CYBERINFRASTRUCTURE INTEGRATION RESEARCH CENTER
PERVASIVE TECHNOLOGY INSTITUTE

Spring 2022 Project 1

January 18th 2022

Suresh Marru

Enrollment is final, lets get started

Project Mechanics

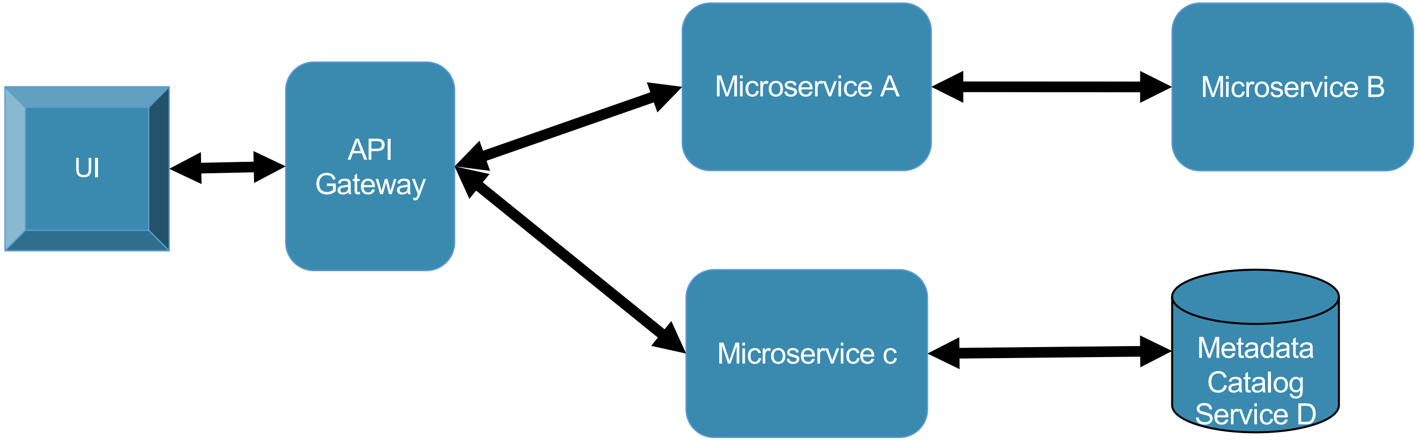
- Project teams
  - Still missing 4 submissions
- Use all GitHub software engineering tools to start working on your project.
- Plan on making your repos and wiki's ready for peer-review.
- Peer-reviews will be your open source user community, your project team is the PMC - https://www.apache.org/foundation/governance/pmcs.
- You submit the project for grading.
- TA's will grade the work of the team and peer reviewers and the team's response to peer reviews.

Project 1 Deadline

- Project 1 due February 4^{th}
  - Bonus point peer reviews until February 6^{th}

