4.3 Proposed Design

4.3.1 Overview

Provide a high-level description of your current design. This description should be understandable to non-engineers (i.e., the general public). Describe key components or sub-systems and how they contribute to the overall design. You may wish to include a basic block diagram, infographic, or other visual to help communicate the overall design.

We have three major components in the current design: backend, frontend and the database. Database is where all the data will be stored and it has it's own request hander to handler all the requests. Backend is where all the logic of the application is performed like for example all the machine learning models and API's live here and it also has it's own request handler, so it could send and receive request to and from the frontend and the database. Frontend is where all the visual are processed it has it's own request handler so it could request data from the database and display it.

4.3.2 Detailed Design and Visual(s)

Provide a detailed, technical description of your design, aided by visualizations. This description should be understandable to peer engineers. In other words, it should be clearly written and sufficiently detail such that another senior design team can look through it and implement it.

The description should include a high-level overview written for peer engineers. This should list all sub-systems or components, their role in the whole system, and how they will be integrated or interconnected. A visual should accompany this description. Typically, a detailed block diagram will suffice, but other visual forms can be acceptable.

The description should also include more specific descriptions of sub-systems and components (e.g., their internal operations). Once again, a good rule of thumb is: could another engineer with similar expertise build the component/sub-system based on your description? Use visualizations to support your descriptions. Different visual types may be relevant to different types of projects, components, or subsystems. You may include, but are not limited to: block diagrams, circuit diagrams, sketches/pictures of physical components and their operation, wireframes, etc.

Our project will be a system consisting of a React web app frontend, and several backend services connected through an API, as seen in Fig. 1. The frontend will allow users (power company personal, government monitoring agents, etc.) to view the current state of the system, as seen in Fig. 2. Users will be able to examine individual nodes including their past and

predicted future values. Furthermore, the frontend will alert users to current or predicted future anomalies which may indicate dangers to the power grid's health. Users will have the option of setting filters on the alerts received, for example a user may limit their alerts received to a specific state or municipality. To increase ease of use, users can create specific profiles to store their information, such as location or role, as well as their selected filters. An additional feature of the frontend will be the implementation of a voice assistant, allowing more natural and flexible interaction with the system. The voice assistant will be integrated with the web app frontend. The backend will consist of several microservices, deployed to Google Cloud Platform (GCP). Influx and Neo4j will be used for backend data storage. The backend data storage will utilize time-series and graph based databases to efficiently store and serve recent data with accurate geographical positions on the frontend views. The frontend will use GCP Firebase for data storage specific to the frontend, such as profile information. Our system will leverage Tensorflow on Python, through two machine learning (ML) services designed to predict future values and detect anomalies respectively. The fontend and backend services all communicate with each other through a RESTful API implemented in Go (also known as Golang). External data sources, shown as the transmitter nodes in Fig. 1, will push data to the system through this API. This is how data in the system at large is updated.

