WeatherCitizen

A Software Suite for the Collection, Assimilation, and Distribution of Traditional and Crowd Sourced Environmental Observations

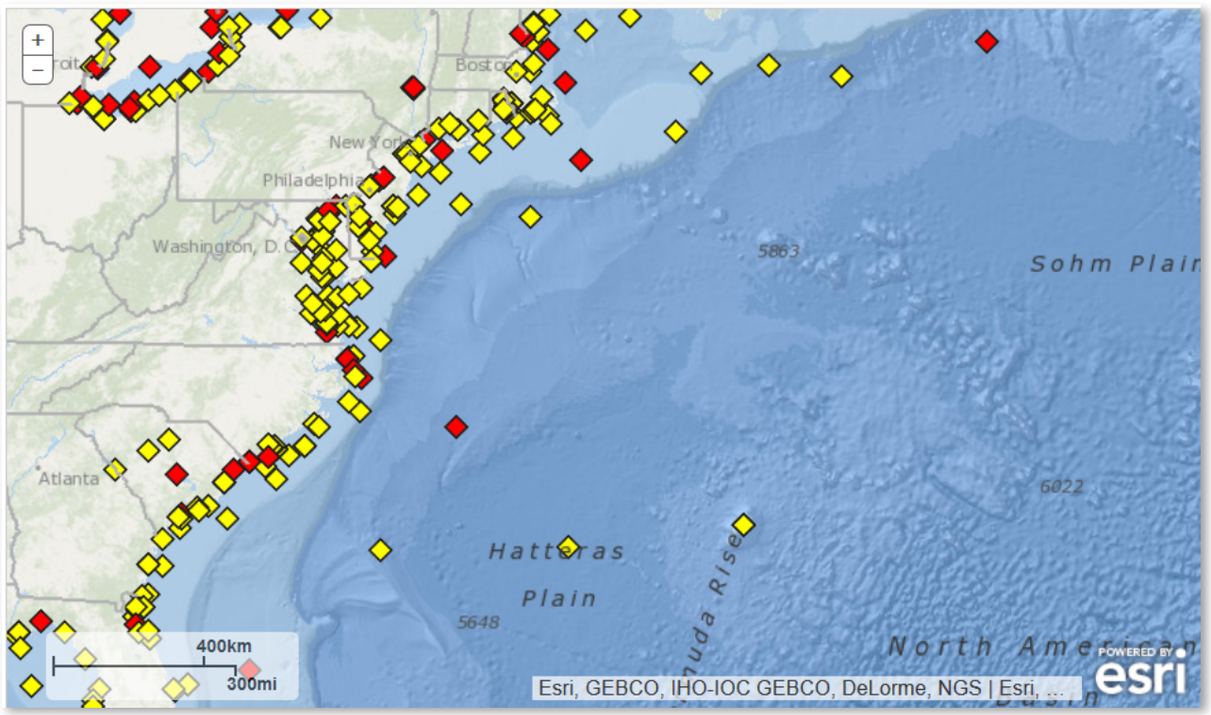
Marc Shapiro, Jerry Bieszczad, David Callender
Creare, Hanover, NH

8 January 2019

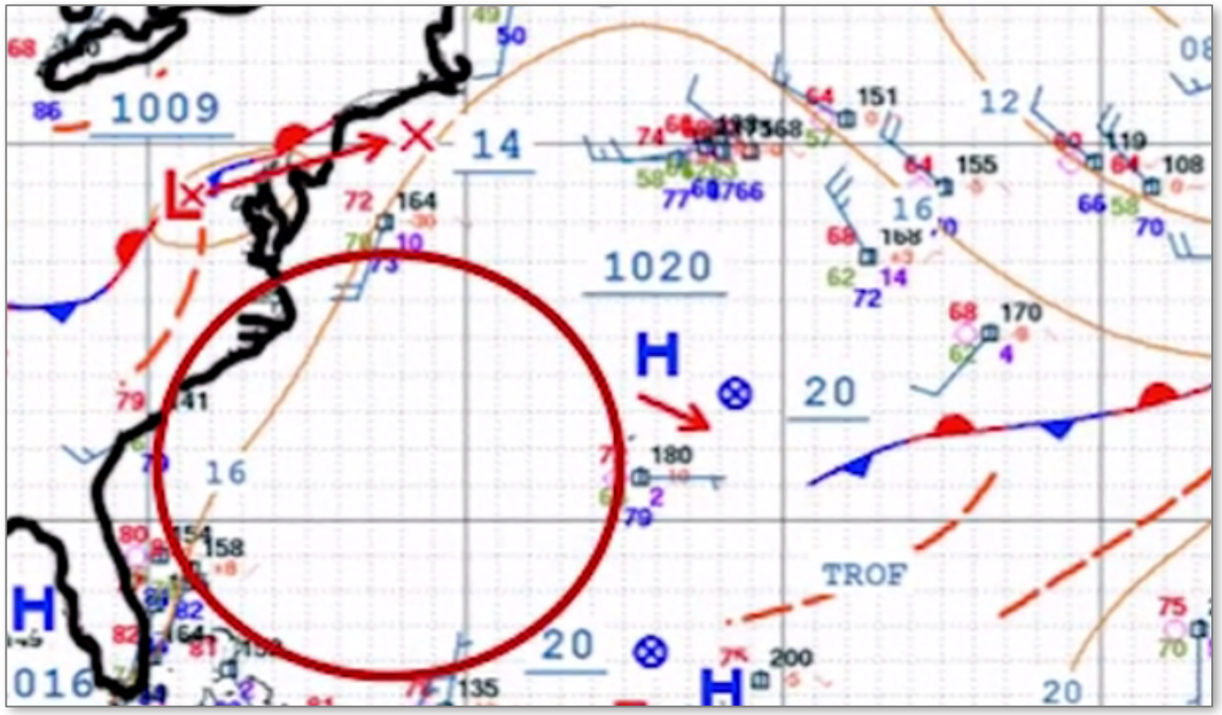
Creare

Motivation

- **Marine weather** "nowcasts" and forecasts are **critical** to maintain **situational awareness** and ensure **safe navigation**
- Existing **marine** weather **observations** are **sparse**



