**Project Framework Document**

July 11, 2025

HING JOC Wildfire Detection Project

TABLE OF CONTENTS

[Juvare WEBOS 3](#_Toc1)

[Overview 4](#_Toc2)

[Proposed API 5](#_Toc3)

[BACKUP Framework 6](#_Toc4)

[Decision Process Workflow 7](#_Toc5)

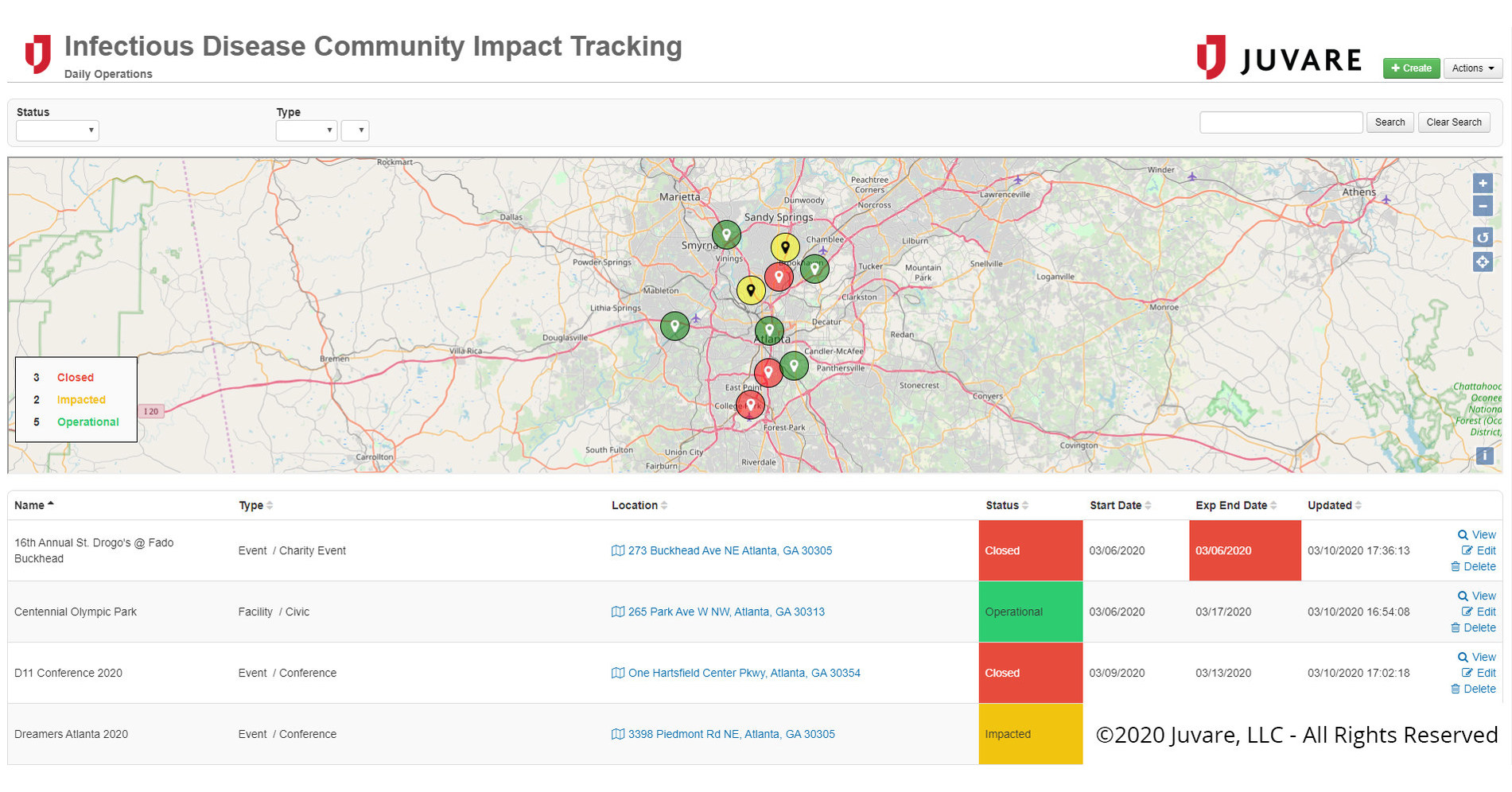
[Other Resources 9](#_Toc6)

# Juvare WebEOC



WebEOC is a web-based software solution that encompasses emergency, state, military, and federal agencies and departments for emergency notifications, event tracking, and incident management systems.

