store embeddings and metadata in DynamoDB.
3. **Query Handling:**
   - User submits a query via SageMaker notebook interface.
   - Convert query to an embedding.
   - Retrieve relevant document embeddings from DynamoDB.
   - Combine query and documents for LLM input.
4. **Claude LLM Interaction:**
   - Send the combined input to Claude API.
   - Receive the generated response.
5. **Response Delivery:**
   - Display the response in the SageMaker notebook.