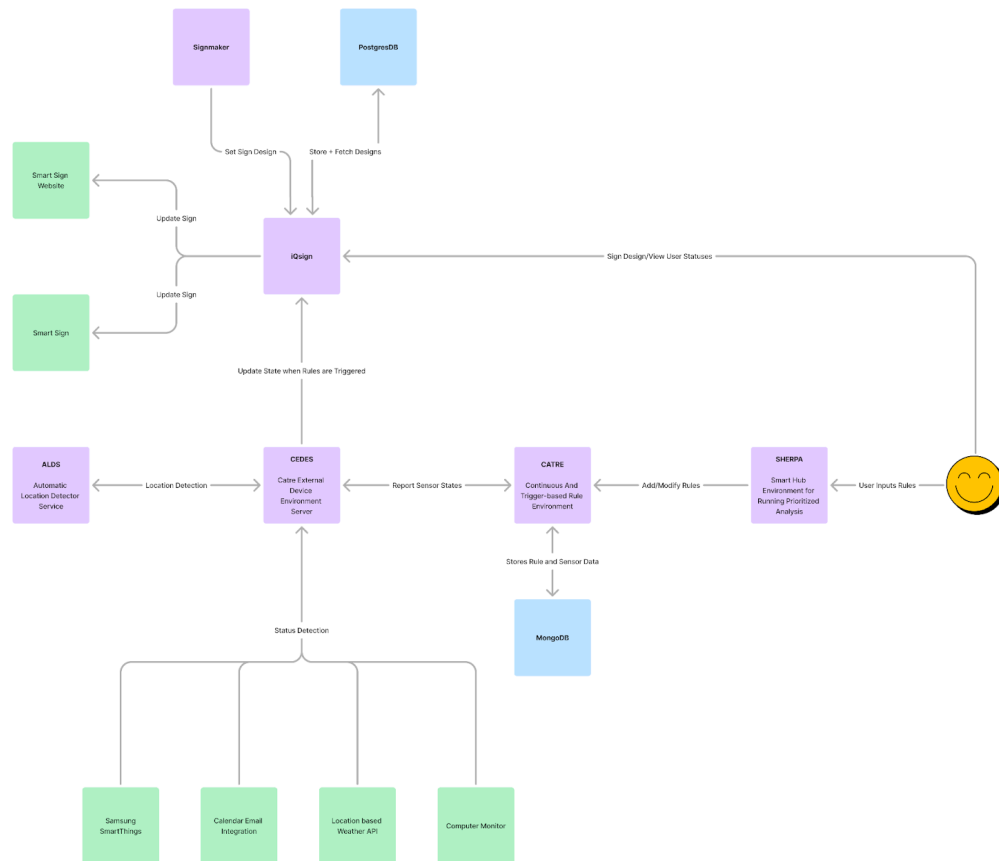


# SENSE UI STATEMENT

Our application is UI-centric, the UI drives the application and we are generally designing the UI first. We have three separate user-interfaces: Sherpa, IQSign, and ALDS. Sherpa's interface is finished, and is currently being left untouched.



## IQSign

The main goal of SENSE is to display the user's status based on sensors and rule-based triggers. The user-interface for IQSign is pivotal to the application. IQSign allows users to edit what signs are displayed, allowing for the final product to be more personalized towards the users and allowing every user self-expression.

Although the backend is already implemented, the UI team has decided to enhance the iQSign portal with a UI-first development approach. Our approach included identifying the existing operations, and gathering user feedback to determine the final design. We identified the following operations that should be available through the iQSign UI:

- Add new Sign Templates (Background)
- Update Sign Design
- Use Profile
- View Sign in different formats (web, image)

After identifying the main operations, we started building our UI prototype to support these operations. In upcoming sprints, we will update the necessary backend components to implement the new UI flows.

This front end will be coded with React.js and Javascript.

## ALDS

The Automatic Location Detector Service (ALDS) is part of the SENSE system centered around managing IoT devices using the CATRE rule engine. The ALDS UI pulls location data from a user's phone and communicates it to the CATRE backend via CEDES, which acts as a bridge for various external IoT devices and systems. CATRE, in turn, uses the location and status information to control devices based on user-defined rules, managed through the SHERPA interface.

Main ALDS Operations:

- **Location Tracking:** The ALDS app gathers location data from the phone using GPS and Bluetooth signals, which are processed to determine the user's current location.
- **Status Reporting:** ALDS also tracks the phone's current status (e.g., whether the user is on a phone call) and reports this to CATRE through CEDES.
- **Data Submission/Storage:** The app sends location and status updates to CATRE via CEDES, which processes this data as part of its rule-based system for controlling devices.
- **User Interaction:** Users can manually select or modify their current location in case automatic detection is insufficient, and the app allows for setting preferences related to location accuracy or manual adjustments.

The ALDS UI is built using Flutter, a popular framework for building cross-platform mobile applications. We will need to get things running and examine the current stage of things before we can confirm the UI. But the key components for the UI will likely include:

- **User Location Display:** A screen that shows the current detected location and provides an option for manual adjustment if needed.

- **User Status Display:** A simple interface to show the user's current phone status (e.g., on a call or not) and any other relevant device states.
- **Solid Integration with Backend:** The UI will need to be tightly integrated with the CEDES backend to handle sending location and status updates to CATRE in real time.
- **Periodic Device Updates:** Using background tasks like `Timer.periodic`, the app will periodically update location and device status in the background.