Buoy Map from NOAA NDBC



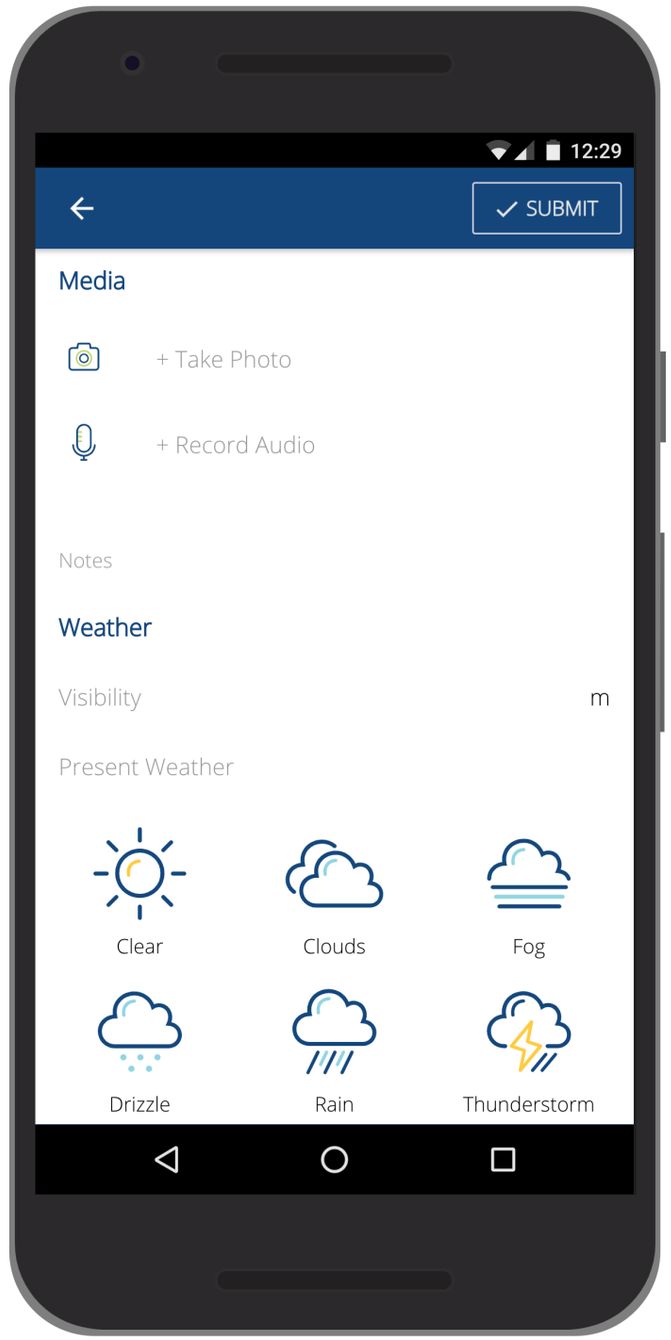
Example Map from VOS Observations

Smartphones for Crowd Sourcing

- **Smartphones** support multi-modal **environmental sensing**
  - **Observations**: UI, Camera, Microphone
  - **Sensors:** Pressure, Temperature, Light, ...
  - **Location**: GPS, Network

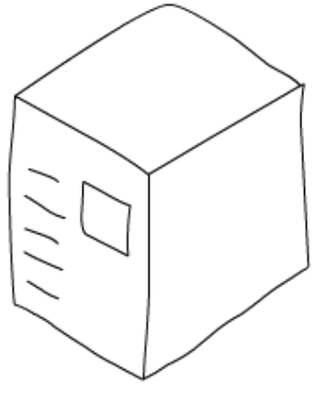
- **Project Goals**:
  - **Enable** crowd-sourced weather **observations**
  - **Disseminate observations** in real time
  - **Promote** data **access** and **control**

Advance development of data products and insights using crowd-sourced data

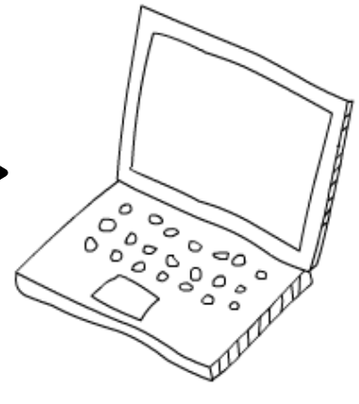


Crowd Sourcing Challenges

Data aggregation infrastructure



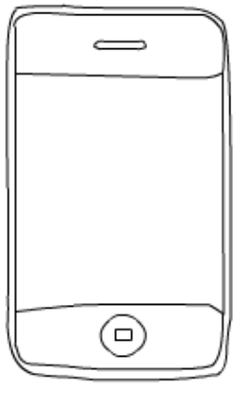
Quality control and data validation



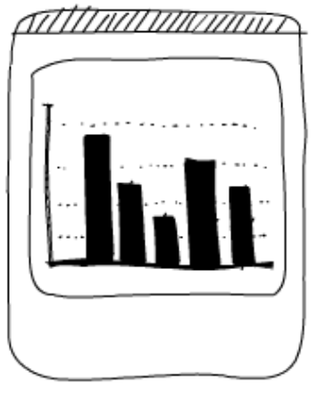
Data access, control, privacy, interoperability

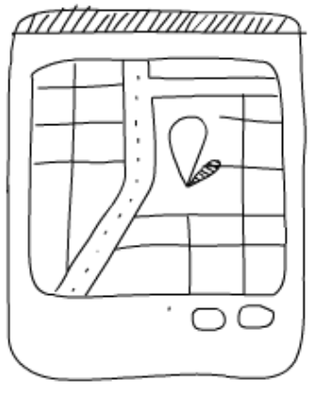
Incentivizing data collection

Curating Inputs



Data wrangling and harmonization





Products

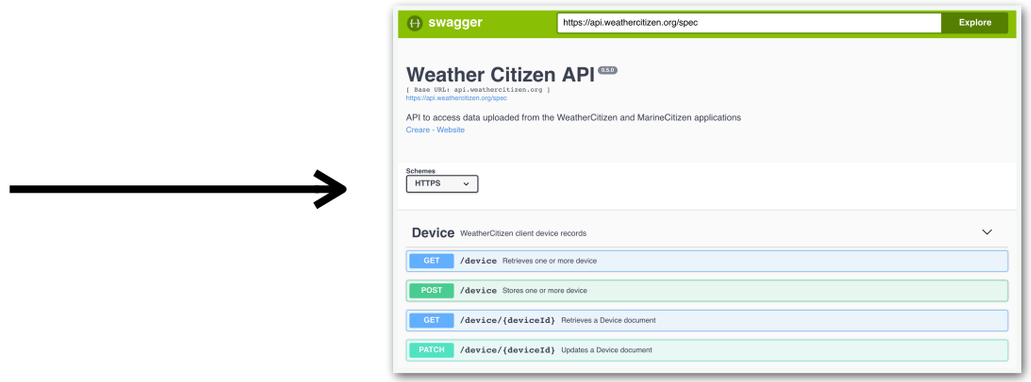
WeatherCitizen

Modular full-stack platform to support weather-related crowd-sourcing

REST Server with Geospatial Database

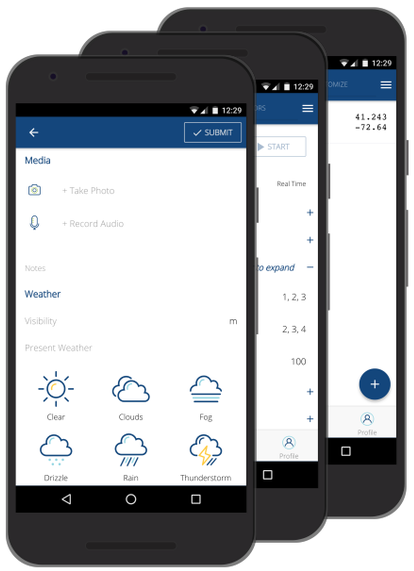
Eve.

