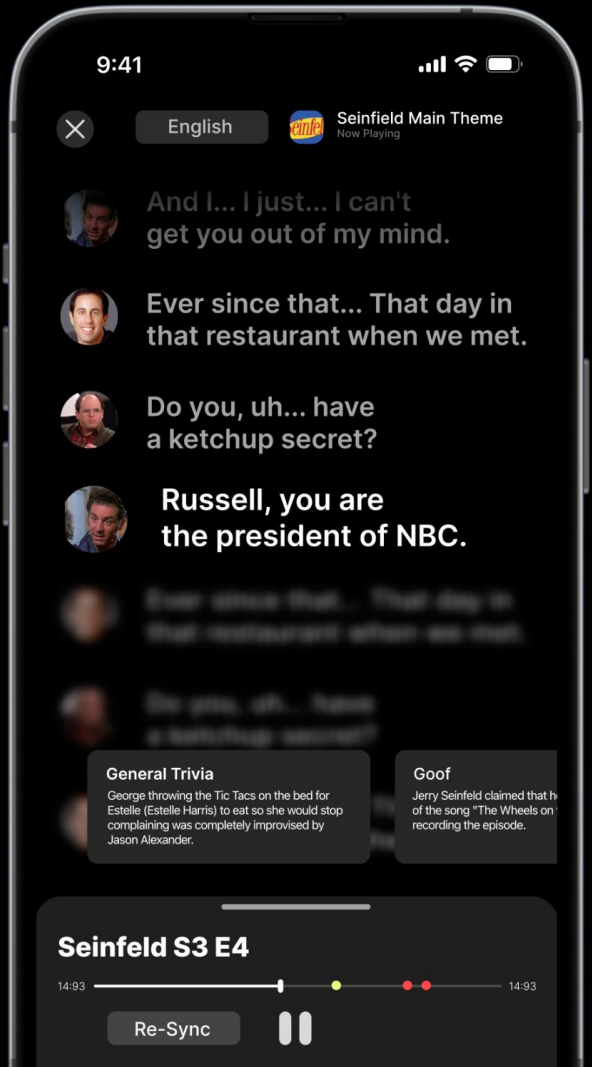


Subtitle**Vision**

Technical Design

Aaron Kovacs, Julia Showl, Ryan Rosica

System Goals

- Your real-time movie and TV Show guide

- Using microphone: Recognize Movie/TV Show and determine current playback time

- Collect and show tangential information: Subtitles, Who is speaking, Actors, etc. in an iOS app