Implement a small full stack distributed system

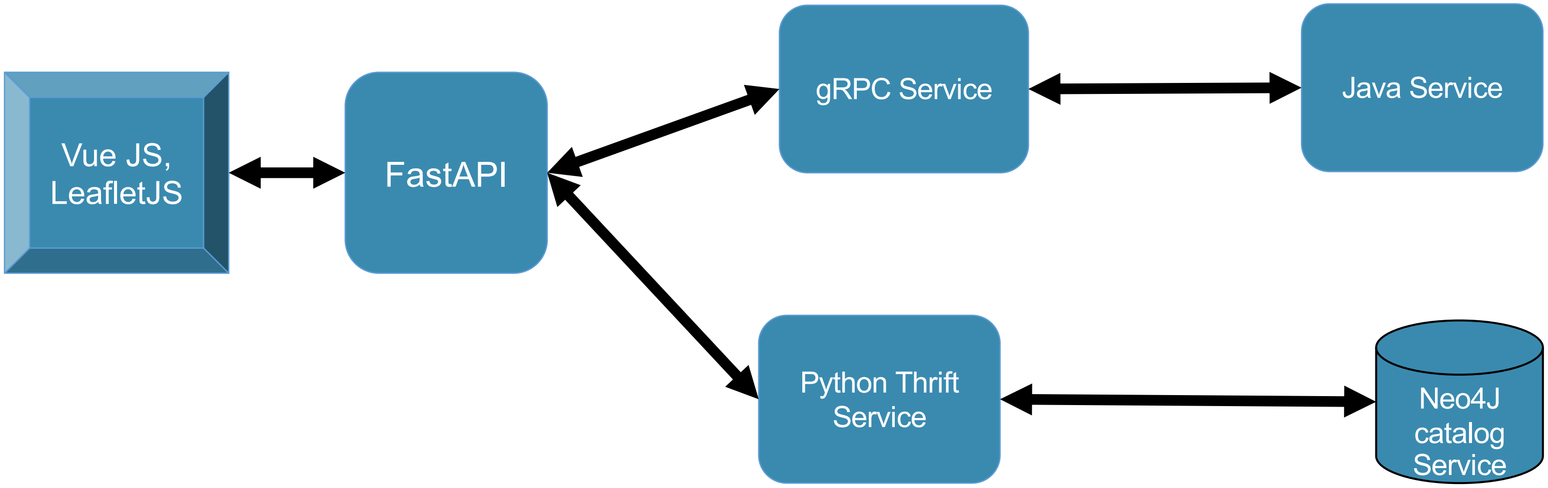
Sample Architecture



Technology Choices

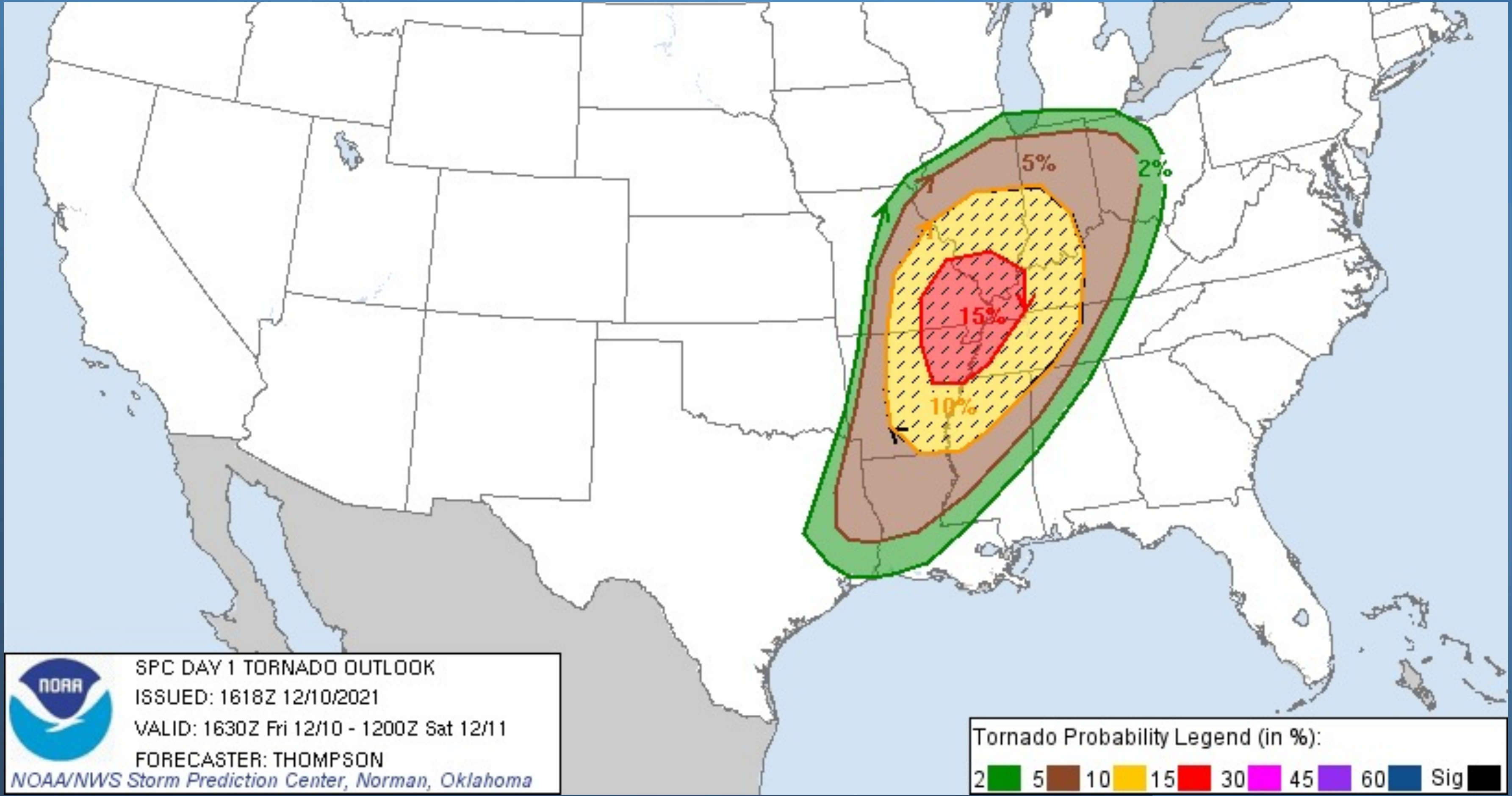
- We will not be prescriptive but can make suggestions.
- Need to choose at least 3 programming languages.
- All components (including UI) need to use a build framework.
    - Make, Maven, Bower……
- Required to have a README instructing how to checkout, build, run, verify.

