KB Retrotool - Tech Documentation

Overview

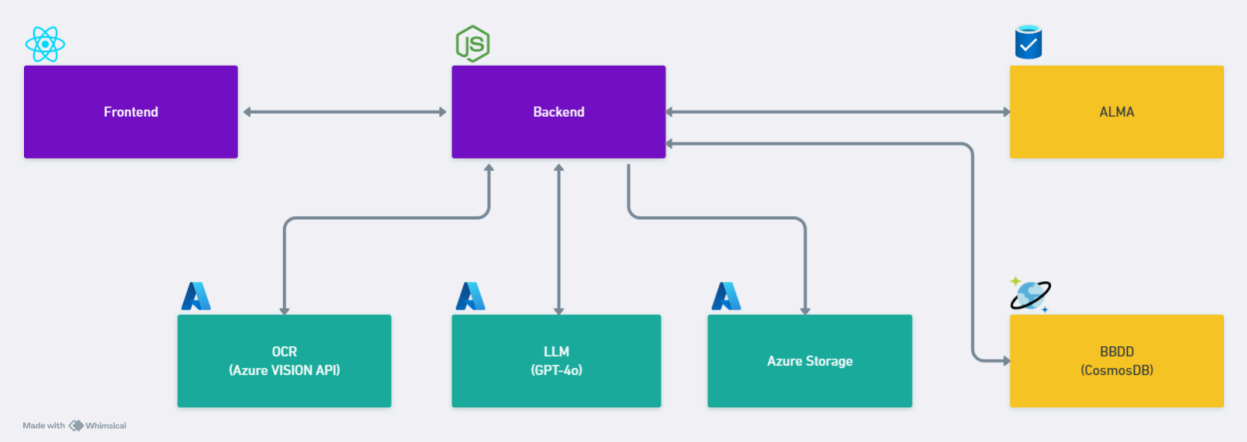
The KB Retrotool is designed to streamline the process of identifying books by leveraging Optical Character Recognition (OCR) and Large Language Models (LLMs). This system facilitates the extraction of text from book images and determines if the scanned books are already cataloged in the collection of the National Library of the Netherlands. The backend interacts with the ALMA Database to perform this lookup.

System Architecture

1. **Frontend**:
    - Users take photos of books with a specific camera for scanning documents using the web interface.
    - After capturing the image, it is submitted to the backend.
    - A form is pre-filled with metadata extracted from the book's content and presented to the user.
    - The user can take multiple photos from the different parts of the book and manually edit the form if some corrections are needed.
    - The user can just submit the form or save it as a draft. There is a draft page where the user can load and use all the saved drafts when needed.
2. **Backend**:
    - Receives book images from the frontend.
    - Performs OCR processing to extract text from images.
    - Sends the extracted text to a Large Language Model (LLM, specifically GPT-4o) to generate a structured JSON containing metadata about the book (e.g., title, author, publication year, etc.).

3. **Alma Database Interaction**:
    - After the user reviews and submits the form, the backend queries the Alma Database to check if the book already exists in the library's collection.
4. **Result Handling**:
    - If the book exists in the Alma Database, the user is notified with the matching records.
    - If no match is found, the user can proceed with additional steps, such as download the XML with the metadata for adding the book to the library's collection.

Architecture Diagram



Technical Details

Frontend

- **Runs on**: Azure Static Web Apps (in West Europe)
- **Technology Stack**: React.js (Web)
- **Hardware used**: CZUR Fancy S Pro - document camera for scanning the book pictures.
- **Key Features**:

- Image capture and upload functionality.
- Form generation based on JSON metadata.
- User-friendly interface for reviewing and submitting data.

Backend

- **Runs on**: Azure App Service (in West Europe)
- **Technology Stack**: Node.js, Azure Functions, CosmosDB, Azure storage.
- **Key Features**:
  - Endpoint for receiving book images.
  - Integration with OCR engine and LLM APIs.
  - Handling Alma Database queries and responses.
  - CosmosDB database for storing users, drafts, logs etc.
  - Azure storage for save the images for a limited period of time (7 days).

OCR Engine

- **Technology**: Azure Vision API (in West Europe)
- **Functionality**: Extracts text from images with high accuracy, supporting multiple languages if required.

LLM Integration

- **Model**: Azure OpenAI GPT-4o (in West Europe)
- **Functionality**:
  - Processes extracted text to generate structured JSON metadata about the book.

Alma Database

- **Database Type**: Proprietary Database of the National Library of the Netherlands
- **Interaction**:

- Queries are made using a predefined API provided by Alma.
- Results are processed and returned to the frontend for user review.
- Book metadata is saved in MARC XML 21 format, the standard used by most of the libraries.