Key Components

Speech to Text
(Julia)
Using Microsoft Azure speech to text API

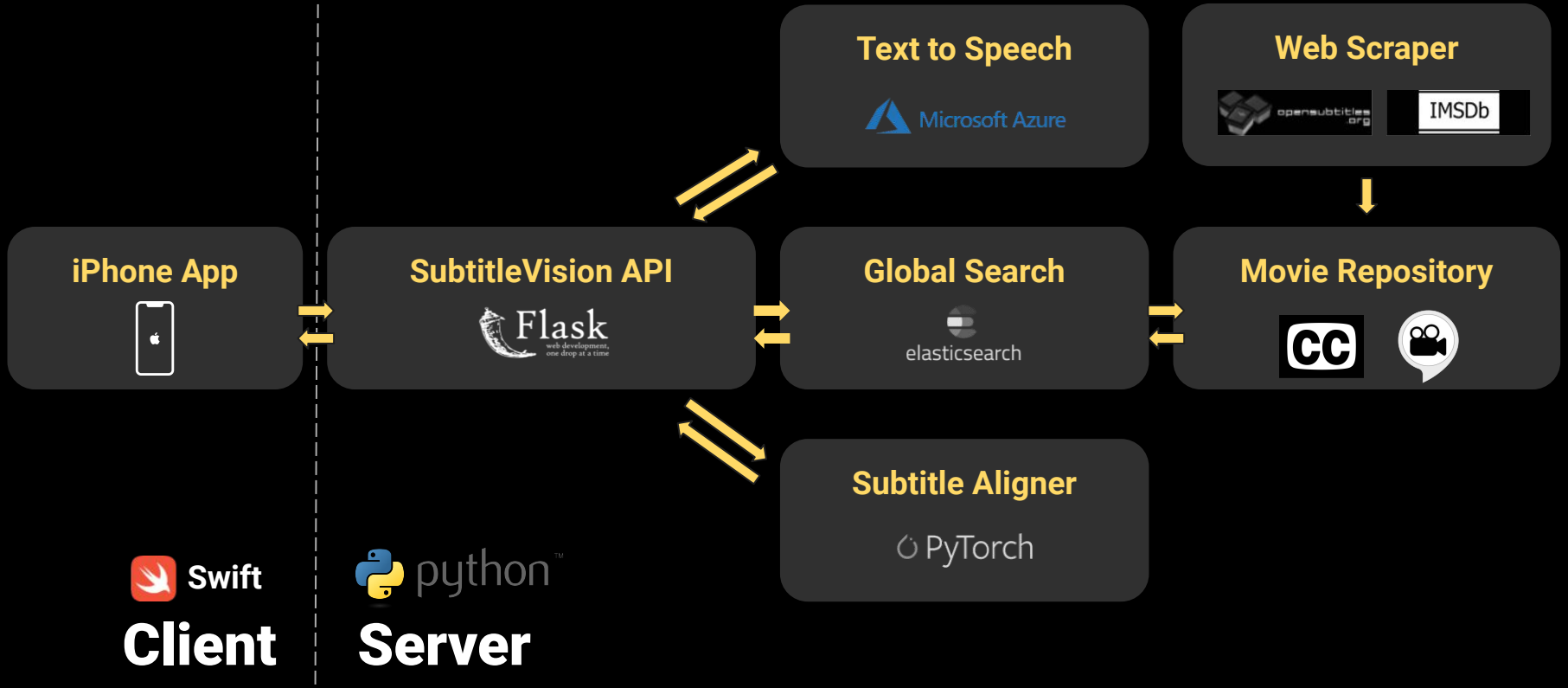
Global Search
(Aaron)
Search for substring within subtitle file dataset

Subtitle Aligner
(Julia + Ryan)
Forced alignment between audio and subtitle file

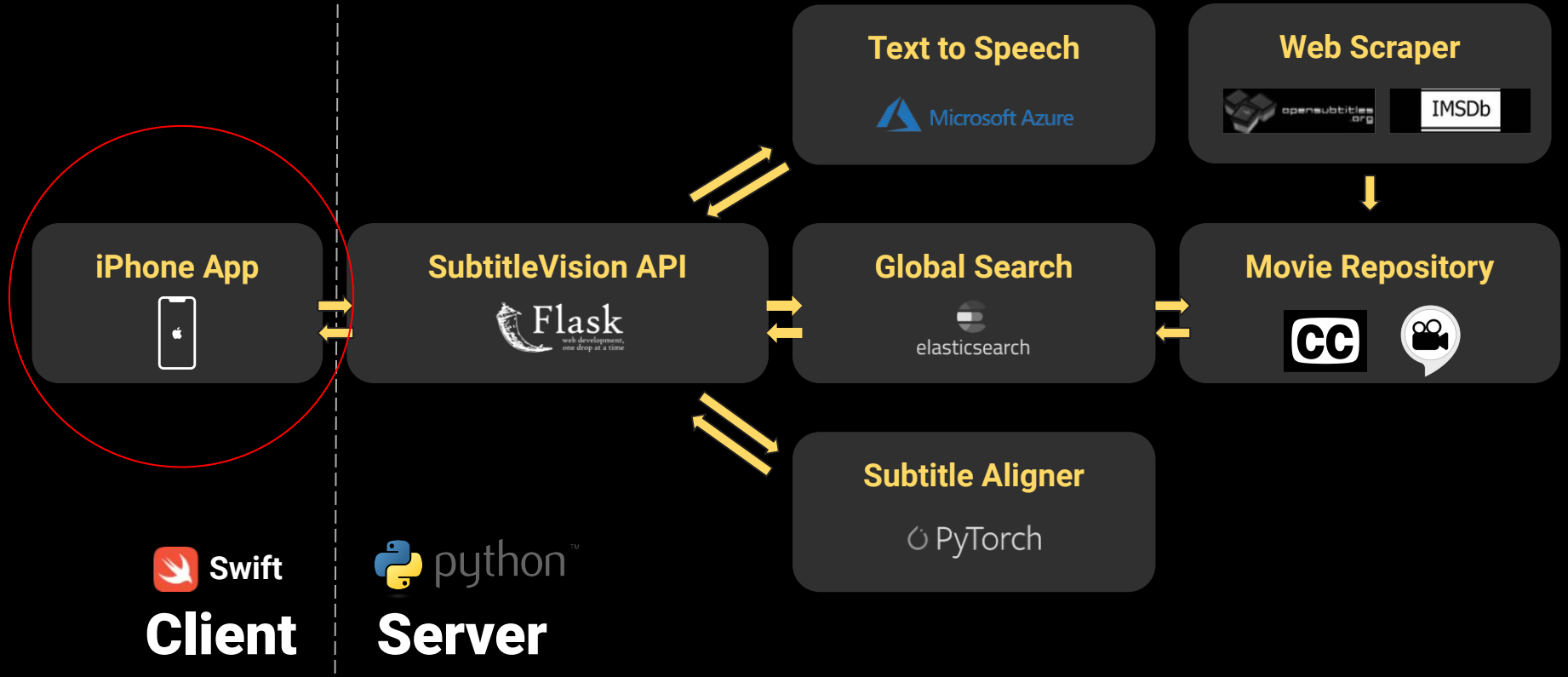
iOS App
(Aaron + Ryan)
Front end user interface for displaying the information

