



Client Overview

- The client is a pioneer in providing technology related services and consulting to the power and utility industry



Application Overview

- Development of a Line Mapper application featuring Multi-tenancy, Multi-threading, data compression, effective handling of Geo JSON files, efficient processing of data for performance enhancement etc., to replace a legacy system



Business Challenges

- When Sync was initiated, a full set of information was being transmitted requiring a lot of time to load the map Geo JSON
- Geo JSON updates happen based on the time of sync instead of last updated time of records
- Download of data associated to each point was not supported for document related data sync
- Severe performance issues in the system



Tools

- MapBox, Java, Swift, Angular 8, Spring Boot, MongoDB, Spring Cloud, Auth0



Our Solutions

- The architecture was designed with a centralized API backend, with all features integrating with it in a seamless manner
- Leveraged Auth0 to create a multi-tenancy structure based on the company profile attached to the logged-in user
- Created dynamic JSON files and a single JSON file based on points count, along with cluster loading of Geo JSON for loading map and points
- Dynamic geofencing based on number of points and boundary data load related to customer location, implemented in offline mode
- Data download accompanied with compression download frameworks made available across, using lazy loading techniques
- Multi-Threading by creating threads when plotting in map, within a cluster
- Mobile DB Data added after plotting the point in maps, to enhance performance