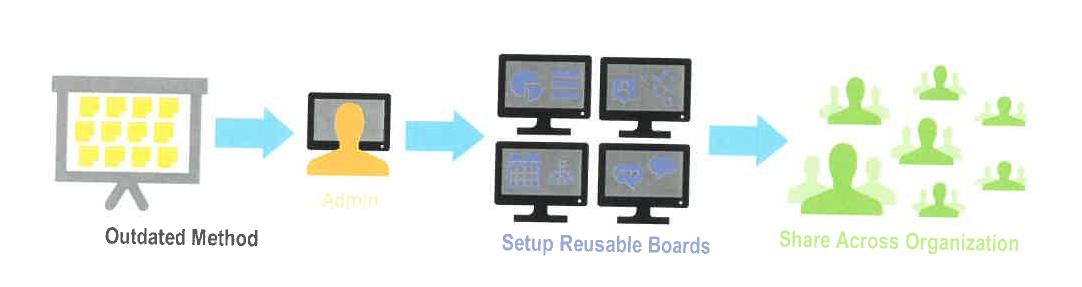
WebEOC contains multiple interlocking APIs that pull data from supported state and government agencies that feed that data into the system. Most of these data are locked behind request-only and government authorization. Due to time constraints from research and development and the additional required datasets and feeds, it would be recommended to develop a system under WEBEOS. Some examples of the supported APIs are ARCGIS, NASA’s FIRMs. Which are either limited by a threshold usage count or limited by an enterprise request.



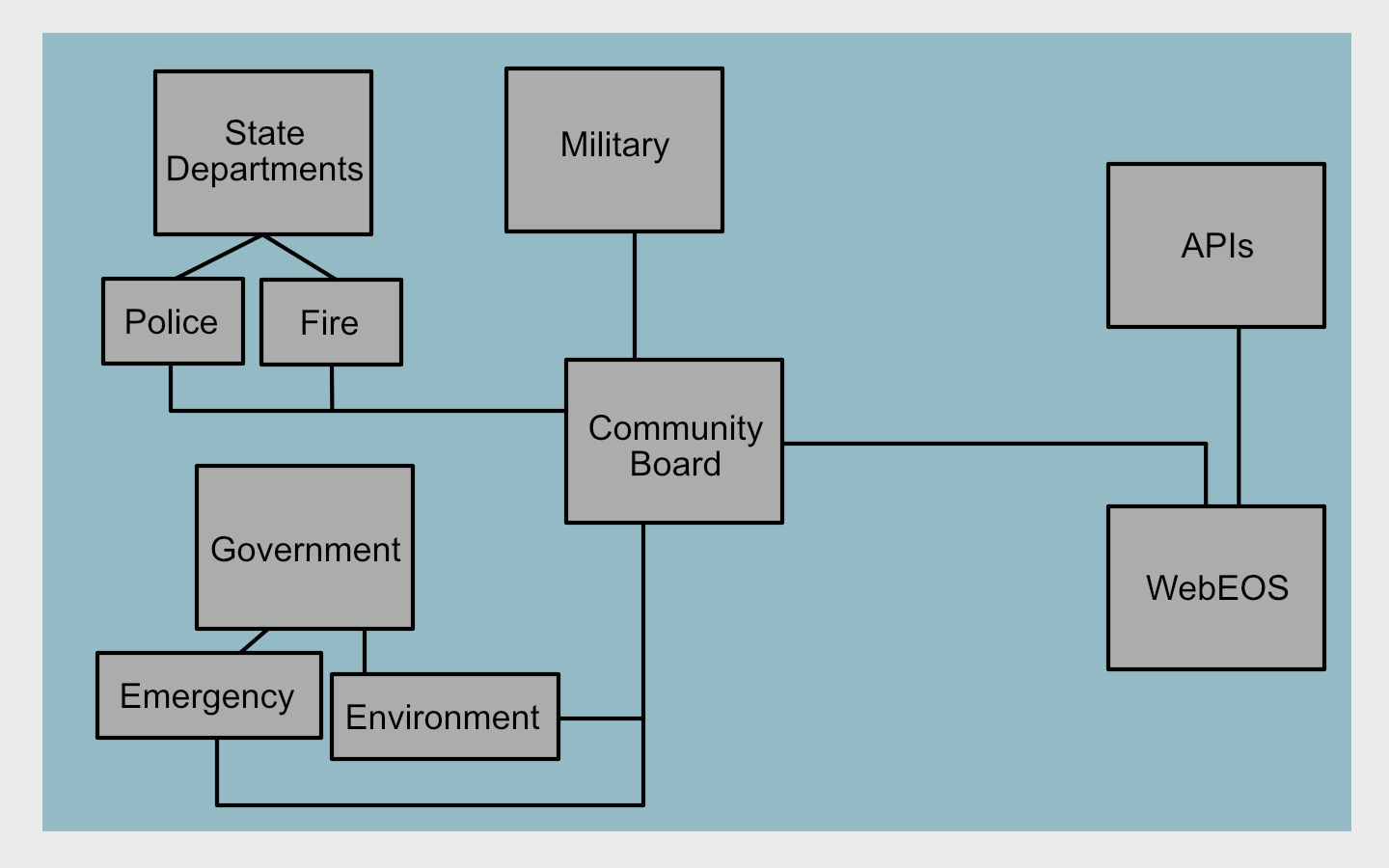
**UPDATE (7/11/25):** Note WebEOC is no longer considered due to additional software solutions under Juvare that are required in addition to WebEOC Nexus. Nexus does not have the capability for API integration besides a limited plug-in available.