Movie Data Repository
(Julia)
Scrape and organize movie data

System Architecture



System Architecture

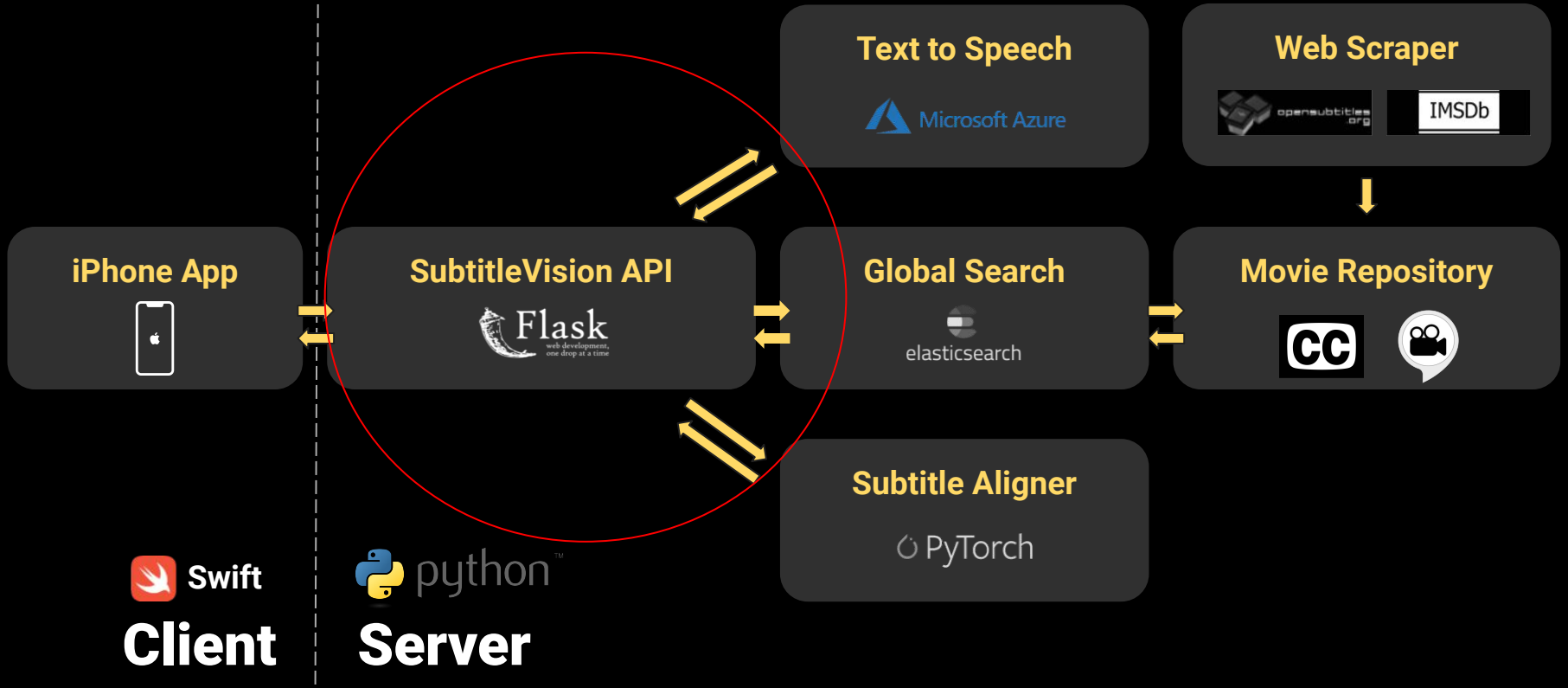


Step 1: iOS App

- Goal: A display that is convenient and intuitive for the Movies, TV Shows, and Subtitles

- The Composable Architecture
  - Redux-like
  - focus on dependency injection
- UI: SwiftUI and UIKit
- Networking:
  - RESTful
  - Potentially WebSockets for progressive recording