mongoDB

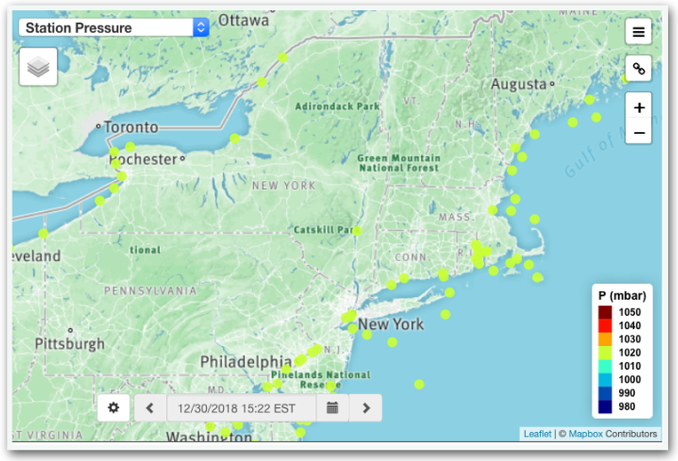


OpenAPI for Data Access

Cross Platform Mobile Application for Data Collection

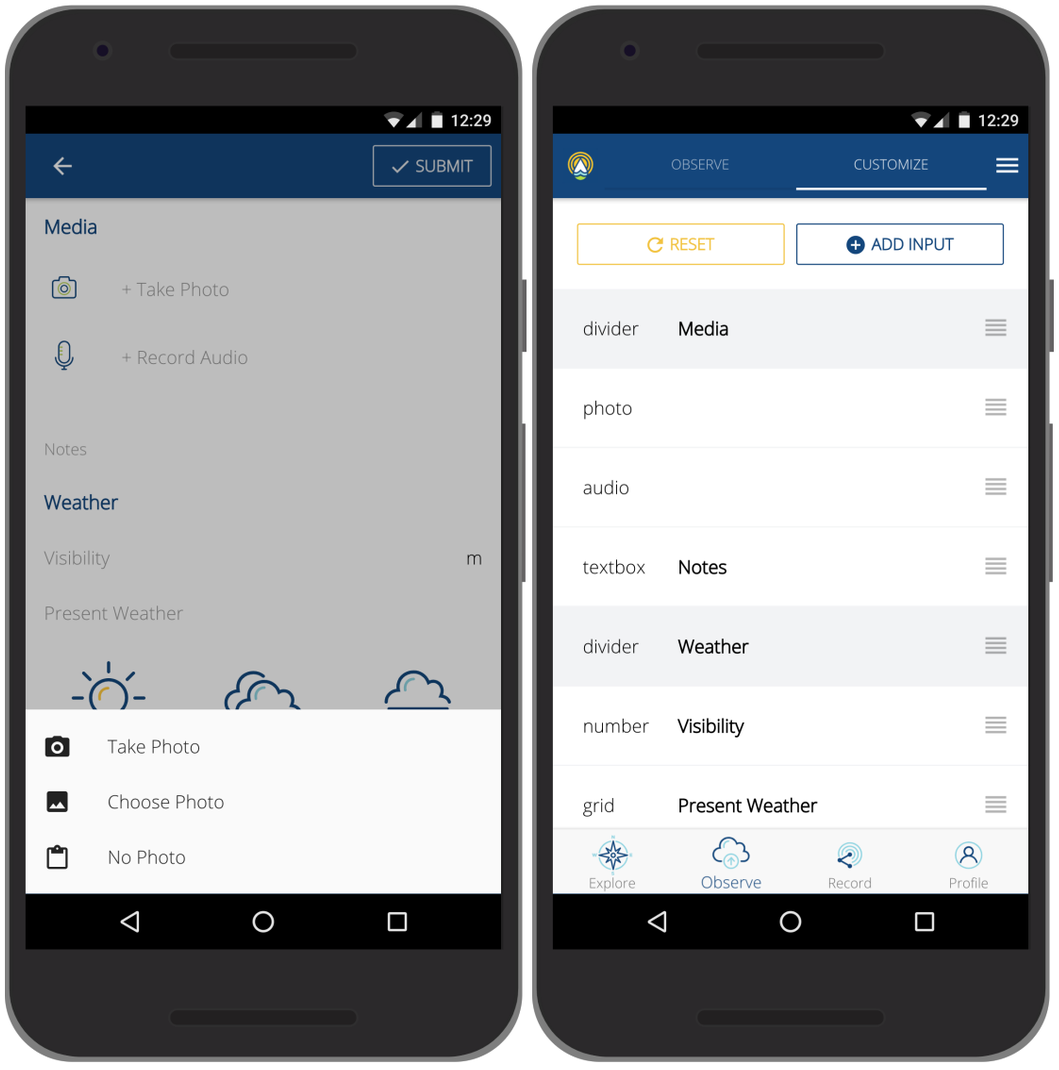


Data Processing Libraries (MATLAB, Python)