Example Choices



Assignment 1 Preparation

- Learn how to write API's in REST or Apache Thrift or gRPC/ProtoBuff
- Decide on your Programming Languages.
- Decide on your Web Framework.
- Learn how to use build systems like Apache Maven.
- Test-Driven Development



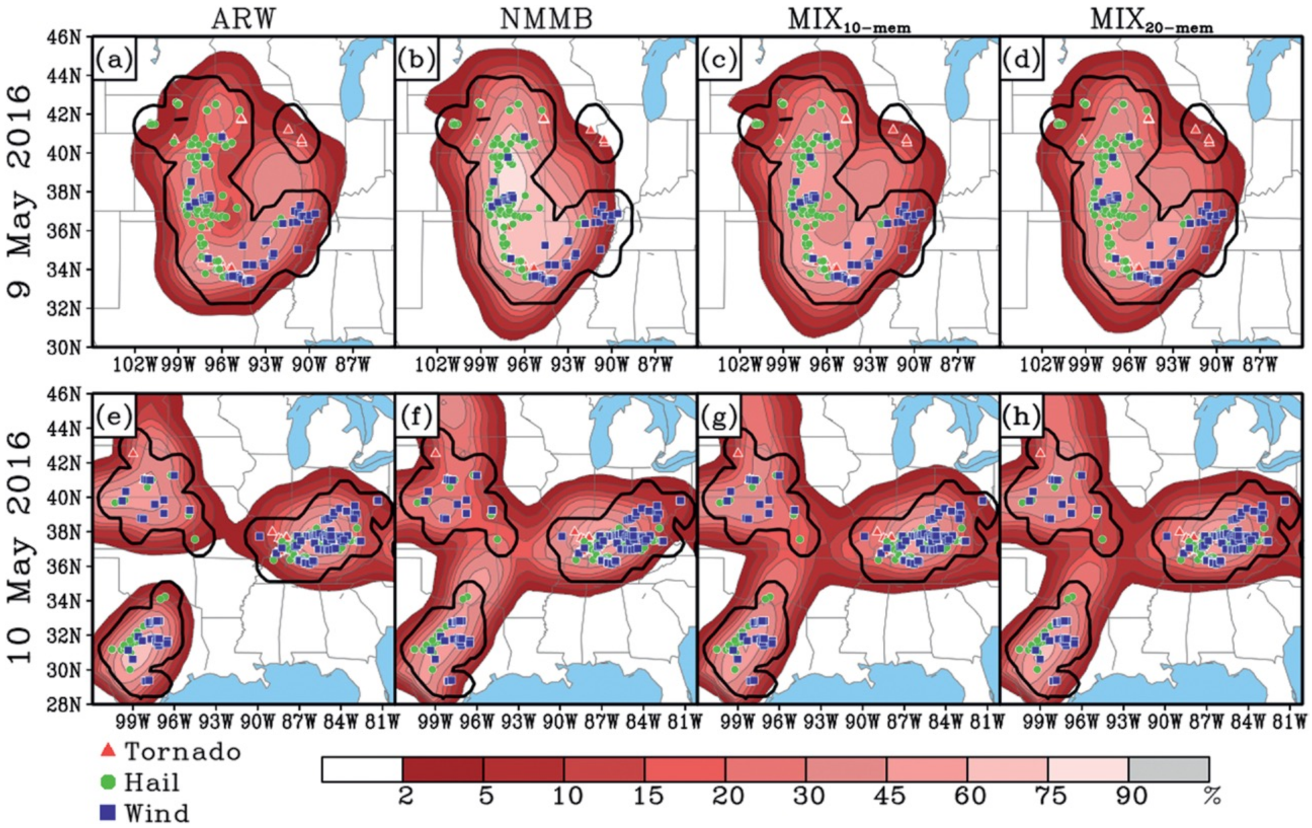
5%
2%
15%
10%
SPC DAY 1 TORNADO OUTLOOK
ISSUED: 1618Z 12/10/2021
VALID: 1630Z Fri 12/10 - 1200Z Sat 12/11
FORECASTER: THOMPSON
NOAA/NWS Storm Prediction Center, Norman, Oklahoma
Tornado Probability Legend (in %):
2 5 10 15 30 45 60 Sig