System Architecture



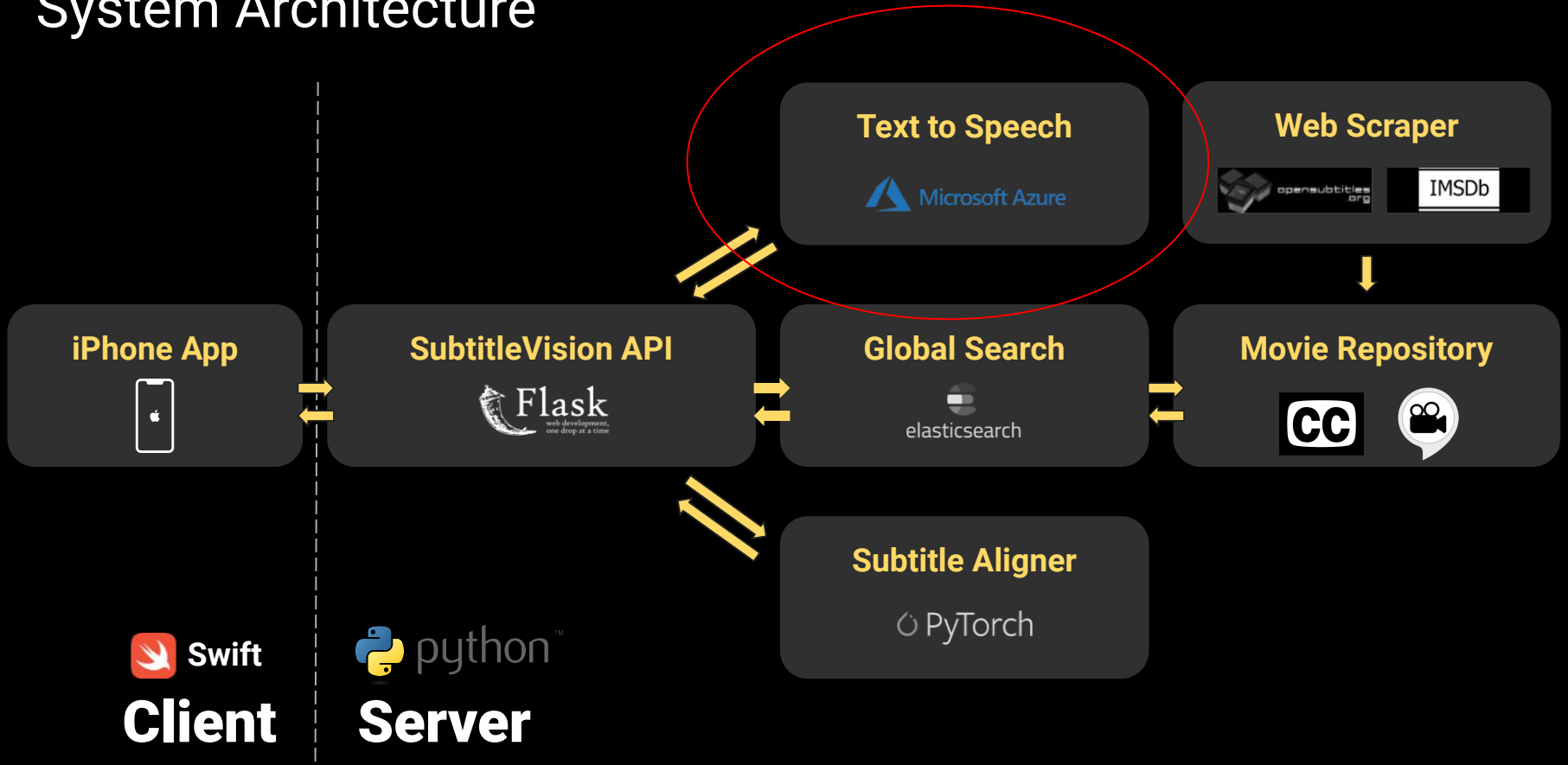
Step 2: Flask RESTful API

- Inputs: HTTP requests from the iOS app (Audio Recording Files)
- Goals: Utilize speech to text, global search, and subtitle aligner services to provide necessary information back to the client

- Flask - Simple, flexible web framework in Python
- RESTful - Stateless, responds with JSON format
