

ABSTRACT:

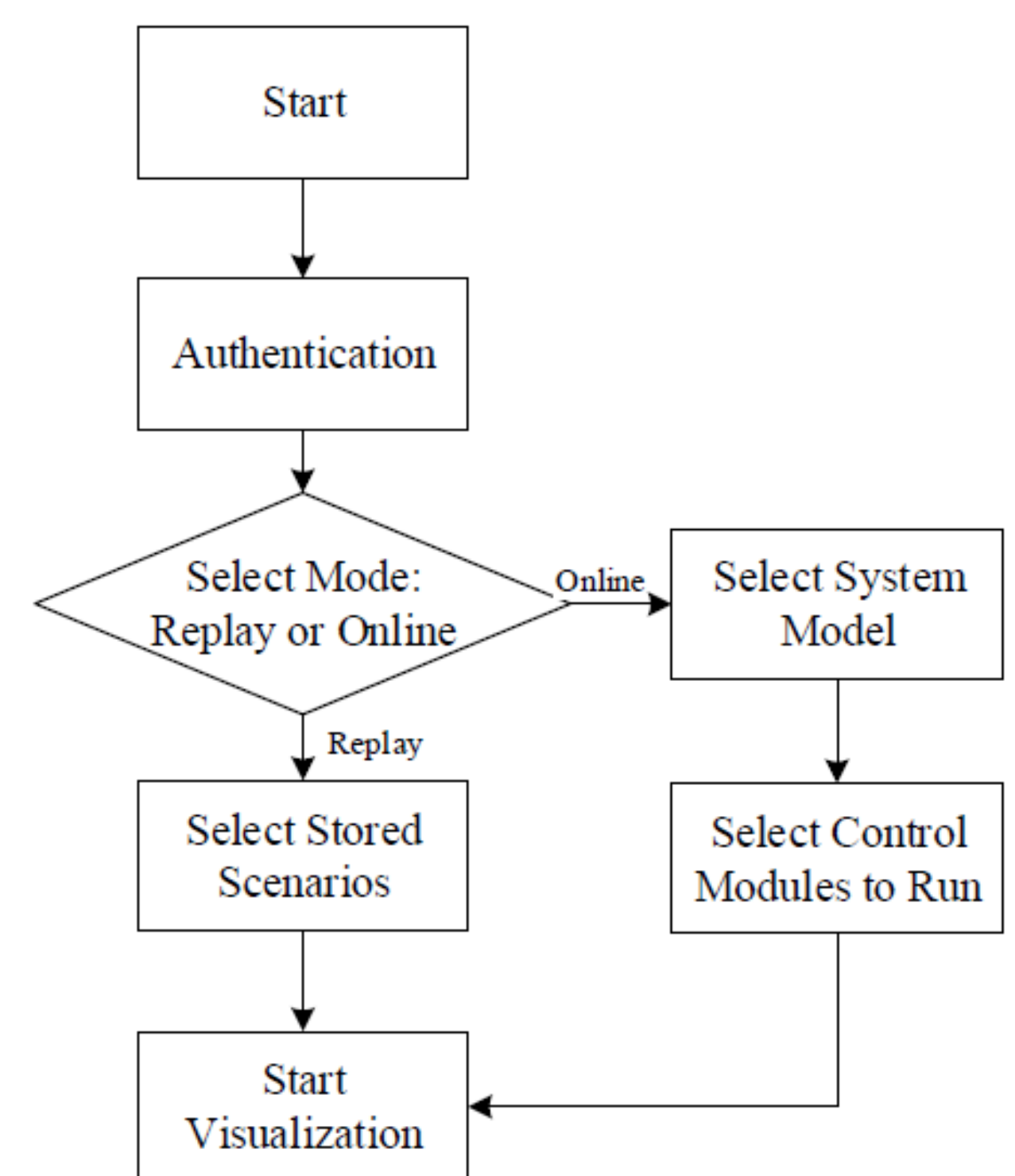
The LTB Visualization tool uses the DiME program previously created by Parker Diamond and Richard McEver to simulate large power grid networks. It then displays information about simulations in an understandable form. This is accomplished using a Django server to handle client requests and connections, spawn and destroy simulations, and save information to disk, and a browser-based visualization tool that relies almost entirely on JavaScript.

FRONT-END

The front-end for this visualization tool consists of a map view, a control panel, and a plotting system. The control panel uses Ajax requests to handle most of the visualization tool's communication with the back-end. The plotting system uses the Highcharts JavaScript library to display detailed information pertaining to user-selected simulation components in the form of custom line graphs. The map view displays basic information by coloring a canvas layer that sits on top of a map generated via the Leaflet JavaScript library.

BACK-END

There are two parts to the back-end, a module that handles data and metadata fetching, and a module that handles LTB simulations. Data fetching uses the python Bottle library's webserver to listen for requests from the web application. The client sends a request for metadata about a previously ran simulation, and then will request the data from that simulation. Simulation handling uses a Linux program called Systemd. Systemd acts as the operating system's initial program and will spawn all other startup programs. We use this to manage our scripts resources, cleaning up after the programs finish or fail. This system also allows us to have pseudo servers as we just have to make a request to Systemd and it will spawn the appropriate script instead of having them stay online. This group of programs start, stop, destroy, list, and communicate with the simulations.



LIFE-CYCLE

When a client connects to the webserver they will be prompted to log in to the application. Anyone can replay old simulation data that has been saved on disk, but only privileged users can start a simulation. If the client requests saved data then web application requests metadata on the previous simulation, processes this information, and then starts requesting actual data. The requested bus variable will be displayed on the map and all other variables are available for plotting. The user must create a new simulation configuration to begin a new simulation. Once this is done and the user starts the simulation, the web app will send the appropriate data to the backend to start the simulation. The backend will spawn DiME and the proper Matlab modules, and begin listening to DiME for the proper data. This all gets saved to a local cache and saved to disk.

