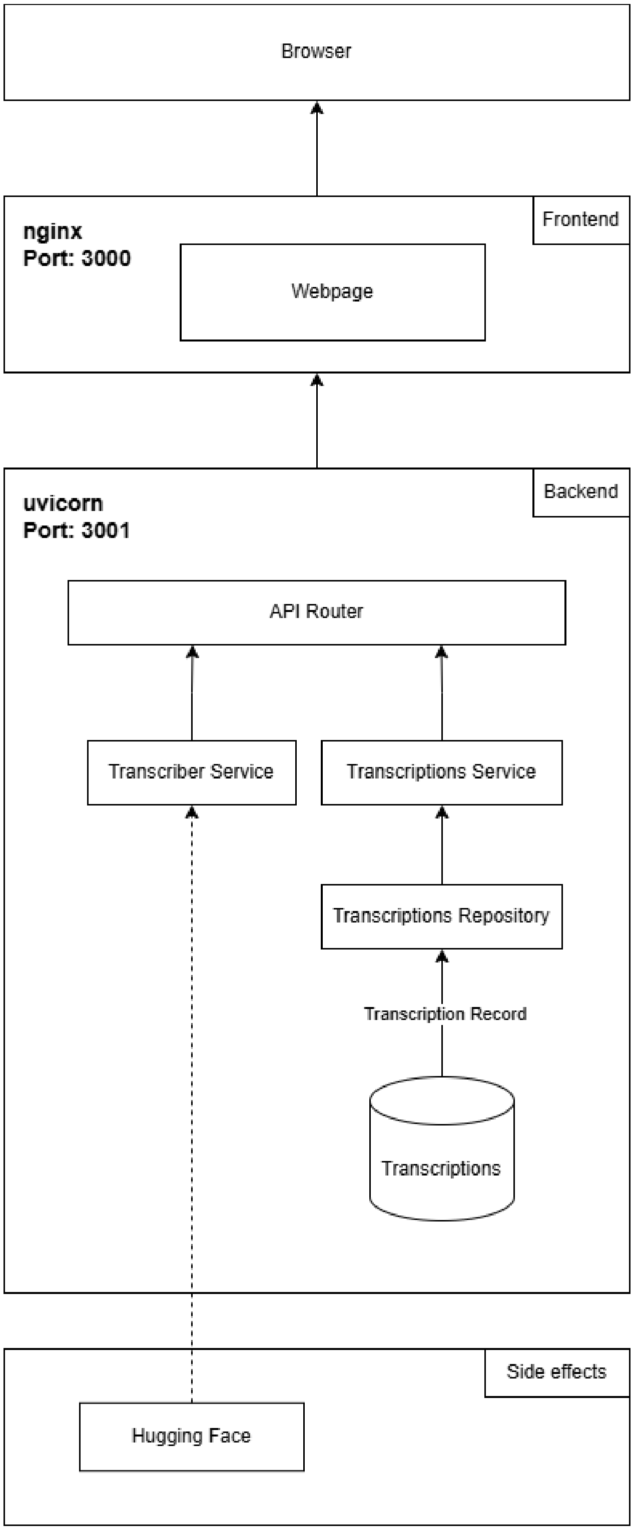
Web Application Architecture Diagram



Frontend

A single page application written using the React.js framework. It uses Material UI as its CSS library and the axios library to make requests to the backend API endpoints. Running via docker, it's hosted with nginx on port 3000.

Features:

- Search for transcription records via file name or transcribed text
- Multi-upload of audio files for transcribing
- Display all transcription records

Backend

A RESTful API running with the Python FastAPI web framework. Running via docker, it's hosted with uvicorn on port 3001.

It has 2 primary services which the router depends on for its endpoints: the transcriber service and the transcriptions service.

- The transcriber service transcribes the audio files.
  - Its primary side effect is downloading the relevant models from Hugging Face upon initialisation of the service.
- The transcription service acts as an interface for managing transcription operations, it depends on the transcriptions repository which is responsible for database related queries.
  - Assuming that transcription records cannot have the same audio file name, a unique constraint was applied, uploading files with the same name will return an error.
  - Typically databases are hosted externally but we're using SQLite which is simple enough to contain within our disk, it's kept within the backend boundary.