Weather Forecasting

- Current weather determined by observations is the initial state.
- The atmosphere is a physical system governed by the laws of physics
  - these laws are expressed as mathematical equations.
  - models start from initial state (observations) and calculate state changes over time.
  - Models are very complicated (non-linear) and require supercomputers to do the calculations.
- Forecast duration defines temporal boundary conditions
  - the accuracy decreases as the range increases; there is an inherent limit of predictability.

Tornado Prediction Data Sizes

- Center for Analysis and Prediction of Storms (CAPS) at the University of Oklahoma runs high resolution simulations as part of HWT - https://hwt.nssl.noaa.gov/
  - Domain: Continental United States (CONUS): 1683x1155x53 with 3 km grid spacing (starting from center with 3km space, move 1683 in x direction, 1153 in y direction and 53 steps in z direction – vertical, above the earth).
  - Time steps: 6 minutes – every 6 minutes from forecast initialization do an I/O of model outputs.
  - A single forecast writes 1639 GB of data.
  - 20 ensembles of simulations are run with varying emphasis on initial conditions, changes to physics.
  - Collectively a single day forecast produces 132 TB.


ARW
NMMB
MIX_{10-mem}
MIX_{20-mem}
9 May 2016
10 May 2016
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
46N
44N
42N
40N
38N
36N
34N
32N
30N
28N
102W 99W 96W 93W 90W 87W
99W 96W 93W 90W 87W 84W 81W
Tornado
Hail
Wind
2
5
10
15
20
30
45
60
75
90
%

AWS NEXRAD Data

- https://registry.opendata.aws/noaa-nexrad/
- https://docs.opendata.aws/noaa-nexrad/readme.html
- https://s3.amazonaws.com/noaa-nexrad-level2/index.html

Plot NEXRAD Data

- Example Library: Py-ART - http://arm-doe.github.io/pyart/
- A concrete example - https://arm-doe.github.io/pyart/dev/auto_examples/plotting/plot_nexrad_reflectivity.html

