Microservices Architecture

**DAIKY**

**D**ocument **A**nalysis and **I**ntelligent **K**nowledge for **Y**ou

1. Document Service

Responsible for document ingestion and preprocessing

Components: DocumentParser, TextChunker

APIs:

POST /documents: Upload a document

POST /documents/{id}/parse: Parse a document and return the text

2. Embedding Service

Responsible for generating embeddings for text chunks and questions

Component: EmbeddingModel

APIs:

POST /embeddings: Generate an embedding for a text chunk or question

POST /embeddings/{id}: Retrieve an existing embedding by ID

3. Vector Database Service

Responsible for storing and retrieving vector embeddings

Component: VectorDatabase

APIs:

POST /vectors: Store a vector embedding

GET /vectors/{id}: Retrieve a vector embedding by ID

POST /vectors/search: Search for similar vector embeddings

4. Language Model Service

Responsible for generating answers to user questions

Component: LLM

APIs:

POST /answers: Generate an answer to a user question

5. UI Service

Responsible for handling user interactions and displaying results

Component: UI

APIs:

GET /: Display the user interface

POST /questions: Handle a user question and display the answer

Benefits of this Structure

Each microservice has a clear responsibility and can be developed and scaled independently

The APIs between microservices are well-defined and can be used to integrate the services

The structure allows for easy addition of new features or services as needed