- The “glue” for combining our core services (speech to text, global search, subtitle aligner)

1. Call speech to text service to produce transcript
2. Call global search to Identify the movie and rough playback time
3. Call subtitle aligner to retrieve exact synchronization information

System Architecture

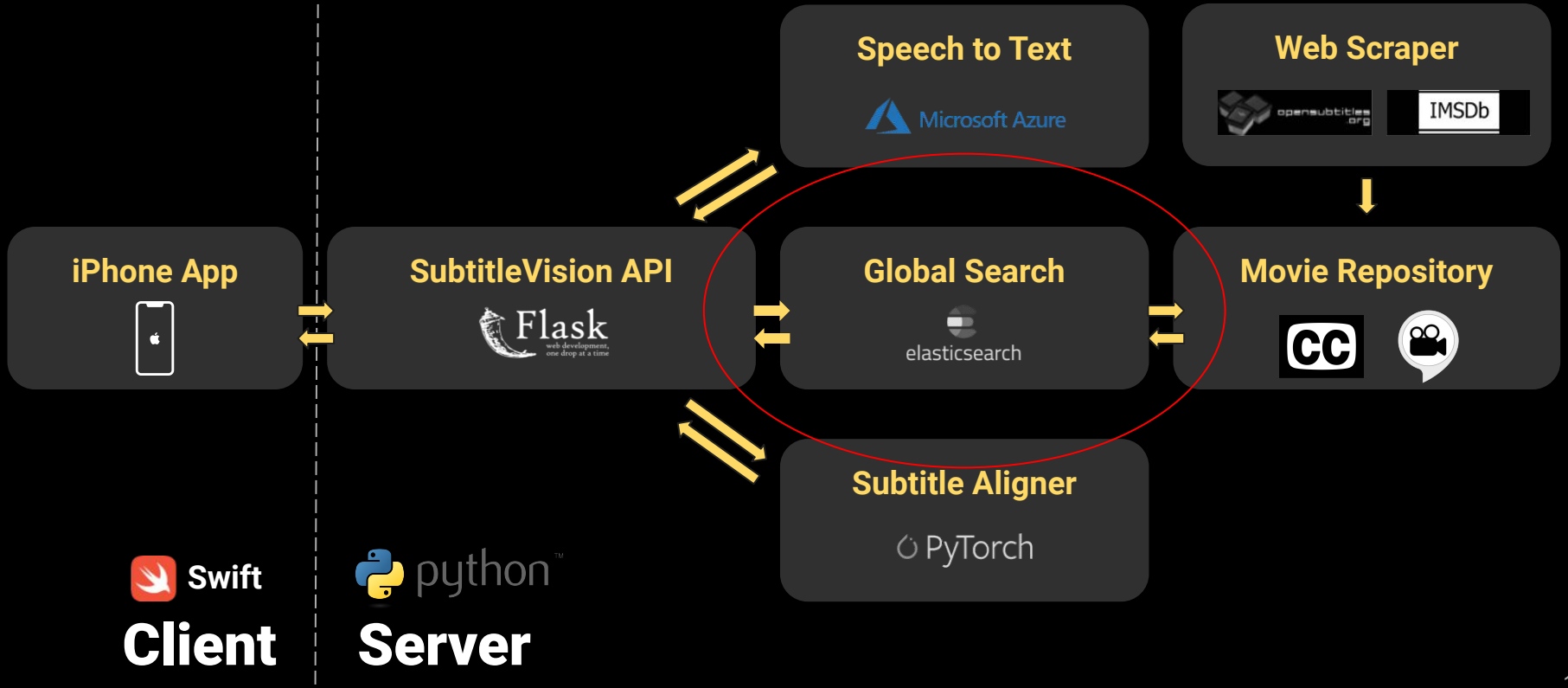


Step 3: Speech To Text

- Inputs: Phone’s microphone recording rolling audio sample
- Goal: Transcribe the audio sample to text