# **Overview**

Data from the inputs are validated before being posted to the “board.”



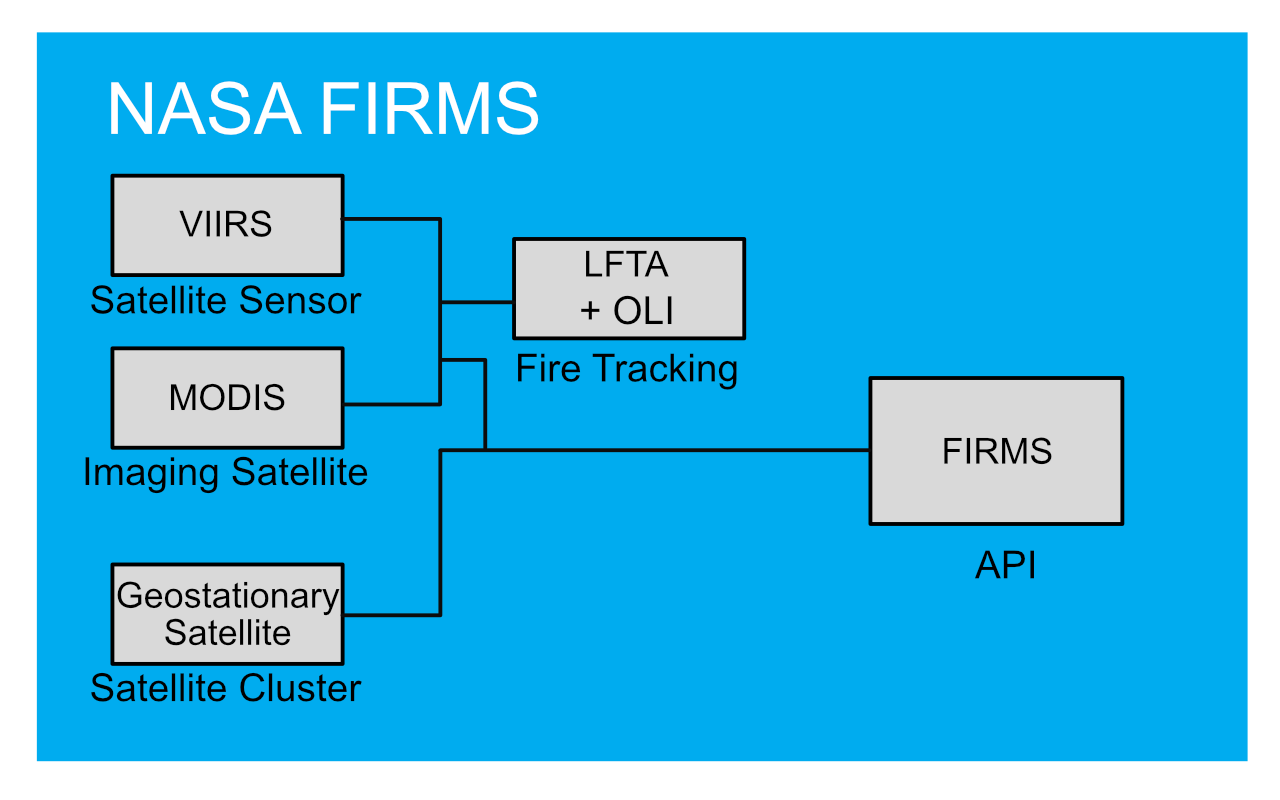
**Figure 1** Shared information streams between different departments federal, state, and military.



