



# Assumptions / Considerations:

## 1) Debounce and Rate Limiter for Performance

- Debounce minimizes repetitive API calls triggered by user interactions, optimizing resource usage and improving backend efficiency
- Rate limiting prevents excessive requests from overwhelming the server

## 2) Mounted File System for Playback

- Storage of upload path to file system to audio playback
- Files referenced by UUID to safeguard files against unauthorized access
- Possible file size limits to prevent overloading the system and checking of file MIME type etc to prevent unintended/malicious uploads
- Off premise object storage such as S3 can provide better security / scalability for large storage sizes

## 3) Nginx as Reverse Proxy to Deploy Frontend Page

- Optimised to serve static content (HTML / JS / CSS)

## 4) FTS5 Virtual Table for Full-Text Search

- Transcription FTS table based on SQLite's FTS5 supports file name searches
- Create, update and delete triggers to ensure the transcription table and fts table are in sync