Example

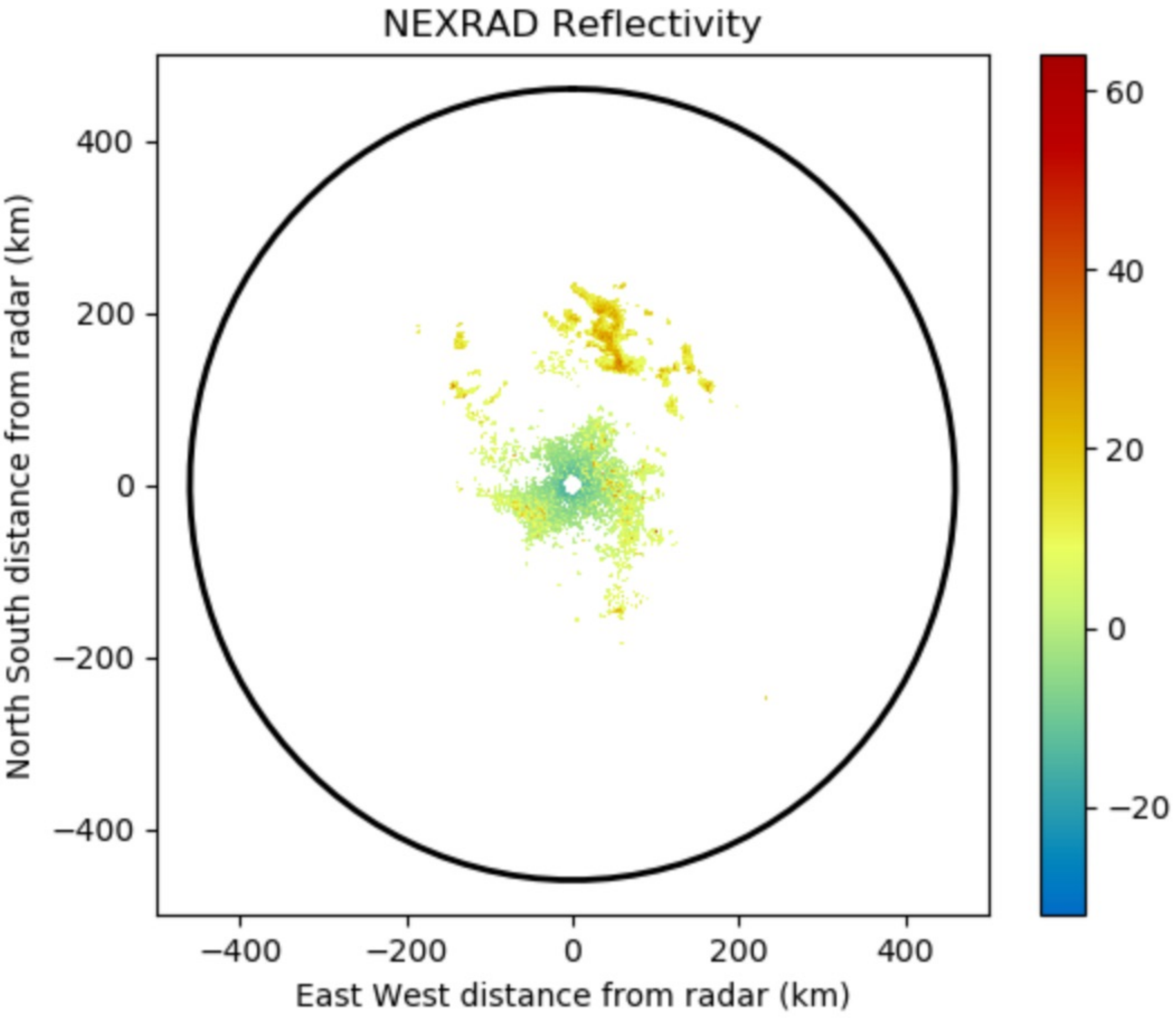
```
print(__doc__)

# Author: Jonathan J. Helmus (jhelmus
# License: BSD 3 clause

import matplotlib.pyplot as plt
import pyart

# open the file, create the displays
filename = 'Level2_KATX_20130717_1950
radar = pyart.io.read_nexrad_archive(
display = pyart.graph.RadarDisplay(ra
fig = plt.figure(figsize=(6, 5))

# plot super resolution reflectivity
ax = fig.add_subplot(111)
display.plot('reflectivity', 0, title
             vmin=-32, vmax=64, color
display.plot_range_ring(radar.range['
display.set_limits(xlim=(-500, 500),
plt.show()
```



User Interface

- Pick your Favorite web framework/language
- Have a user management, ok to use cloud services, but preferably open source software.
- Milestone 1: User triggers "diagnose current atmospheric conditions"
  - Provide input of Date, Time and NEXRAD station name (http://www.nws.noaa.gov/tg/pdf/wsr88d-radar-list.pdf)
  - List all interactions queried from a database.

Microservice A – Registry

- Persist all actions of the science gateway and show a quarriable audit trails.
- Log all requests, responses and times and display them through API.

Microservice B - Data Ingestor

1. Accept users input and return an acknowledgement.
2. Outputs a Data file URL
   - Refer to https://aws.amazon.com/noaa-big-data/nexrad/
     - /<Year>/<Month>/<Day>/<NEXRAD Station>/<filename>
     - <filename> is the name of the file containing the data (compressed with gzip). The file name has more precise timestamp information.
3. Advanced Track
   - Real Time triggers using Amazon Simple Queue Service or Amazon Lambda NoOps.

Microservice C – Storm Detection

- Detect 3D storm characterized by the reflectivity over a given threshold.
- Basic Track will mock it up and output dummy kml.
- Advanced Track will port an existing C++ library to “Big Data” compatible techniques.
- Advanced++ Track will compare and contrast with other approaches like “Connected Component Analysis”.

Microservice D – Storm Clustering

- Group the storm events detected into spatial clusters using Density based clustering algorithm.
- Basic Track will mock the application and return dummy clusters.
- Advanced Track will port the existing C++ library.
- Advance Track will use EC2 "Big Data" pipelines and services like Kinesis.

Microservice E – Forecast Trigger

- Make Decision on to run forecasts or not.
- Basic Track can mock the decisions but show both stopping and moving foreword of control.
- Advance Track will use real decisions.

Microservice F – Run Weather Forecast

- Basic Track will mock it up and return dummy forecast outputs.
- Advanced Track will invoke Apache Airavata API to launch a WRF application and track progress.

Implement services