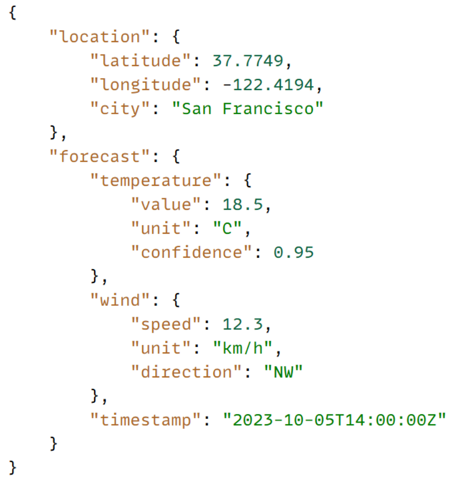
Sample of existing information structure:

Apple’s Weather app features atmospheric variables like temperature, humidity, and wind speed, location data like GPS coordinates and cities names, and time-series forecasts that give hourly or daily estimates – among other things.

We predict the data format of Apple’s current Weather app to look something similar to this:



```
{
    "location": {
        "latitude": 37.7749,
        "longitude": -122.4194,
        "city": "San Francisco"
    },
    "forecast": {
        "temperature": {
            "value": 18.5,
            "unit": "C",
            "confidence": 0.95
        },
        "wind": {
            "speed": 12.3,
            "unit": "km/h",
            "direction": "NW"
        },
        "timestamp": "2023-10-05T14:00:00Z"
    }
}
```

With the exception that instead of direct inputted values like 18.5 celsius, the data is actually accessed from sites like the Weather Channel, likely through Swift or REST APIs (WeatherKit). Which is then communicated into Apple’s Weather app.

Sample of the new structure our team has designed:

We aim to integrate ocean data into Apple’s Weather app that includes variables such as surface water temperature, tides, and so forth.

Similarly, to our prediction of Apple’s data format, our structure would look something like this:

```
{
    "location": {
        "latitude": 34.0522,
        "longitude": -118.2437,
        "is_coastal": true,
        "ocean_zone_id": "US-PAC-001"
    },
    "ocean_forecast": {
        "water_temp": {
            "value": 21.2,
            "unit": "C",
            "source": "NOAA Buoy #46221"
        },
        "wave_height": {
            "value": 2.5,
            "unit": "m",
            "confidence": 0.9
        },
        "tide": {
            "next_high": "2023-10-05T18:24:00Z",
            "level": 1.8
        },
        "timestamp": "2023-10-05T14:00:00Z",
        "data_quality_score": 0.88
    }
}
```

Much like Apple's current setup, we would access the data from sites like NOAA, likely through Swift or REST APIs.

How our structure promotes maintaining and measuring quality easier:

Apple's Weather app with ocean integration features will promote maintaining and measuring quality easier through automatic regression testing and enforcement of data integrity.

Data upload times vary depending on organization and data, for example NOAA's uploads of sea surface temperature maps daily whereas NOAA's water level datasets are updated every 6 minutes. King County's water stations' data can be updated every 15 minutes or every hour. Thus there's differences between the access timing internally in organization and across multiple organizations. We'll need to account for that in our project to maintain data quality. We need to ensure we're referring to the correct dataset for display, and accessing data at the correct times with automatic regression testing. We'll maintain strong data governance practices to maintain and track the datasets as they update, depending on the needs of the datasets (6 minutes vs daily vs every hour and so on).

Aggregating multiple datasets between the county-level and federal-level data will require proper enforcement and monitoring to ensure data sources and the various formats are uploaded into the system's platform and integrated properly to the app.

How our structure enables any information security issues to be identified and managed down the road:

As we will be using publicly accessible government data, there is risk as there would be accessing data on other publicly accessible websites. We will employ a checker in our code to ensure the data's integrity, ensuring intact, accurate, and complete data (Exabeam). Our current information structure provides clear boundaries around locations and forecasts, which will ease our program in checking specific parameters. Implementing checkers in our program for valid values like ranges of water temperature ensures any other value out of those bounds won't go through our programs and will instead notify those managing the program. This also serves as a method to keep our program operating regularly, with minimal disruption from invalid data.

Our ocean data structure will also use access controls. These controls would encrypt locally stored information and use API keys to retrieve data from NOAA's sites. This ensures the data remains confidential, but still available to our program as needed.

Our current information structure also follows the regulations for data, avoiding using PII and providing only the data that is already publicly available like the buoy identification numbers.

References

Exabeam. *The 12 elements of an information security policy.* Accessed May 15, 2025.

"WeatherKit." Apple Developer, developer.apple.com/weatherkit/.