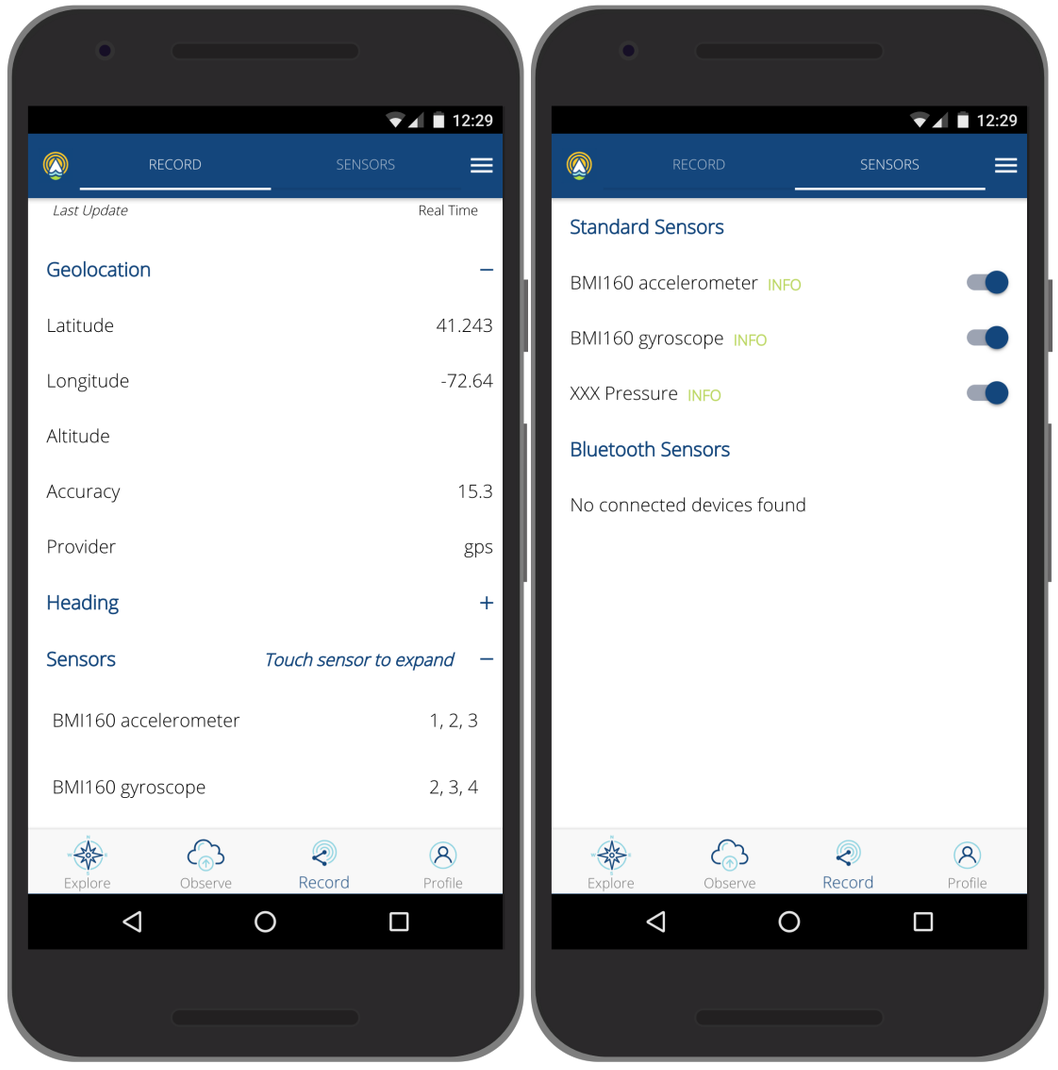
Webmap Visualization

WeatherCitizen App



Collect Input Observations

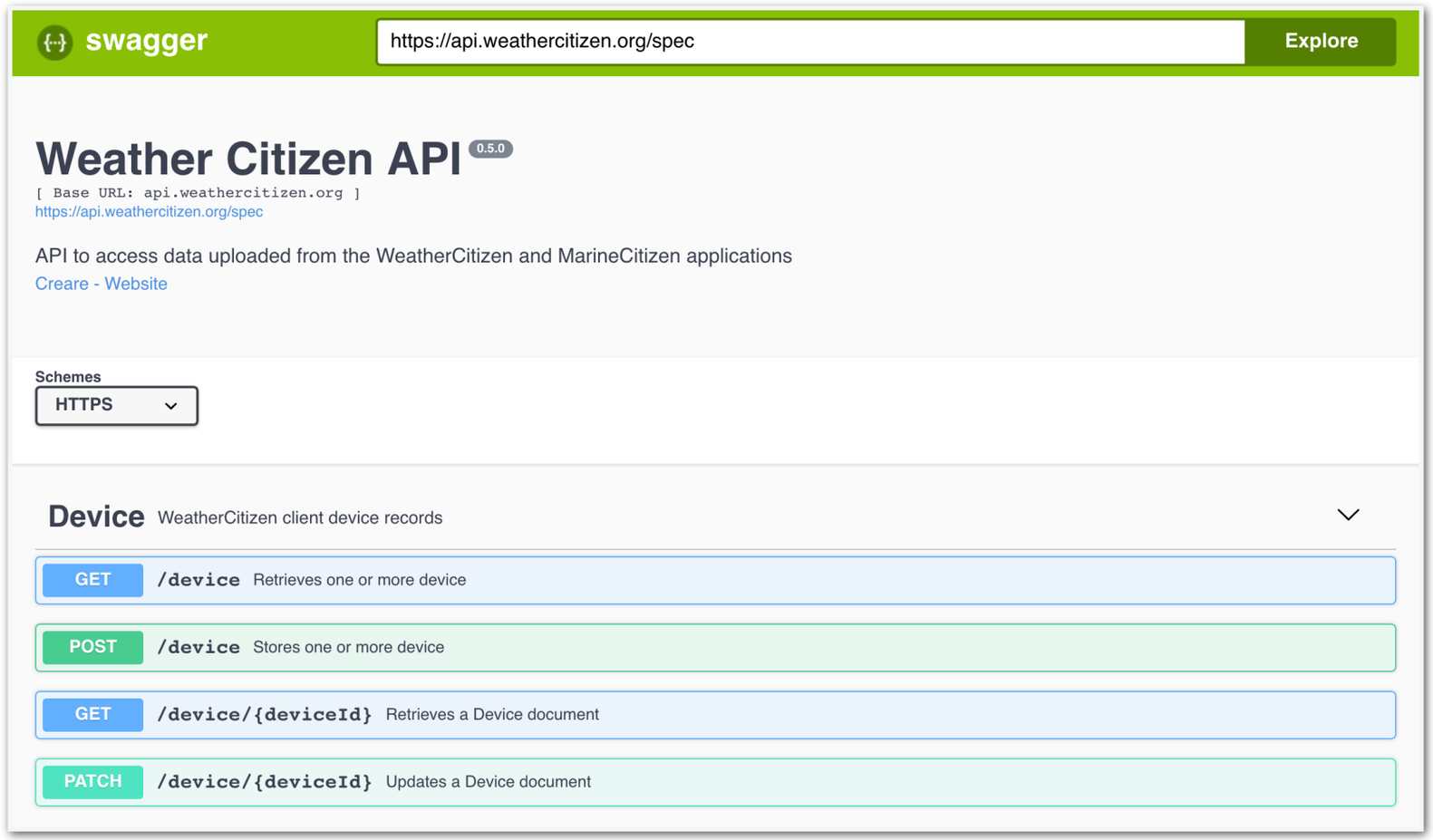
- Input **image**, **audio**, or **data**
- **Customize input fields** to support additional observations