- Backend does progressive processing on the sample
- Utilizing Microsoft Azure Cognitive Services to transcribe sample

System Architecture

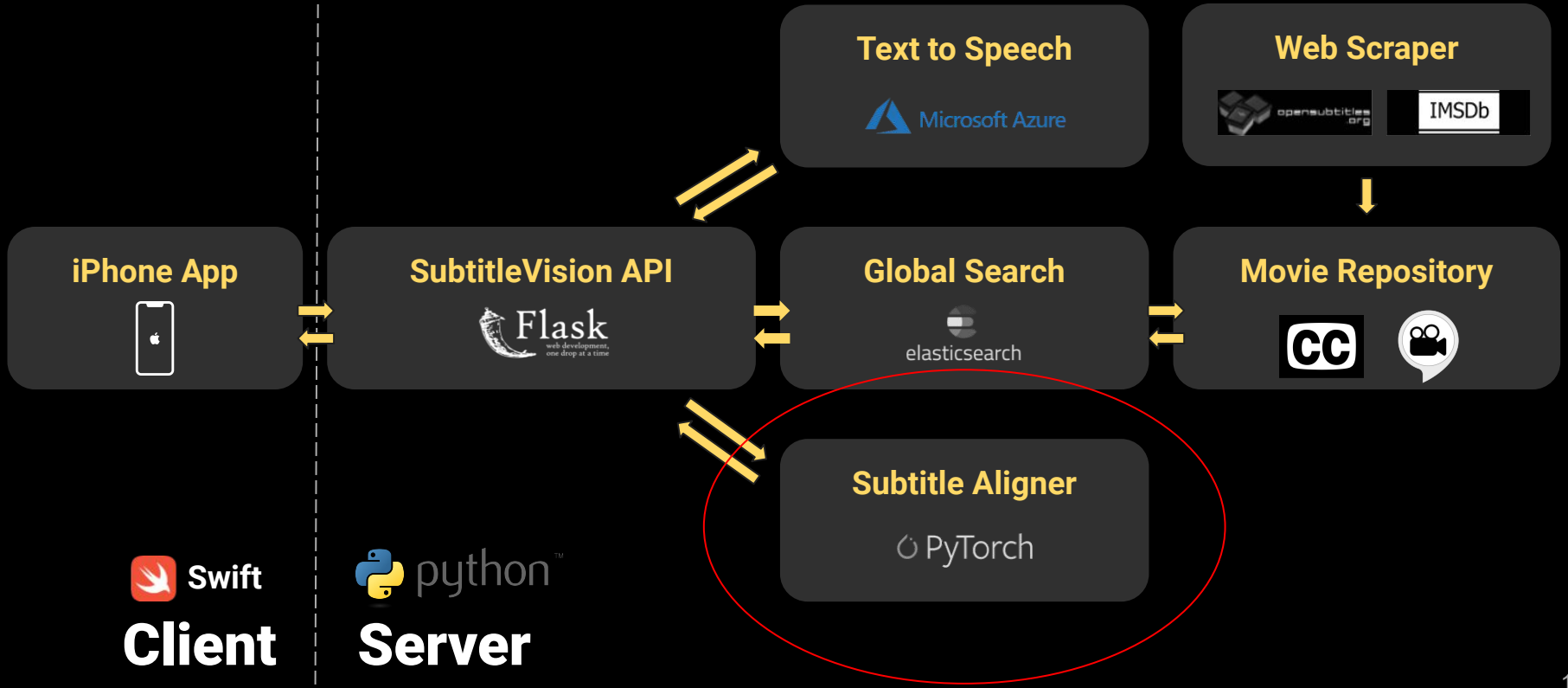


Step 4: Global Search

- Inputs: Transcription of audio recording
- Goal: Determine the playing movie or TV show and the approximate playback time (within 3-5 lines of subtitles)