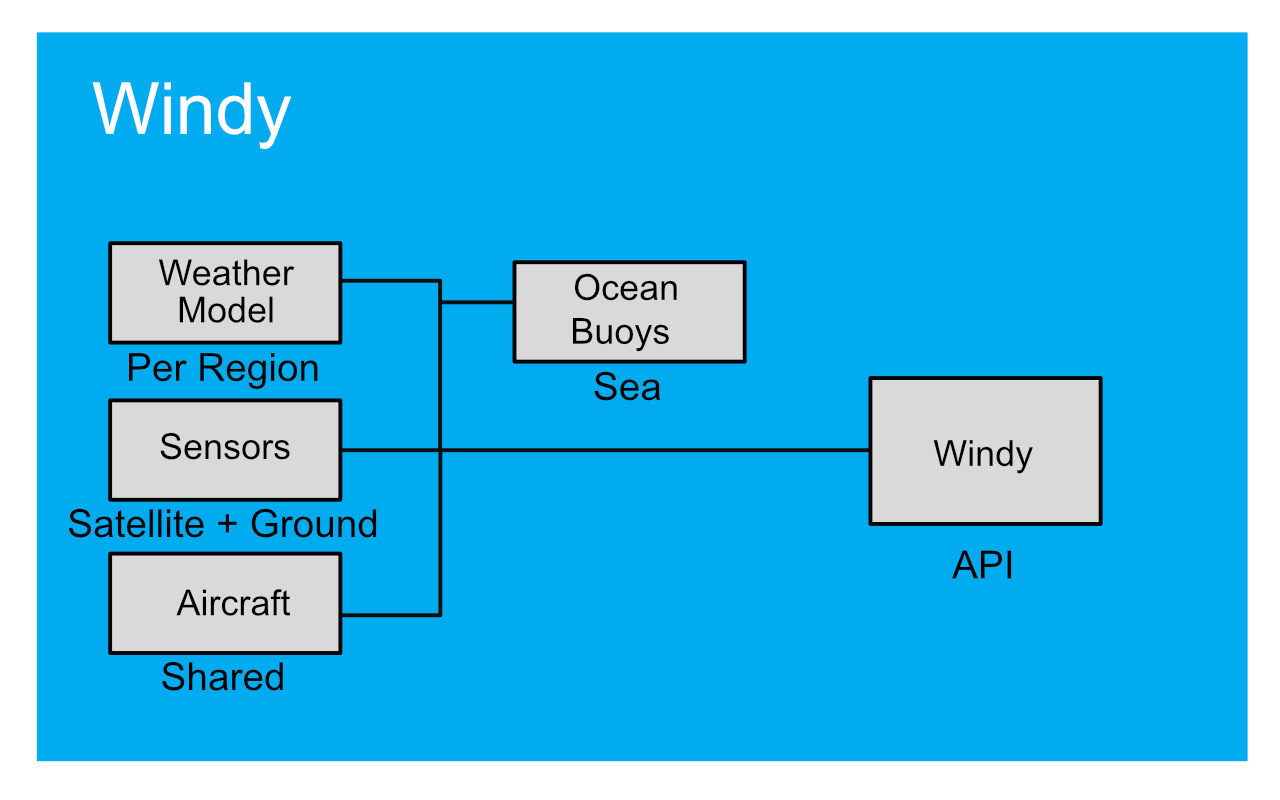
**Figure 2** API and managed information streams into WebEOS.