Record device sensors

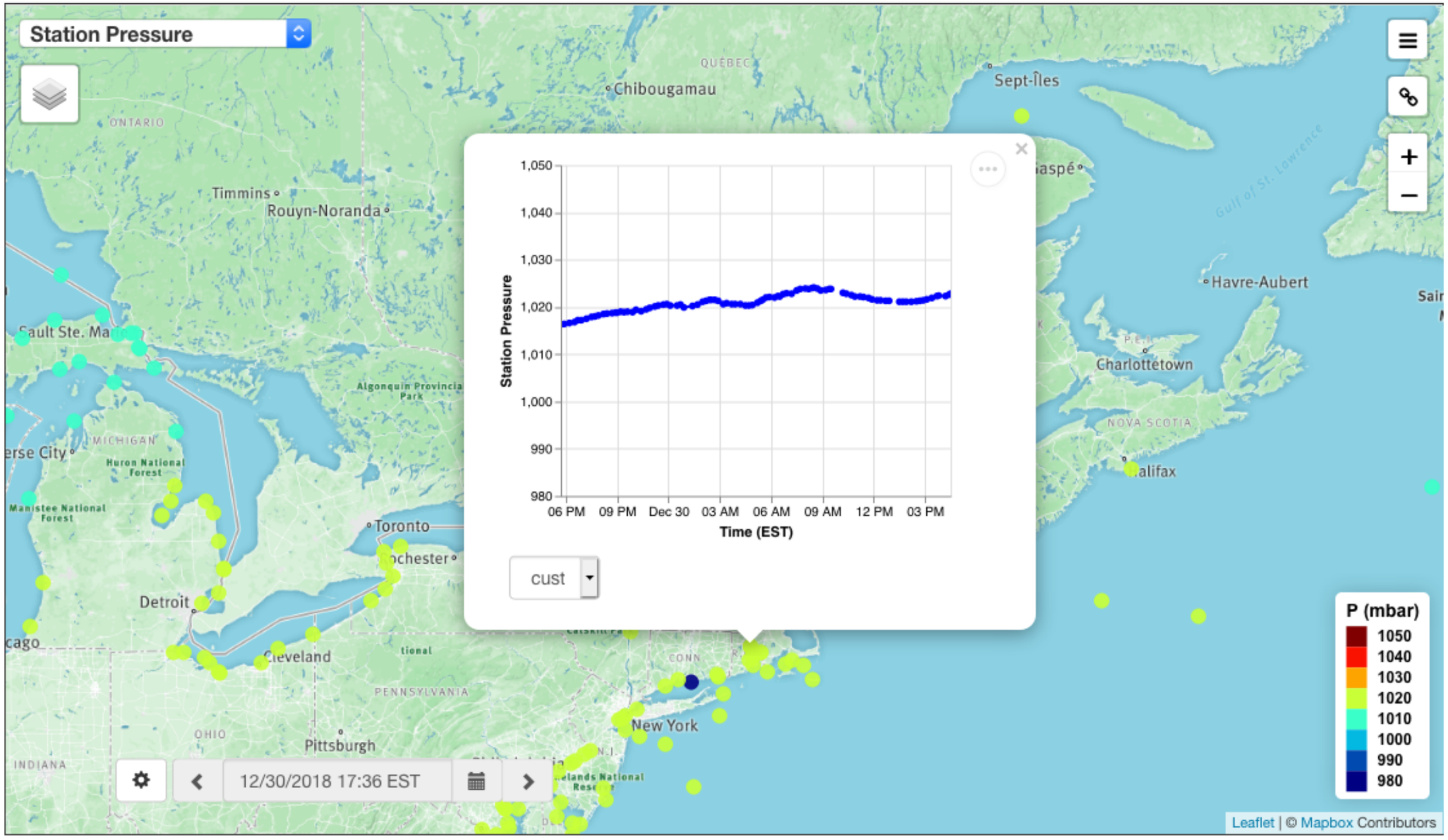
- Record **device sensors** down to 100 ms frequency
- Connect to external **BLE sensors** (i.e. Kestrel)

WeatherCitizen Server



- **Portable HTTP** based server architecture
- **REST API** (OpenAPI) for data upload/download (Python Eve)
- **GeoJSON** formatted database (MongoDB)

WeatherCitizen Tools



- **LeafletJS** webmap visualization to explore datasets
- **View** or **Download** historical data as geojson, csv

WeatherCitizen Tools

Get Data

```
[ ]: # load the wzserver module
from wzserver import tools
```

```
[ ]: # get all devices
devices = tools.get(collection='device')
print(len(devices))
```

```
[ ]: # get sensors
items = tools.get(collection='geosensors', start='2018-05-20', end='2018-05-22', box=[[-72,43]
print(len(items))
```

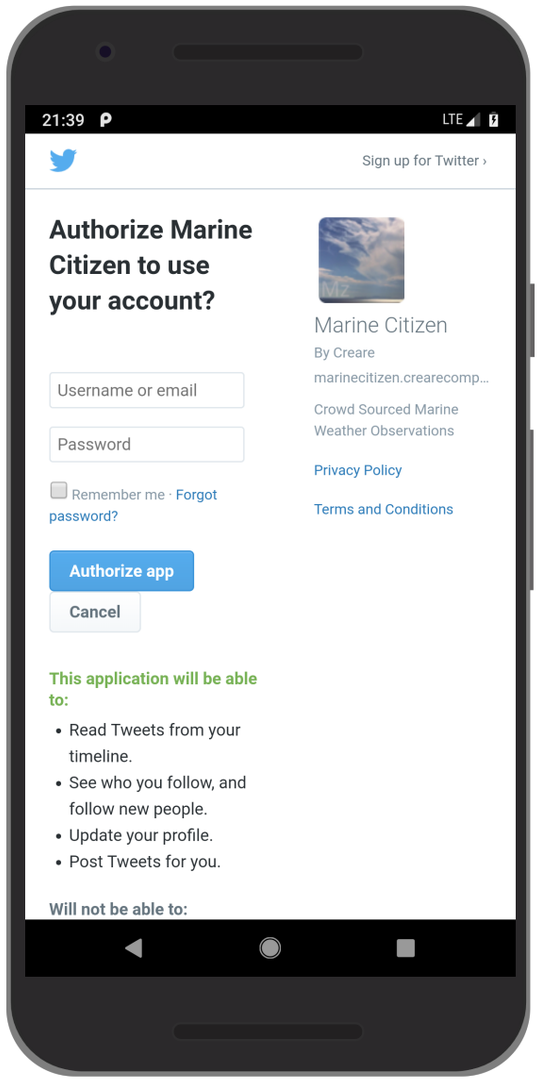
```
[ ]: # get ble devices
items = tools.get(collection=['sensors', 'geosensors'], start='2018-06-06', query={'properties
print(len(items))
```

- **Python module** to facilitate access and manipulation of data
- **In Progress:** Integration with PODPAC geospatial analysis library