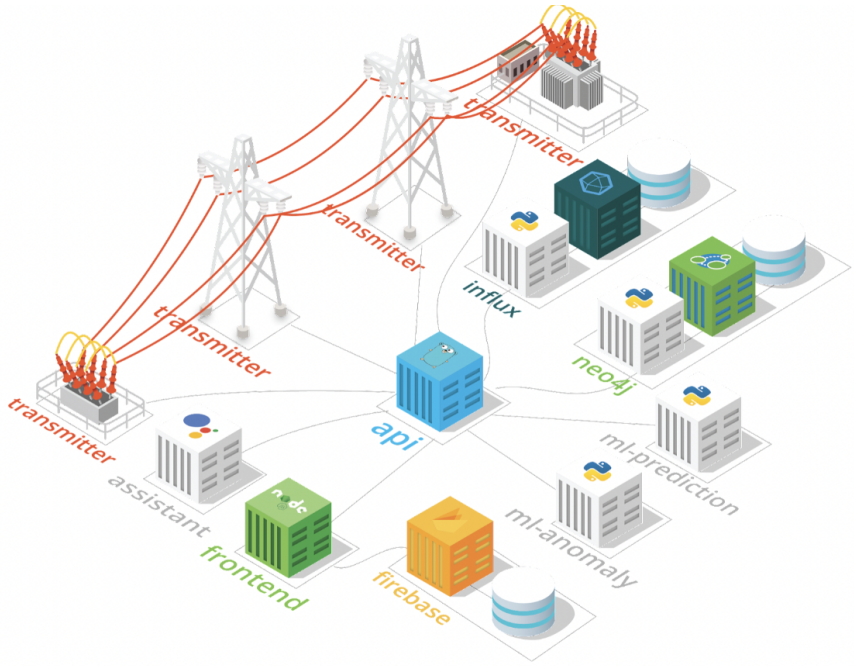

Fig. 1, Design Components Overview

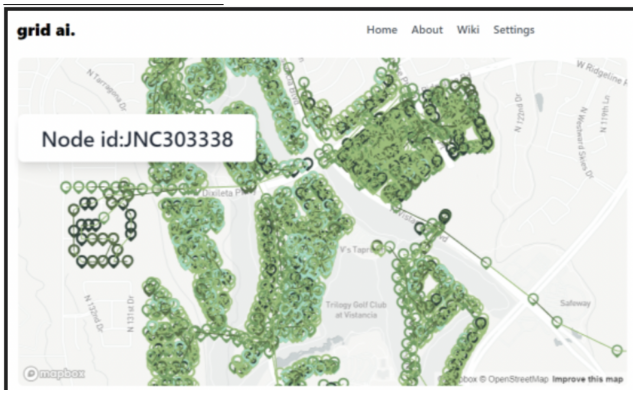

Fig. 2, UI Basic View

4.3.3 Functionality

Describe how your design is intended to operate in its user and/or real-world context. What would a user do? How would the device/system/etc. respond? This description can be supplemented by a visual, such as a timeline, storyboard, or sketch.

In a real world context, a grid operator/inspector would use our design in the form of a website that they'd be able to log on to in a tablet or smartphone. He/she would be able to see different types of data both visually through the help of graphs, charts, and map representations; plus through texts or written format. The user would be able to use his/her voice to get information through a google voice assistant that uses machine learning to predict and provide appropriate responses/data.

4.3.4 Areas of Concern and Development

How well does/will the current design satisfy requirements and meet user needs?

Our team's current design should satisfy the requirements and meet user needs. I belive we will be able to achieve the all goals that we have set for this project

Based on your current design, what are your primary concerns for delivering a product/system that addresses requirements and meets user and client needs?

The area of concern would be that have to made our our machine learning models to make the applications work as there are no machine learning models in place to build on top of them.

What are your immediate plans for developing the solution to address those concerns? What questions do you have for clients, TAs, and faculty advisers?

Our current plans included making a robust design for the whole project and to learn as much as we can on the machine learning aspect of things so we can implement machine learning models in the near future.

4.4 Technology Considerations

Describe the distinct technologies you are using in your design. Highlight the strengths, weakness, and trade-offs made in technology available. Discuss possible solutions and design alternatives.

Our design will purely be software based so most of or all of our technologies shall be on the software side of things. That said, we'll be implementing neural networks for the Machine learning model. When it comes to the voice assistant, we'll be using google's virtual assistant action center and webhooks to pass api and expected responses. This part shall be implemented in python. Neo4j shall be used to represent our influx database visually, plus react for the front-end too! We also have a top of the line server.

4.5 Design Analysis

Discuss what you have done so far, i.e., what have you built, implemented, or tested? Did your proposed design from 4.3 work? Why or why not? Based on what has worked or not worked (e.g., what you have or haven't been able to build, what functioned as expected or not), what plans do you have for future design and implementation work? For example, are there implications for the overall feasibility of your design or have you just experienced build issues?

We have a two year old rich repository for our design — courtesy of the two teams that worked on GridAi before us. In terms of what we've done so far, we've made weekly slides and presentations to our client to discuss/plan various aspects of the project. We have presented on what we know, on how to improve the project, on deeper understandings of the project (aided with visuals) and many other important topics. We've also worked on class documentation like Design context, proposed design and many others; plus a lightning talk we did this week. From a technical standpoint, we have migrated to our own virtual machines, accessed our own GCP accounts and gotten our firebase, gitlab and it's environment up and running. We are actively studying the code base and learning the skillsets we need to make bigger strides into the project. Our proposed design works but it's not complete since it misses major components. We plan on getting our react website running and integrating it with the voice assistant, plus also making new Machine Learning Neuronetworks.