- Using our own pre-processed database of subtitle files
- ElasticSearch index + “Fuzzy search”
- Challenges:
  - Optimizing performance
    - Keeping entire database in memory for quick access
  - Optimizing Accuracy
    - Eliminating false positives

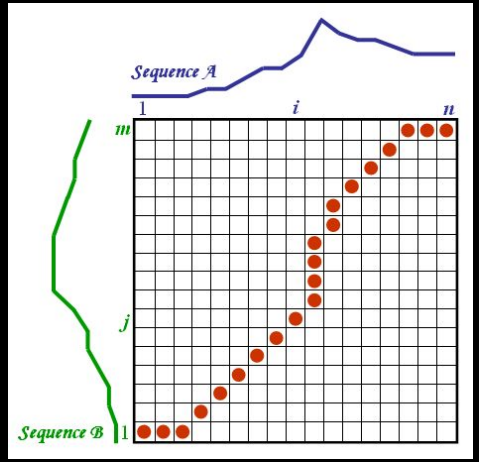
System Architecture



Step 5: Subtitle Synchronization

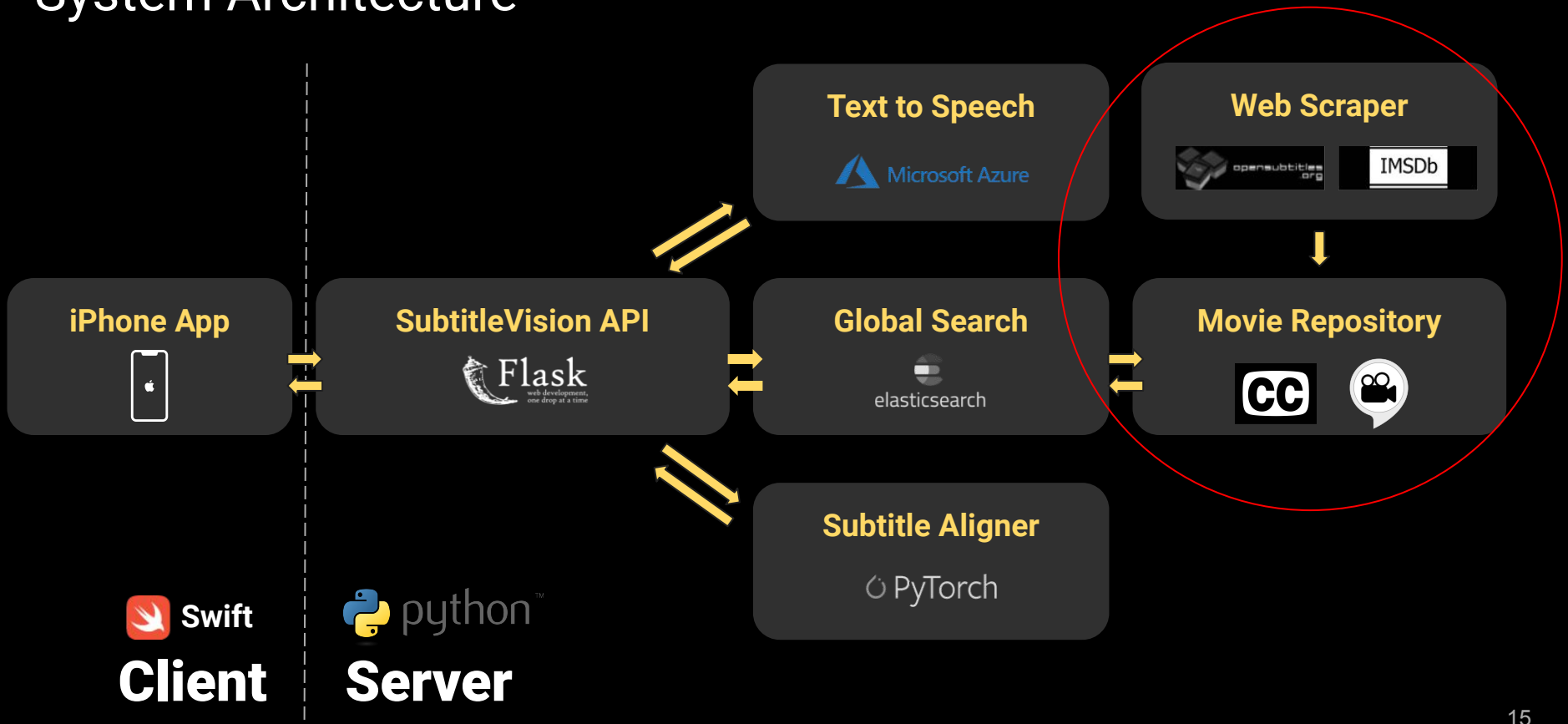
- Inputs: Speech Sample and 3-5 lines of subtitles
- Goal: Match a time in the speech sample with an exact playback time in subtitle file