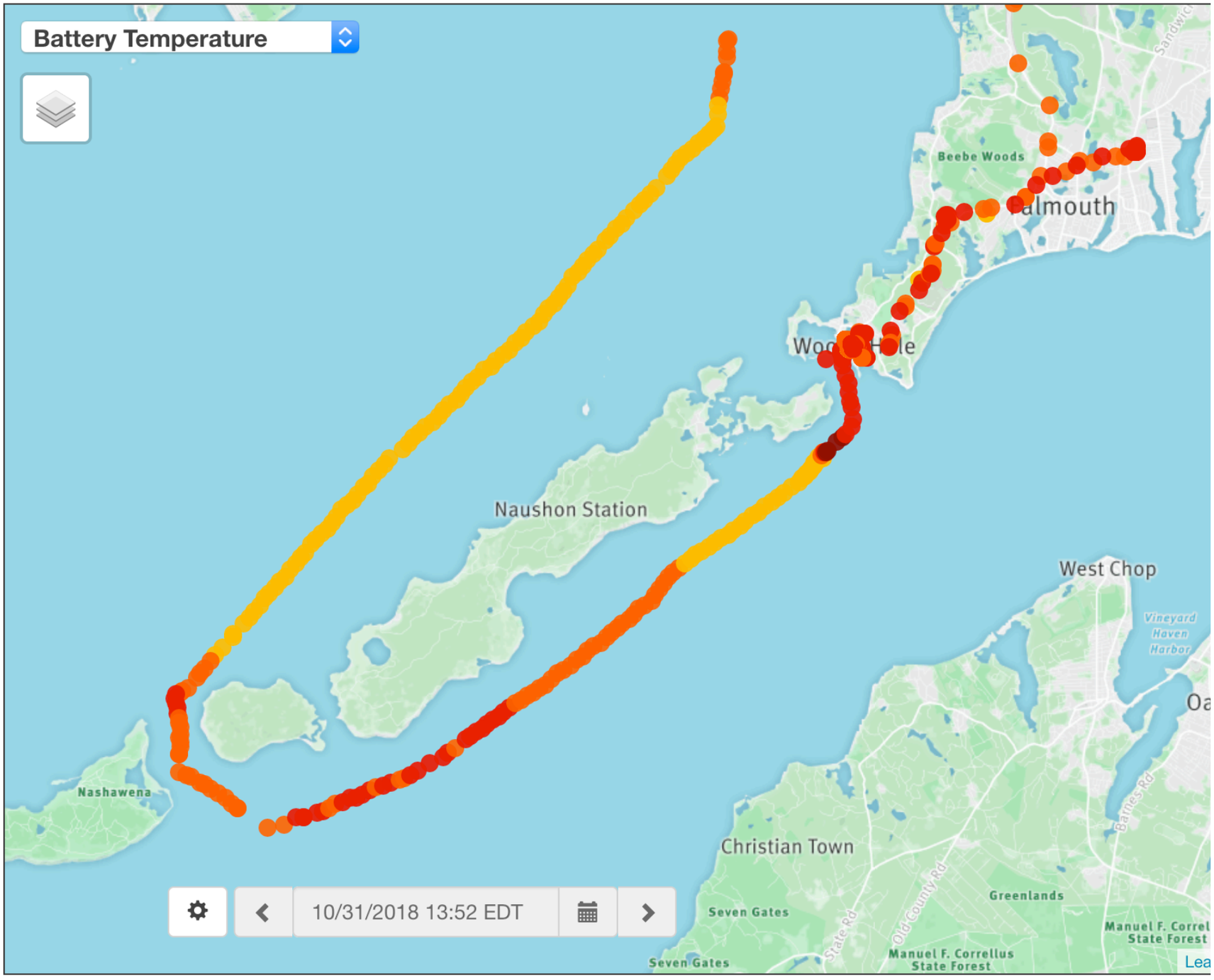
Pipeline for **O**bservational **D**ata
Processing, **A**nalysis, and **C**ollaboration

Additional Capabilities

- **Export** app data directly to **.json / .csv** (skip server)
- Push manual inputs to **Twitter** with specific hashtag
- **In Progress:** View nearby public observations in app



Example: Marine Weather



Research vessel collecting daily sea state statistics

Example: Marine Weather

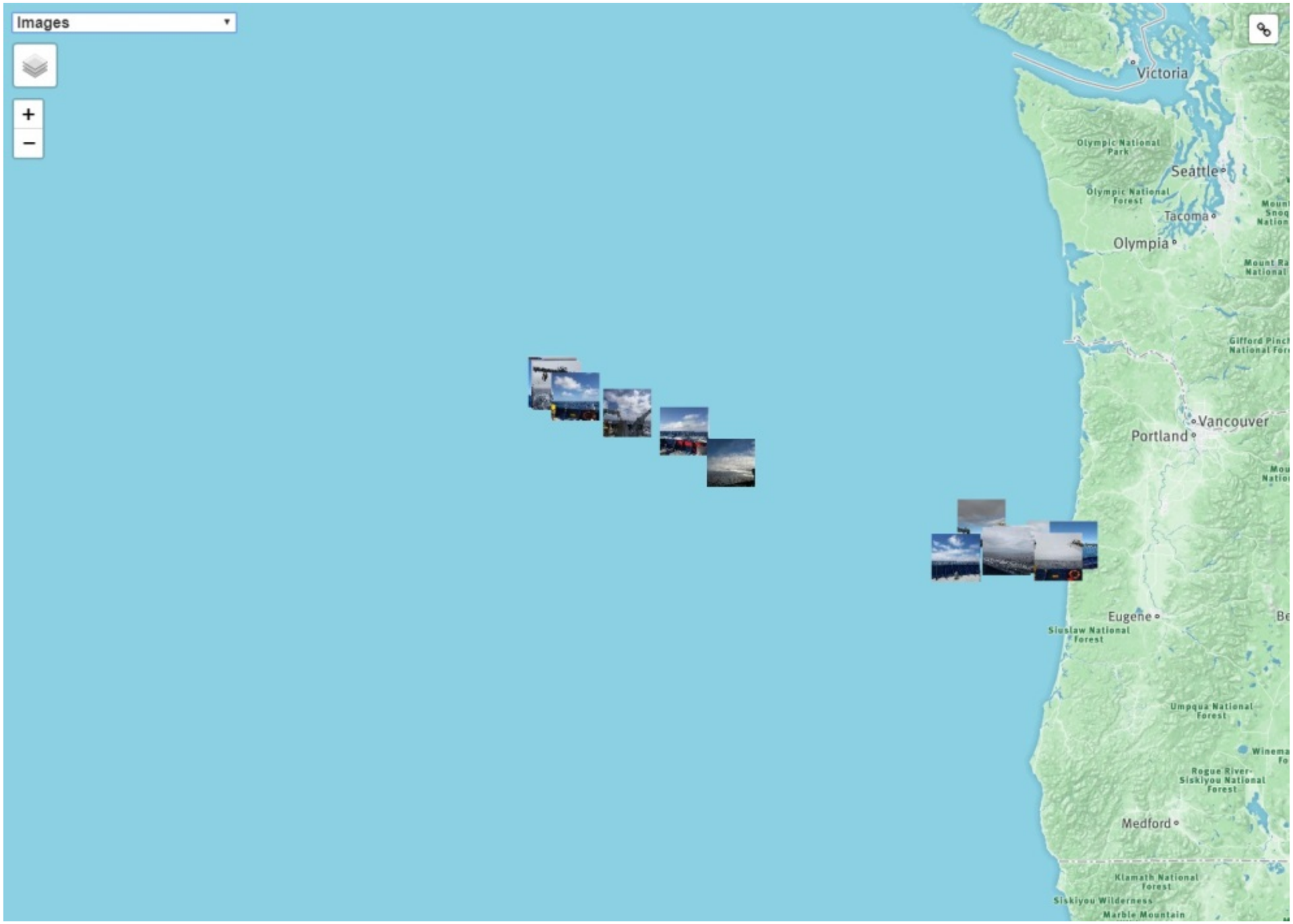
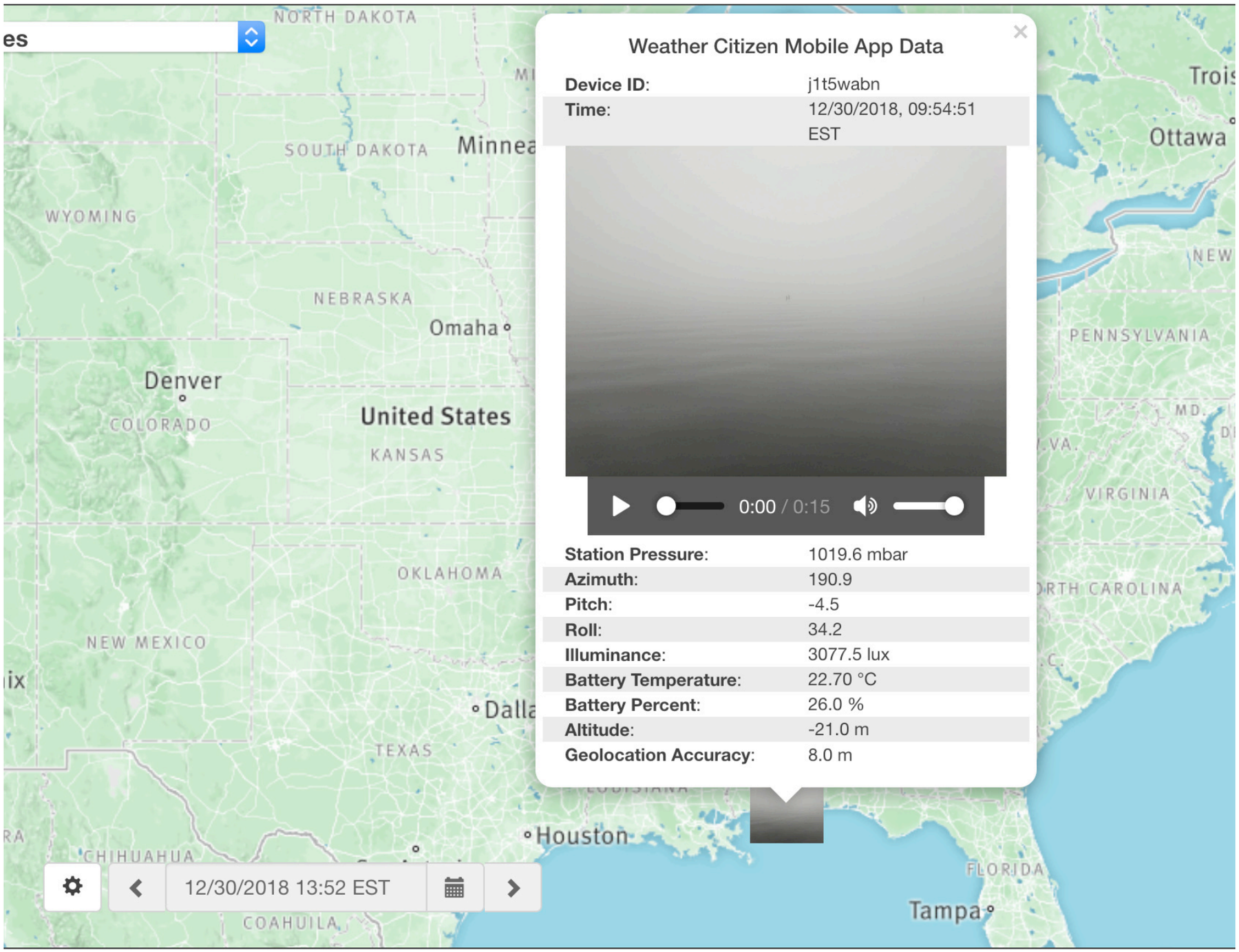


