Capstone Project Proposal

Group 16
COMPSCI 4ZP6
Dr. Mehdi Moradi
Sep 20, 2024

Alex Eckardt	eckardta@mcmaster.ca	400390784
Owen Gretzinger	gretzino@mcmaster.ca	400407289
Jason Huang	huanj168@mcmaster.ca	400374849
Sahib Khokhar	khokhs5@mcmaster.ca	400396918
Sarah Simionescu	simiones@mcmaster.ca	400363648

Description

We are developing an open-source meeting bot API that integrates with online meeting platforms like Zoom, Microsoft Teams, and Google Meet. Essentially, the meeting bots can join video calls and record audio, which can then be accessed programmatically through an API. A couple of examples of applications that could be built using our API are an AI meeting notetaker or an AI sales coach.

This project will involve building a frontend for the user to manage configuration, a backend for controlling bot activity & providing API routes, bots that can automatically join scheduled meetings to record audio and upload recordings to AWS S3, and setting up an AWS ECS infrastructure for scaling bots. Additionally, we will create a deployment feature using Terraform to enable users to set up their infrastructure with one click.

The project offers a challenging scope, requiring our group of 5 to design and implement containerized bots, configure cloud-based storage, create a backend API, and develop and host a full-stack project. Outside of development, the team will dedicate time to having user conversations and gain a deep understanding of user needs and requirements.

Expected Functionality

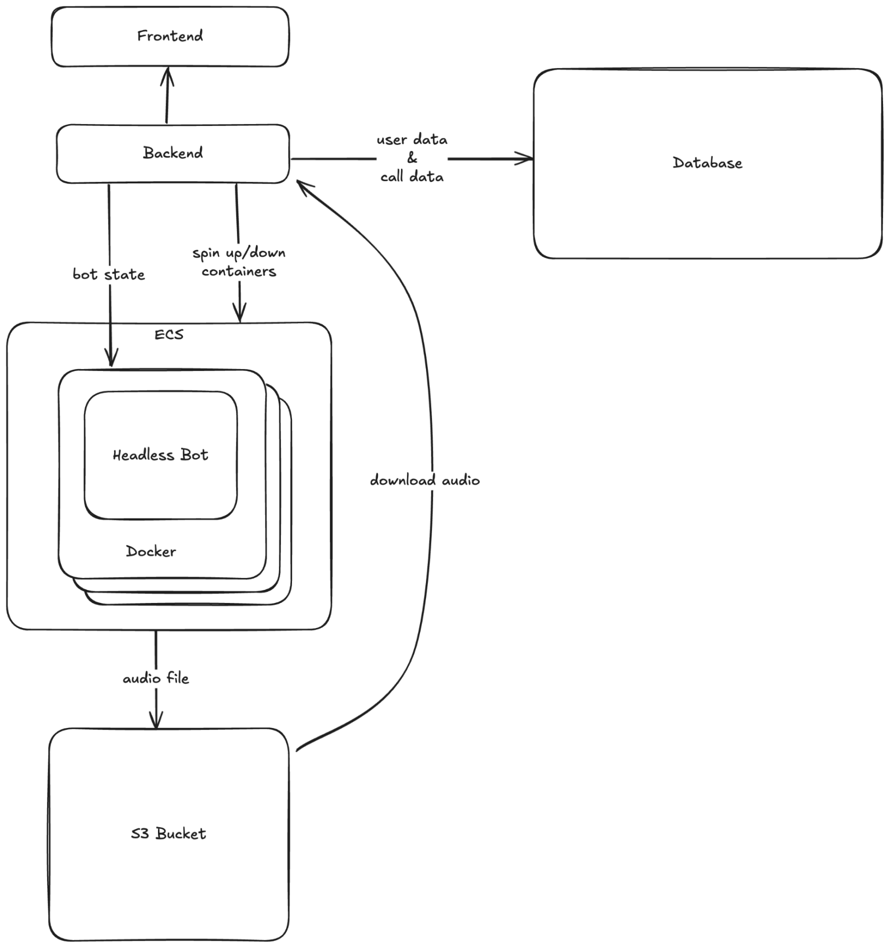
Our project's most basic expected functionality will include a bot that will join a meeting through a provided link, then record audio and make it available for download through an API. This is what we will build first as our MVP.

By the end of our project, we would like to support the following functionality additionally:

- Include support for multiple meeting platforms
- Generate meeting transcripts for access through the API
- Develop a deployment feature using Terraform, which would automatically deploy and configure a user’s AWS account to host the service in one click
- Make it so that bots can be added as a participant at the creation of the meeting, which would then schedule it to join as the meeting begins
- An example application using our project that summarizes meetings and generates action items
- Optionally record meeting video and upload it to S3

Appendix

High–level Architecture Diagram



- **Frontend** – Frontend application, hosted at meetingbot.tech. Provides configuration functionality and the ability to access MP3s/transcripts through a visual interface.
- **Backend** – Acts as a control plane for the bots and provides state management, services API requests, and handles auth.
- **ECS** – Elastic Container Service: Scales up and down based on demand. Holds bot instances.
- **Headless Bot –** Contained within Docker images, these are the bots that join and record the meeting, controlled by the Backend. Converts and uploads recording data to S3 when the meeting concludes.
- **S3 Bucket** – Stores meeting recording files.
- **Database** – A PostgreSQL Server. Contains user data, links to the MP3 files stored in the S3 Bucket, transcription data, and other metadata.

Dev Tools and Languages

- **Javascript** – frontend (TypeScript)
- **Python** – backend
- **NodeJS** – bots
- **Docker** – containerizing the bots for scalability & smooth deployment
- **Terraform** – infrastructure as code for one-click self-hosting
- **PostgreSQL** – database
- **AWS** – hosting

Product Management Tools

- **Linear** – Task/Product Management
- **Google Drive** – File Storage, Collaboration
- **GitHub** – Repository Hosting
- **Discord** – Communication