- Strategy: Force Alignment Algorithm
  - Use pre-trained speech to text machine learning model
  - Calculate probabilities that each section of the audio matches each word in the transcript (matrix)
  - Matches up each word in the subtitle lines with their corresponding time in the recording



https://techfirst.medium.com/forced-alignment-how-to-match-audio-with-a-transcript-via-machine-learning-dd19da8c0f04

System Architecture



(Scrapers) Building Movie Data Repository

- Goal: Build repository of subtitles files and movie information, perform pre-processing

- Scrapers will collect and format subtitles for later use in Global Search and synchronization.
  - IMSDb and opensubtitles.org
- Building up a database is important for testing Global Search
- Scrape screenplays to identify character and scene information
- Pre-processing involves formatting the subtitles files (.srt) into JSON files that can be used unassisted in other parts of the backend.

Feasibility

Global Search

- One of the most challenging technical aspects of SubtitleVision
- Naive solutions will not scale, so creative problem solving is necessary
- Luckily text search is a well documented problem

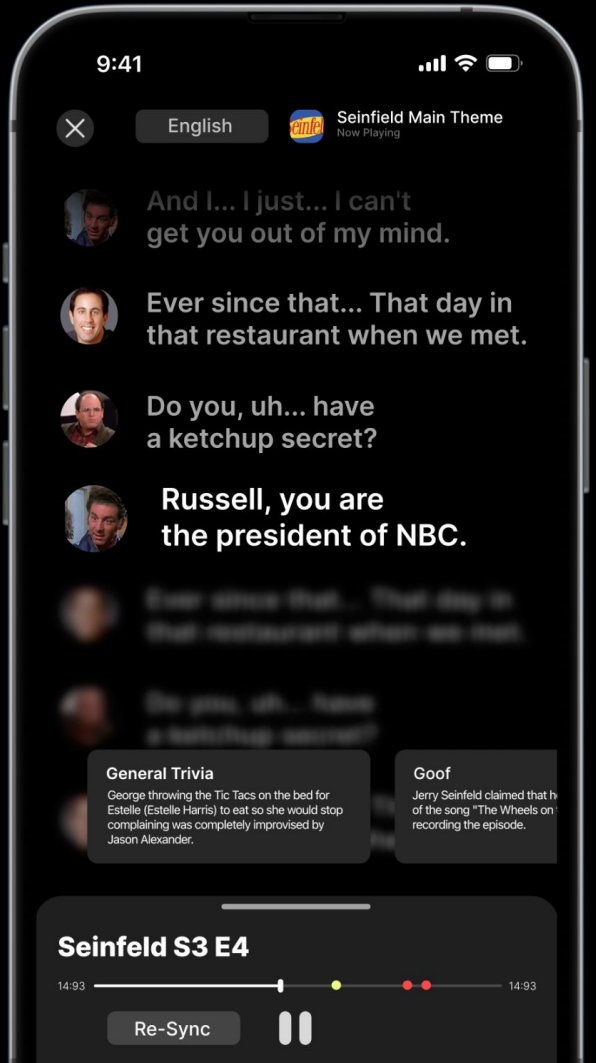
Synchronization

- Our proof of concepts have the potential of naive solutions for macro-level sync. Forced alignments provides a possible very solid answer to the micro-level sync.

iOS App, Speech To Text, etc. have shown to be very doable in early work. We don’t expect trouble in these areas.

Alpha Prototype

- Our alpha will be working demo of that initial flow
  - Limited Global Search
  - A strong implementation of synchronization
  - iOS app that can record the audio and display subtitles
- From here the initial (key) pieces are in place



Questions?