Image log of research cruise in Pacific Northwest

Example: Marine Weather

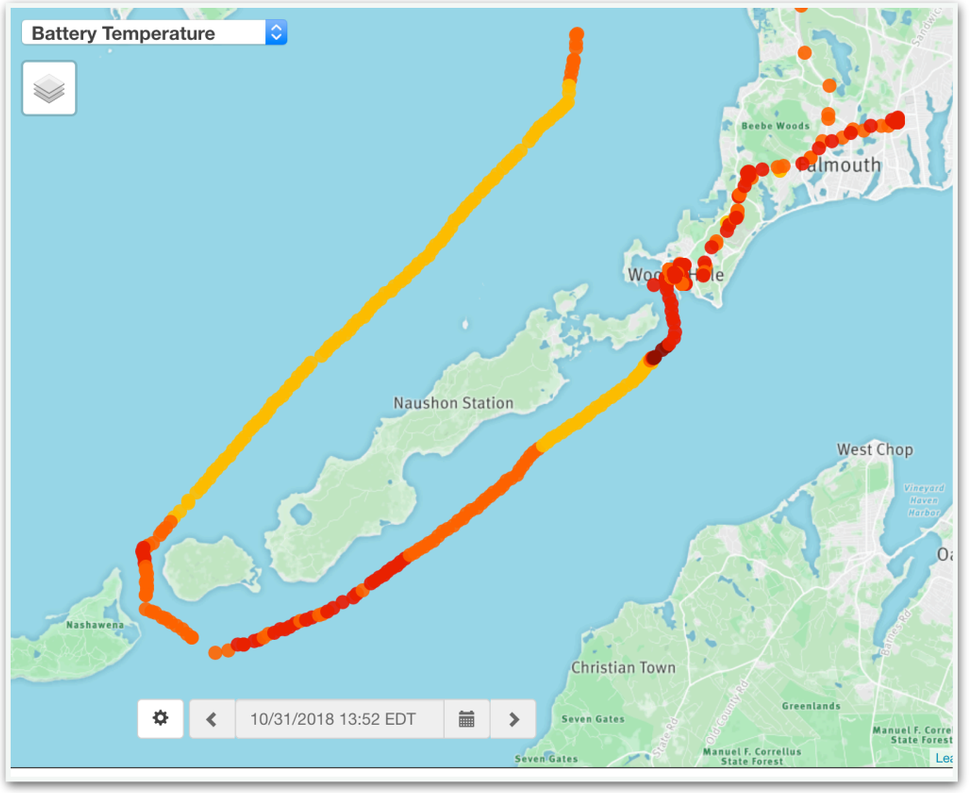


Oyster farmer tracking the salinity of oyster bed

Workflows

- **Manual Observations**
  - Hourly / Daily observations
  - Image, audio, custom input observation
  - App provides notification reminders

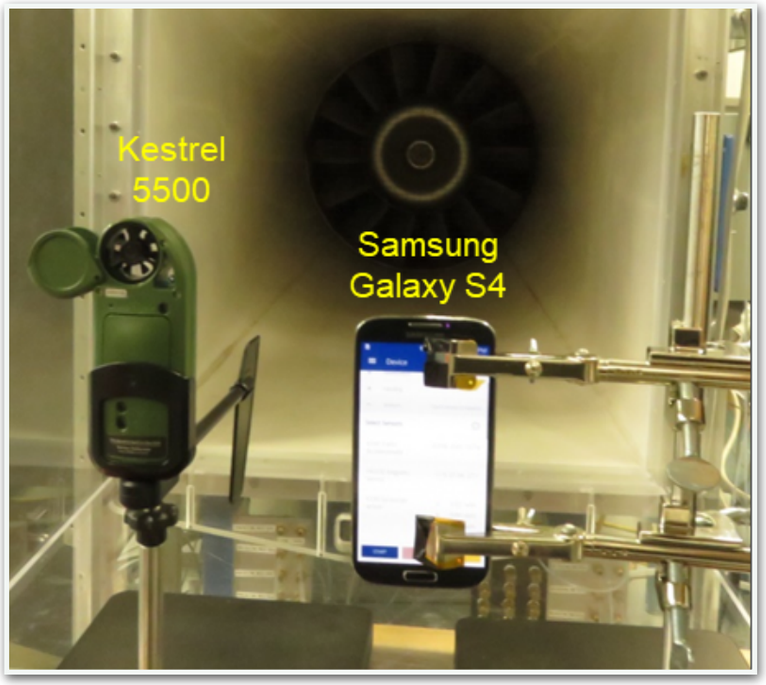
- **Automatic Device Recording**
  - "Weather Station" mode records sensor values every ~1 minute
  - Each data point includes burst of sensor values over 5 seconds