# Proposed API

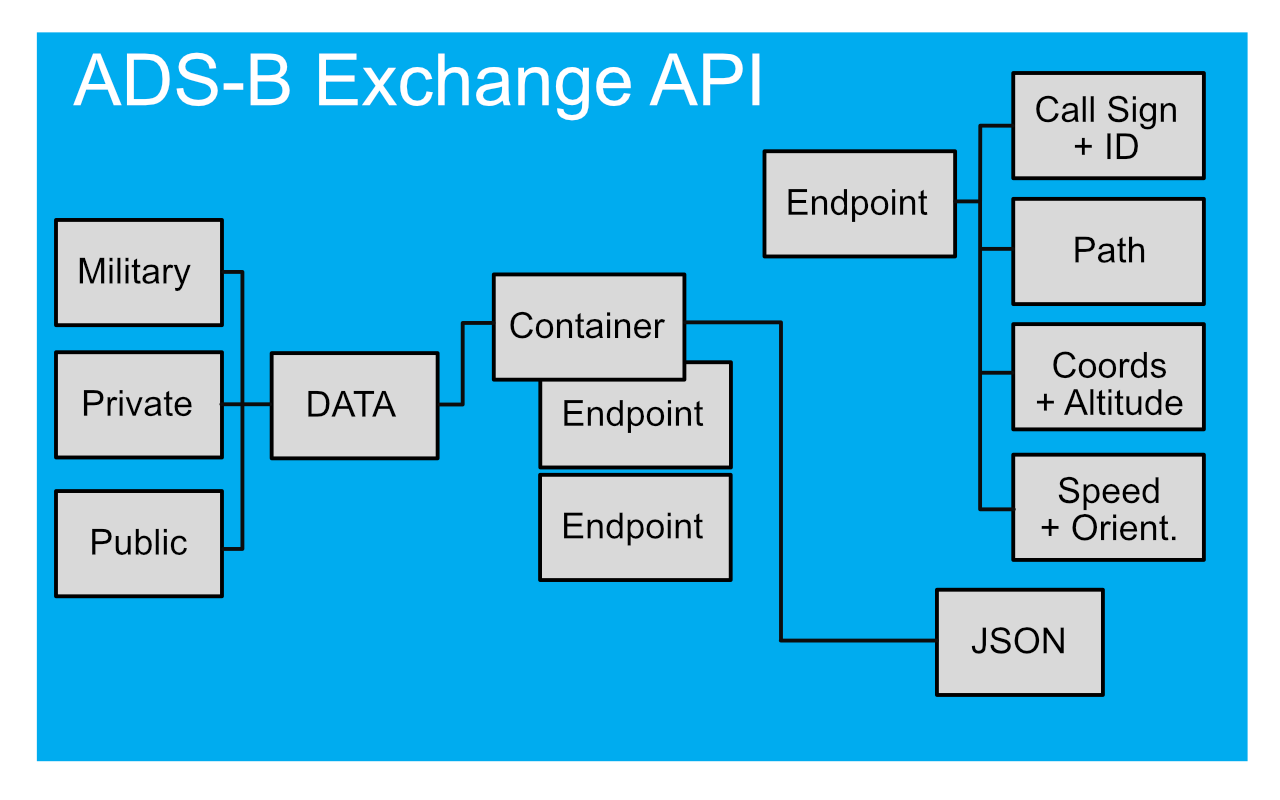
FIRMS API



Windy API



ADS-B Exchange API



# **BACKUP Framework**

The backup solution was considered due to the time constraints experienced throughout the planning process of the WebEOC route. After having access to Juvare’s WebEOC and the lack of integration with third-party APIs by default, we considered making the backup the main solution of consideration. While we did weeks-long research on input sources, we worked on a backup implementation on the side. When the team finally got access to WebEOC, many things did not work out. First off, there is the lack of integration for third-party APIs and uploading datasets from files. Many of the functions on the WebEOC solution required tedious time for data log entries. With no option to read from uploaded files, that process could be inefficient for the mission's objective and response time.

The backup solution is a web-based software page that allows access to a dashboard, login system, and user accounts. This solution would be mostly hosted on either federal computer systems or by the State. It is more likely the latter due to authorization followed by procedures and policies in place. The equipment must be able to host the web page, possibly store weather data, reports, and possibly contain sensitive data such as PII. If the web service is maintained such as renewing API tokens then the solution would not need much user training.

Upon loading the site, users will navigate to the login page which will require a username and password stored on a SQL database. The program used will encrypt user credentials. Users can navigate to the dashboard through an option at the top of the website banner. The dashboard will contain a map with filters and a time selection slider, Additionally, there will be one section containing scraped data from each of the county's fire and police departments. Another section will be a color-coded alert for fires. If implemented another section will be for nearby emergency aerial vehicles. The map will be interactive and will at first, show a cluster of points in a single point which will slowly divide once you zoom in. Each point is interactable and upon clicking will have an overlay for details such as fire coordinates, brightness mapping for fires, and location of the fire. Map filters and layers will contain other elements such as UV and Heat maps, wind speeds, satellite scans, and drought spots. One of the other layers important to the mission objective is the use of flight tracking.

Flight tracking will include all the aerial vehicles near the islands including helicopters, Each point includes details of its flight path, call sign, coordinates, and type of vehicle.

