



Casey
Chow

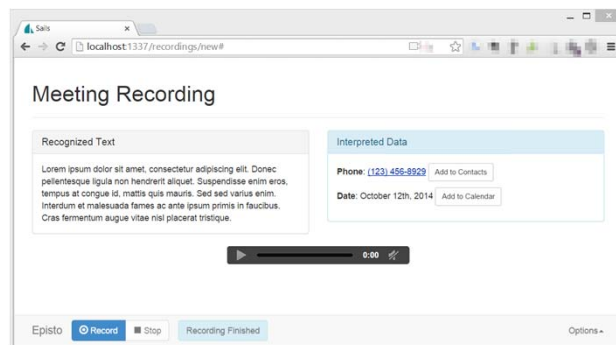
Sean
Duffy

Record. Summarize. Understand.

Purpose

At once note-taker and virtual assistant, Episto acts as an alternate way to interacting with audio recordings. Episto extracts meaningful phrases, numbers, and names from the meeting, presents a meeting summary to the user, and suggests possible actions it can take based on the information extracted from the meeting.

Scenario

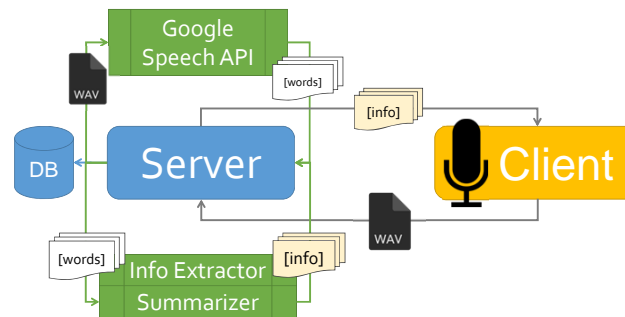


1. User opens Episto on phone or laptop, starts recording meeting.
2. Episto presents live transcript of meeting and collects information.
3. User saves meeting summary and recording to Evernote, adds phone #s and dates to Contacts and Calendar.

Architecture

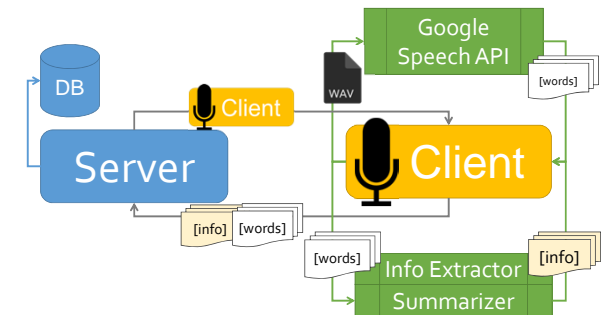
Episto is implemented in a **client-server architecture**, using Node.js and Google Chrome as the respective JavaScript runtimes. Two architectures were devised for this application: a server-centric and a browser-centric model. Both models rely on the same technologies (WebRTC, Google Speech API, WebSockets) for data management.

Server-Centric Arch.



- Client streams audio to the server.
- Server pipes the audio to the Google Speech API, then extractors and summarizers, and returns final result.
- Server-side Google Speech API is poorly documented, making server-side **impractical** for now.

Client-Centric Arch.



- Server only serves the client code and stores outputted information.
- Client handles all speech and data manipulation.
- Used as a stopgap; Google's client-side speech APIs are significantly more reliable.
- Cannot recognize speech and record simultaneously.

Future Goals

- Move to server-centric architecture when functional API is available.
- Improve code reuse to allow other projects to build upon Episto.
- Research ways to improve recognition/processing accuracy.