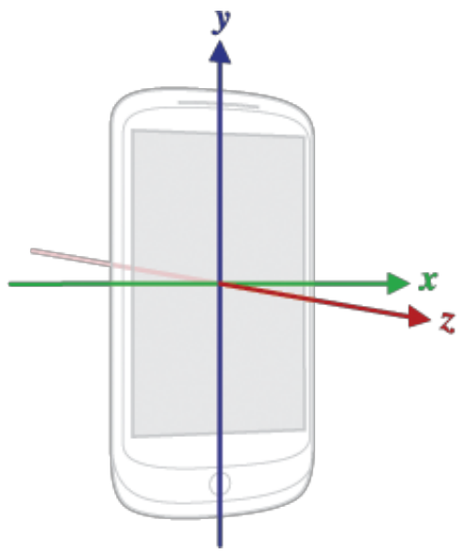


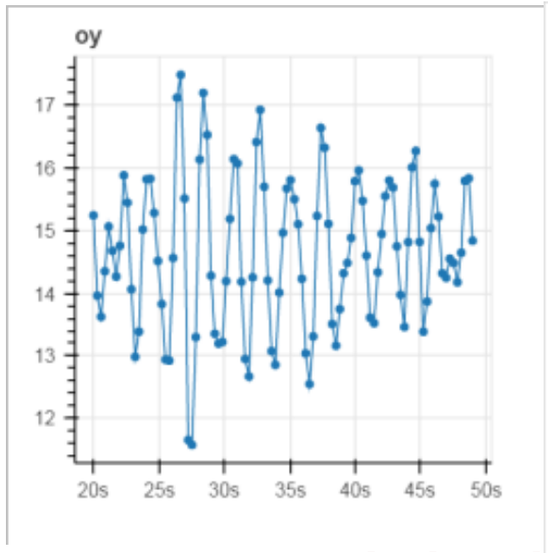
Processing and Filtering

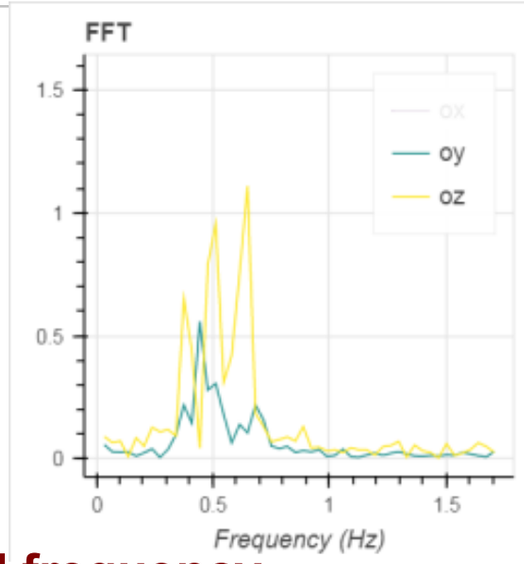
- Data records stored as raw **GeoJSON**
  - Interoperable with **most GIS platforms**
- Sensor "bursts" stored in compressed **Protobuf** format
- Server supports geospatial queries through API
- Python server and tools support **hooks and filter functions**
  - Enables **dynamic validation** of data
- **Example Hooks**
  - Include nearest *OpenWeather* forecast for data record
  - Include USGS Elevation for device location
  - Include the nearest crowd-sourced data point

Insights from *WeatherCitizen* Data

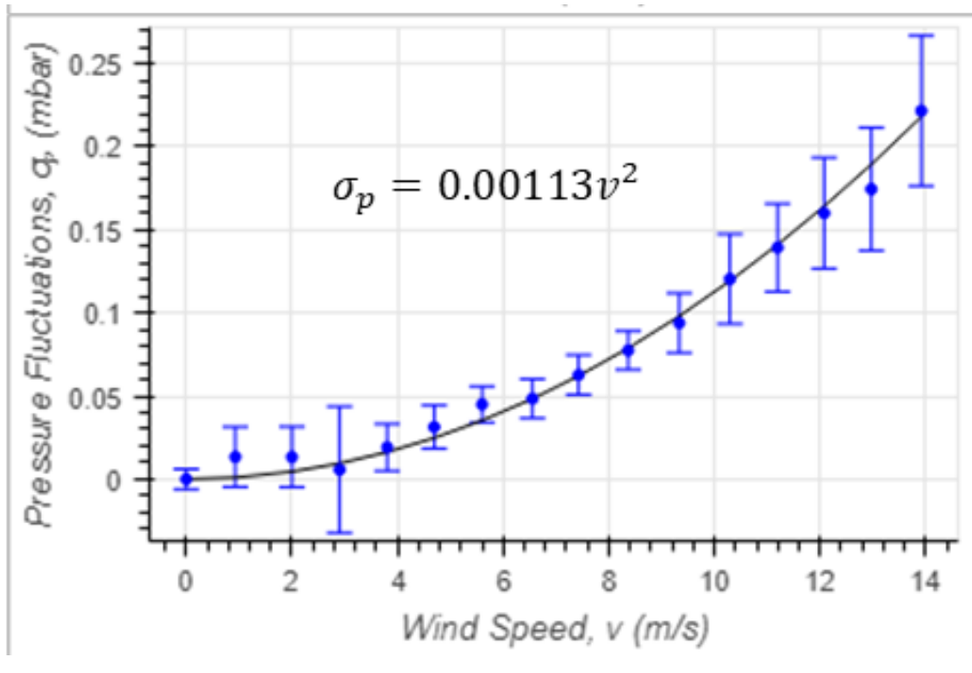








Measure wave period and frequency using orientation sensor



Measure wind speed using only device pressure sensor

Label uploaded images with custom machine learning model

Coming Soon...

- Data assimilation experiment using WeatherCitizen data
- Serverless database architecture to enable self-hosting
- User profiles and granular privacy settings
- Image recognition model for weather-related phenomena
- **Looking for early adopters to provide feedback and drive future use-cases**

Acknowledgements

This work is supported by:

- **NOAA** SBIR Contract No. WC133R17CN0081
- **US Army** SBIR Contract No. W91E518C0001

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Army Corps of Engineers or NOAA.

Ways to Connect

At AMS:

- Poster Session Today - 4:00 PM-6:00 PM in Hall 4, Poster #671
  "*PODPAC: A Python Library for Automatic Geospatial Data Harmonization and Seamless Transition to Cloud-Based Processing*"

Email:

- Marc Shapiro: mls@creare.com
- Request Software: weathercitizen@creare.com

Online:

- Project Website: https://weathercitizen.org
- Creare: http://creare.com