## **Decision Process Workflow**

Determine Data Sources

1. Determine what data sources are openly available to the public. Determine which needs government or federal access and which will need subscription or paid access.
2. Determine the file format of the data and if it can be incorporated during the analysis process before integration into code or programming.

Overseeing API Access

1. See if the API issues tokens for access. In addition, ascertain if documentation for the API is easy to understand and implement.
2. Check API key privileges, service limits, and usage boundaries.
3. Confirm with team if programming languages and libraries are compatible with the current code base.

Determine Security Controls and Risks

1. API Key Encryption and Abuse

Consider the abuse of the website and API keys if someone were to cause a denial of service (DoS) by a flood of service requests.

Note: Consider that someone could create a script that could make automated commands to pull from one source and use up the transaction limit.

2. Web Service Host

There is a consideration for the State to host the website and make it private rather than open to the public. This is due to sensitive data and availability of service.

Other options include hosting by Federal and third-parties. We were told this would be unlikely due to budget concerns and the policy and procedures for consideration for federal use and the security management processes for third-party hosting.

3. Encryption and Sensitive Data

Pulling data from Federal and Commercial providers can be sensitive and may have some legal regulations on storage practices and usage. It is advised that API keys must be protected. This can be achieved with encryption algorithms that use hashing techniques, which can also be layered with salts for complexity. In connection with this, this can apply to user credentials which will require the same principle. Depending on the acceptance of this project moving forward, data could be saved on the hosting systems which could require some form of encryption as well. Otherwise, checks on how the data is secured while in transit to the project would be required if not locally stored

Scope and Improvements

After getting the base features of the dashboard, decide what things could be considered out of scope such as the range of map points (global vs local). In addition to utilizing additional scripts for the closest transport vehicles and filtering of alerts based on fire data. In addition, APIs that were originally considered but dropped due to a lack of access, paid subscriptions, and government requests. This also coincides with the limitations of the interoperability of the different coding languages and libraries used.

