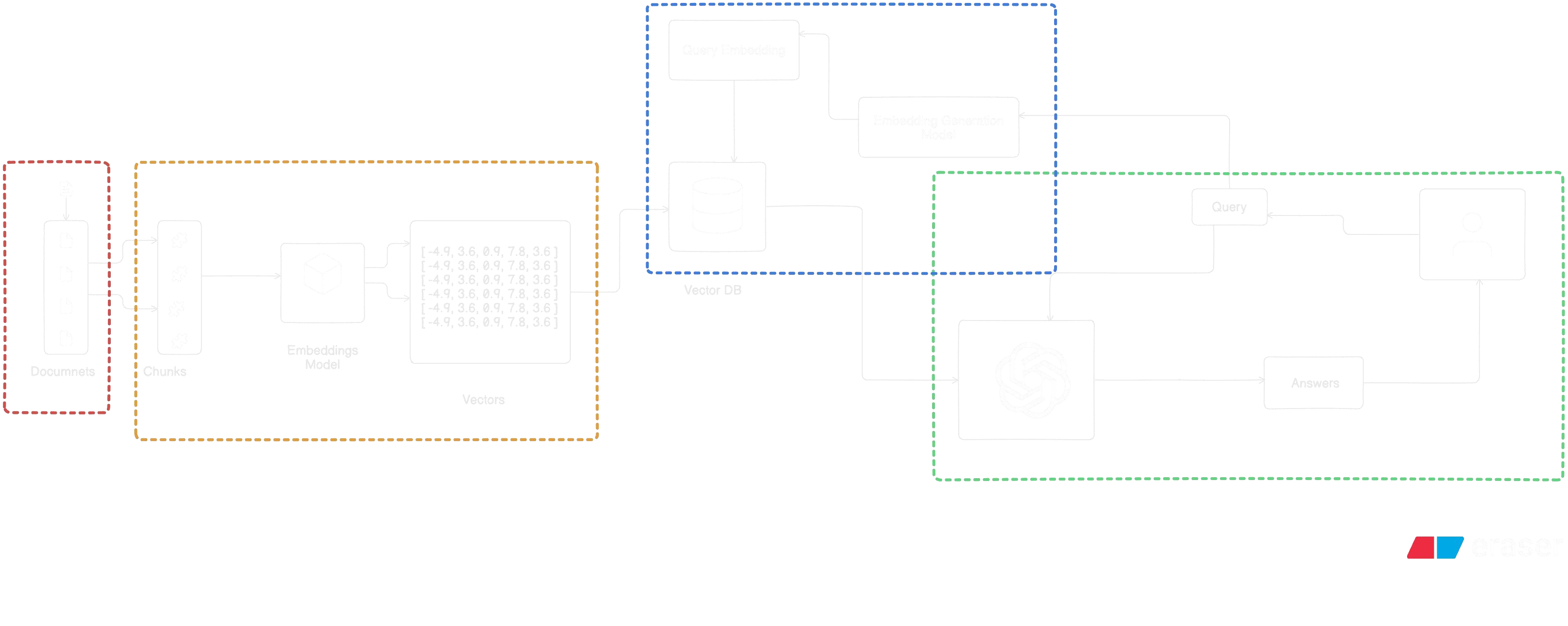
**Retrieval Augmented Generation**



***RAG Working Diagram***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User Uploads**  **PDF** | **PDFs Stored Permanently** | **Vector**  **Embeddings**  **Generated** | **User Submits Query** | **Chunks Passed to LLM** | **Answer Displayed in Chat UI** |
| User initiates the process by  uploading PDF files. | The backend  server stores the  PDFs for future use. | Vector embeddings are created using  HuggingFace models. | User enters a query through the frontend interface. | The retrieved chunks are sent to an LLM for processing. | The generated answer is displayed to the user. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PDFs Sent to Backend** | **Text Splitter Breaks Content** | **Embeddings**  **Stored in**  **ChromaDB** | **Relevant Chunks Retrieved** | **Context-Aware**  **Answer**  **Generated** | **Chat History Stored** |
| The uploaded PDFs are transmitted to the backend server. | The PDF content is divided into smaller chunks. | The embeddings are stored in a persistent vector store. | The system retrieves the most relevant chunks from ChromaDB. | The LLM generates an answer based on the context. | The conversation history is saved for future reference. |

**PDF Processing and Query System**

# RAG-based PDF Chatbot Cycle

**Frontend Sends**

**Request**

Frontend displays the

response to the user.

Frontend sends the

upload request to the

backend.

**Frontend Displays**

**Response**

**Backend Sends**

**Response**

Backend sends the

generated response

back to the frontend.

Backend processes the

PDF and stores

embeddings.

**LLM Generates**

**Response**

**Vector Store**

**Retrieves Chunks**

**Backend Processes**

**PDF**

**User Uploads PDF**

User interacts with the frontend to upload a PDF.