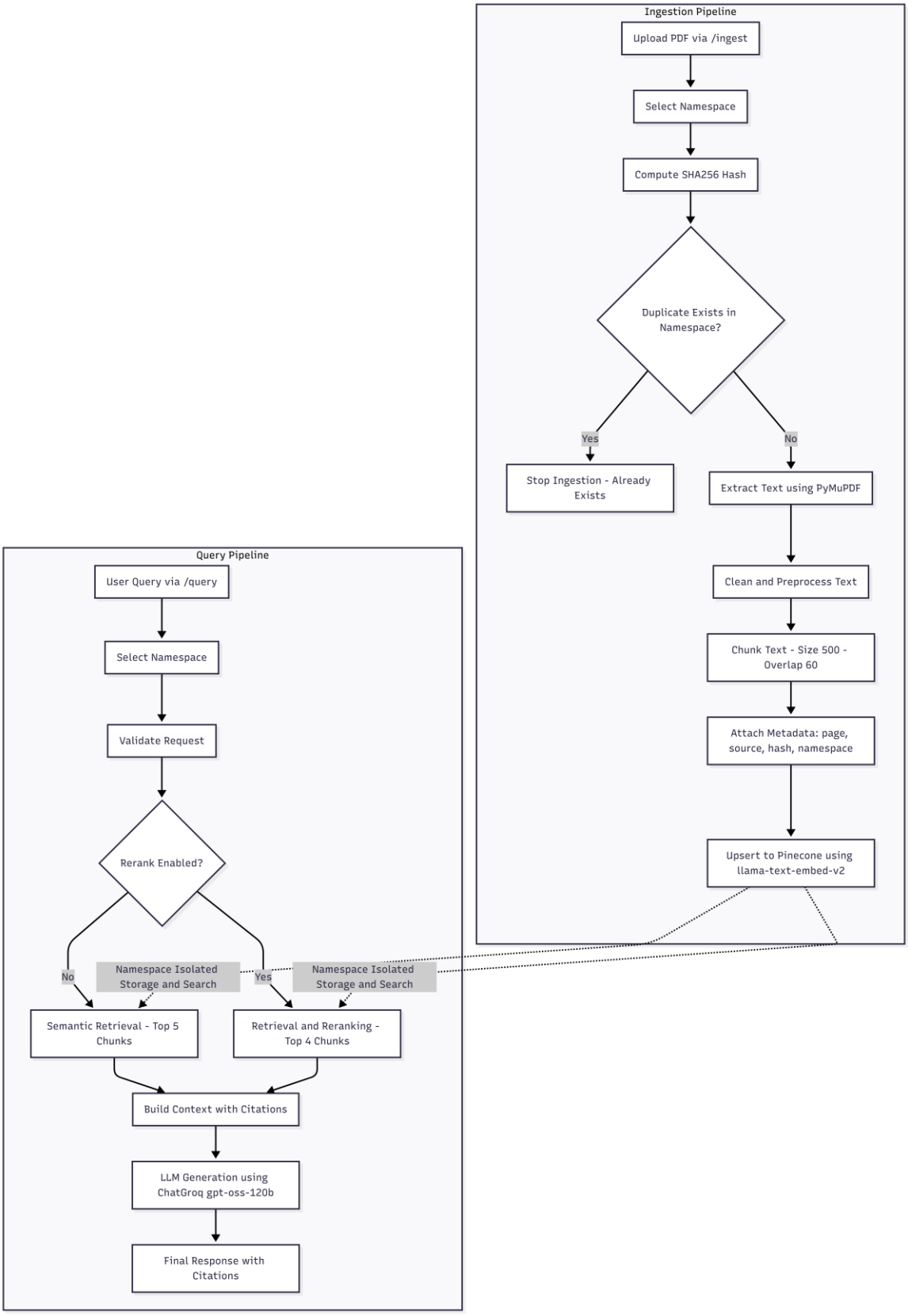
System Architecture: RAG Pipeline with Reranker (Classic)

Overview

This document describes the Retrieval-Augmented Generation (RAG) system with optional reranking. The architecture includes document ingestion, vector storage, semantic retrieval, reranking, and LLM-based answer generation with citations.

Architecture Flow (Diagram)



Architecture Flow (Step-by-Step)

1. User uploads a PDF document via POST /ingest endpoint.

2. System computes SHA256 hash of the document.

3. Pinecone is queried to check if the same hash already exists.

4. If duplicate is found, ingestion stops and returns 'Already Exists'.

5. If not duplicate, PDF is parsed using PyMuPDF.

6. Extracted text is cleaned (removing extra spaces and formatting noise).

7. Text is split into chunks of 500 characters with 60 character overlap. (configurable in config.py)

8. Each chunk is assigned metadata (page number, source filename, hash value).

9. Chunks are batched and upserted into Pinecone.

10. Pinecone generates embeddings using llama-text-embed-v2 automatically. (configurable)

11. User sends a query via POST /query endpoint.

12. System validates request and checks rerank flag.

13. If rerank is False, system retrieves Top 5 similar chunks from Pinecone.

14. If rerank is True, system retrieves initial results and reranks using bge-reranker-v2-m3.

15. Reranked system returns Top 4 refined chunks.

16. Retrieved chunks are combined into a structured context with citations.

17. Context is sent to LLM (ChatGroq using openai/gpt-oss-120b). (configurable)

18. LLM generates final answer with enforced citations.

19. Response is returned to the user.

Component Details

Embedding Model: llama-text-embed-v2 (Pinecone native serverless embedding).

Reranker Model: bge-reranker-v2-m3 (cross-attention reranker).

LLM Model: openai/gpt-oss-120b via ChatGroq.

Chunk Size: 500 characters.

Chunk Overlap: 60 characters.

Batch Size for Upsert: 96(max allowed)

Temperature: 0.0 for deterministic output.

Bonus Features

Duplicate prevention using SHA256 hashing, entire document not specific chunks.

Namespace-based data isolation, user can ingest into specific namespace using ingest endpoint. Also, user has the option to query by entering the specific namespace (if it exist). no auth implemented.