Recommendations

Converse with higher management about rules of engagement and options available. Request that these tools not be readily available due to authorization concerns, long access wait times, technical problems, learning environment, and update them about expectations.

For access to additional State and Federal add it is essential that HIEMA gives access to reports and city data.

It is recommended that these topics be covered before starting the project:

* Set budget
* Authorization and access to systems and datasets
* Timelines and Deadlines
* A person of contact (POC)
* Set expectations of what can't be done

**Leaflet**

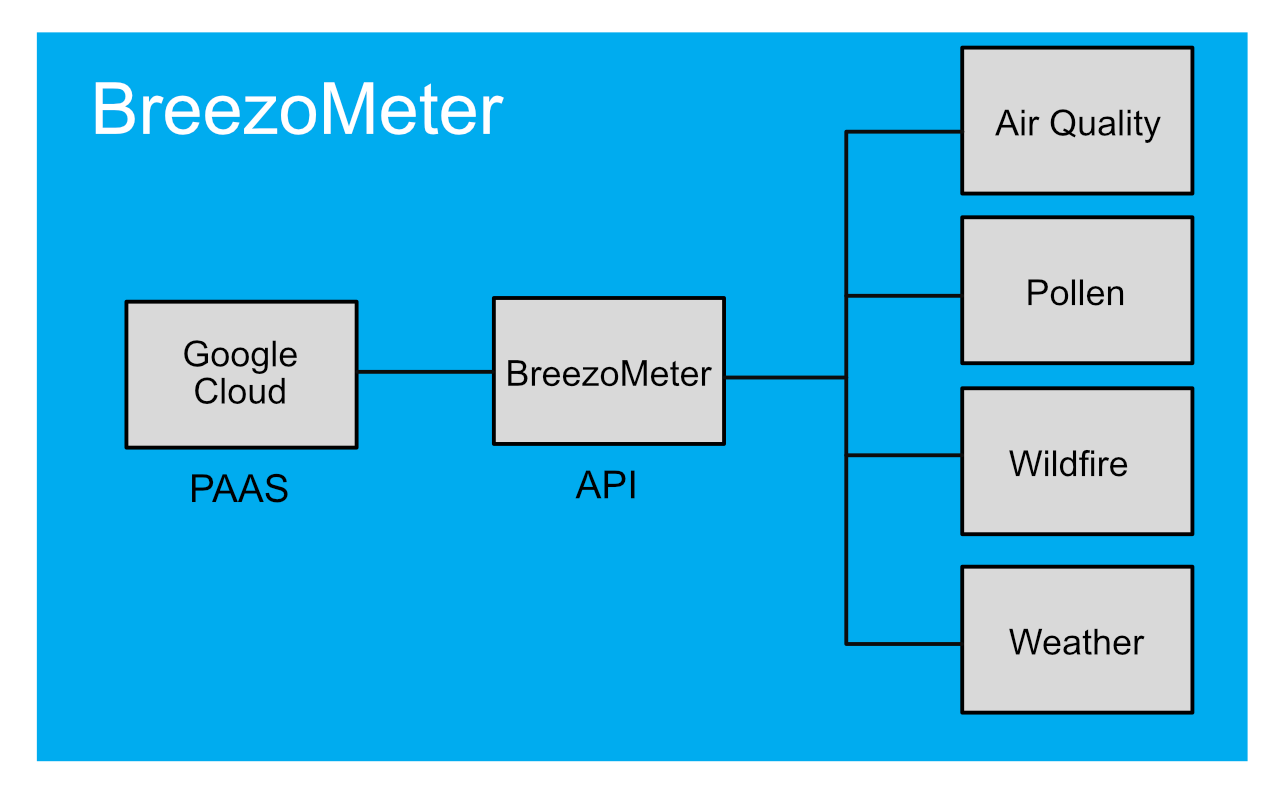
See whether the data can be important and understood depending on what attributes each data point has.

Decide on the default presentation. Zoom in, filtered points (not global), clustered points, and information display on overlay. Decide if the map starts in flat mode or satellite mode.

# **Other Resources**

APIs No Longer In Use:

BreezoMeter API



FlightRadar24 API

