

FEWS based web applications

Shannon Gross

Water Resource Engineer // Environmental
Engineering Applications Developer



Outline

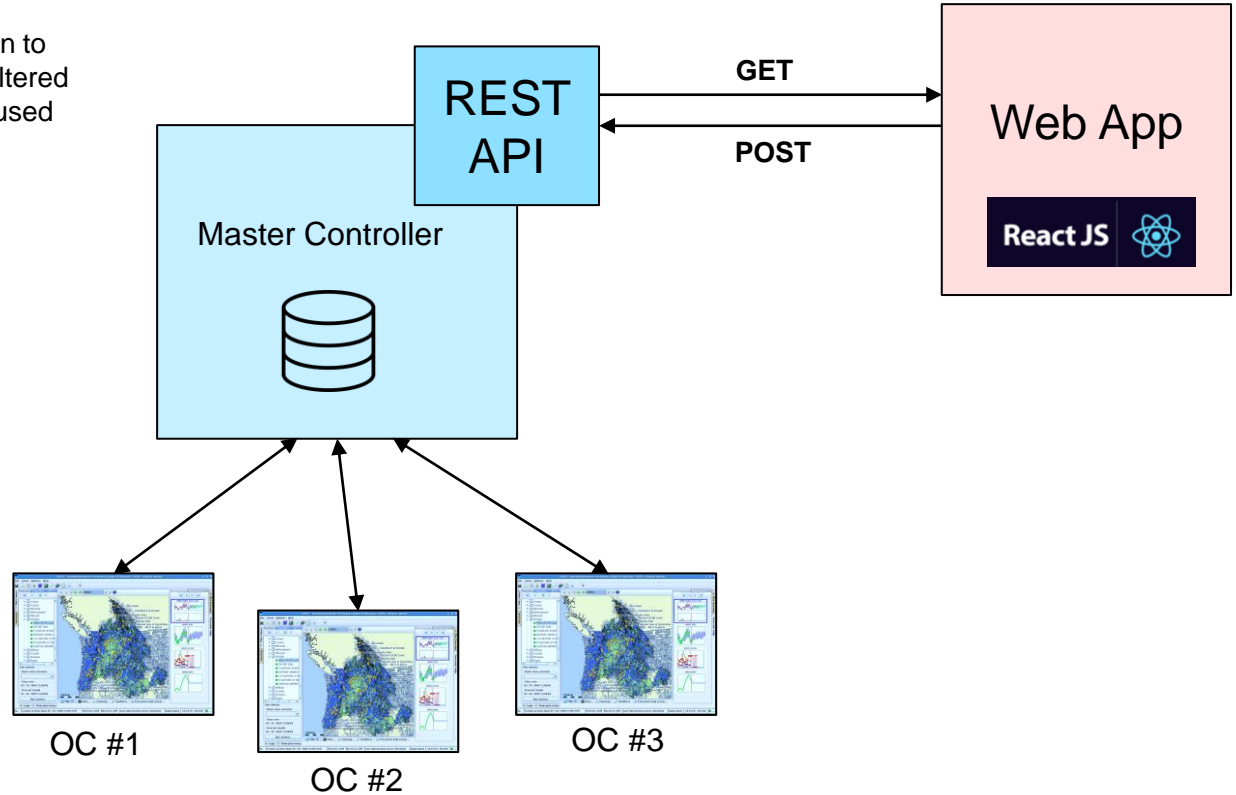
- Concept of building web applications on top of FEWS
- Why would this be desirable (if you already have FEWS)?
- Benefits, challenges
- Two case studies
- Discussion

Concept

- **API** (Application Programming Interface): A set of rules that define how apps can communicate
- The FEWS API allows the external application to request timeseries data from the database, filtered by the query parameters, returned in widely-used JSON format.

Example JSON response:

```
{ "timeSeries": [ {  
  "header": {  
    "moduleInstanceId": "James_ECMWF_Fcast",  
    "locationId": "LakeJames",  
    "parameterId": "Streamflow",  
    "qualifierId": [ "Adj", "ECMWF" ] },  
  "events": [ {  
    { "date": "2023-11-06",  
      "time": "08:00:00",  
      "value": "5.59"  
    }, {  
    { "date": "2023-11-06",  
      "time": "09:00:00",  
      "value": "6.23"  
    }, {  
    { "date": "2023-11-06",  
      "time": "10:00:00",  
      "value": "1.44"  
    }  
  ]  
}
```



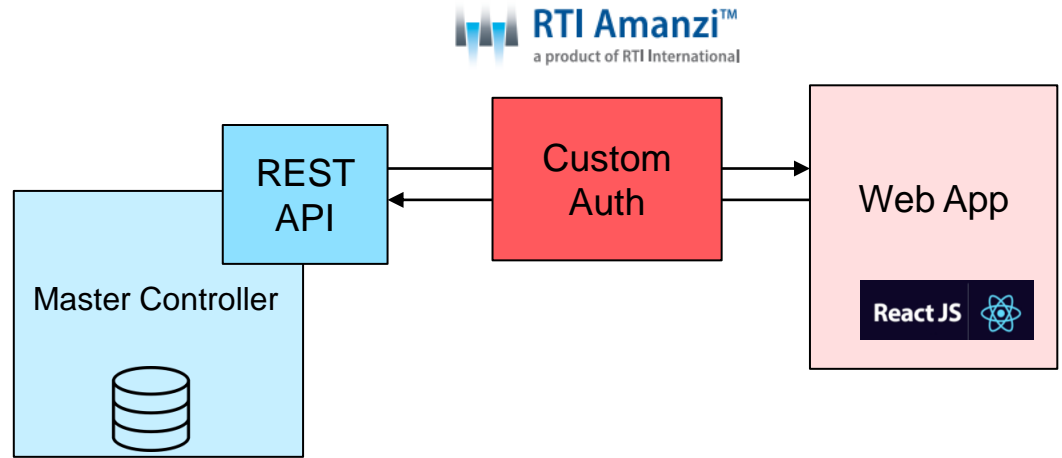
Reasons for web app on top of FEWS

- Web app provides convenient access to forecasts compared to OC
 - Easy to access from web browser, from phone
 - Ability to access or host outside of network

- Endless possibilities for customizing the web app display
 - Simplification of display may be desired (for users that only need a limited set of information)

Challenges faced

- User management
 - Custom authentication and authorization functionality was needed



- Managing user expectations
 - Users may only have a superficial understanding of how their forecast system works if their only interaction is the web app

Demo

<https://wfp-custom-dev.westus2.cloudapp.azure.com/>

Kings Data Entry

MM/DD/YYYY hh:mm aa



Head

Turbine Flow

Gate 1

Gate 2

Gate 3

Total Flow

