Project Milestone 1: Description of “microservices”

Marlon Pierce, Suresh Marru

CSCI-B 649 Science Gateway Architectures

Logistics

- Project Team Questions?
- Associate Instructors
  - Anuj Bhandar (anujbhan@umail.iu.edu)
  - Ajinkya Dhamnaskar (adhamnas@umail.iu.edu)
  - Abhijit Karanjkar (aykaranj@umail.iu.edu)
  - Mangirish Wagle (mawagle@umail.iu.edu)

Project Use Case

- Develop a simplified personal weather predictor
  - Mock the implementations to avoid getting distracted.
  - Focus on the Science Gateway Architecture, a special case of "Distributed System".

Weather Forecasting Summary

- 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.

Weather Forecasting Future Directions

- Atmospheric community is progressing towards:
  - – detailed short-range forecasts, using storm-scale models able to provide skillful predictions of severe weather.
  - – applying progress in data science to data assimilation and extracting the maximum possible information from observing systems, especially remote sensors such as satellites and radars.
  - – More CASA like adaptive observing systems, in which additional observations are placed where ensembles indicate that there is rapid error growth (low predictability).
  - – increase use of ensemble forecasting (particularly for medium range).
  - – Coupled atmospheric–hydrological systems, where the atmospheric model precipitation is down-scaled and used to extend the length of river flow prediction.

Course Projects: Weather Predictor

- Two distinct and dependent aspects:
  - Diagnostic
    - Assimilate initiate state of the atmosphere.
    - Focus on Milestone 1
  - Prognostic
    - Knowledge of the physical laws which determine the evolution of the atmosphere.
    - Requires running weather simulation models on supercomputers.
    - Focus of project Milestone 4

Project Milestone 1 Preparation

- Learn how to write API’s in REST and Apache Thrift
- Decide on your Programming Languages.
- Decide on your Web Framework.
- Learn how to use build systems like Apache Maven.
- Test-Driven Development

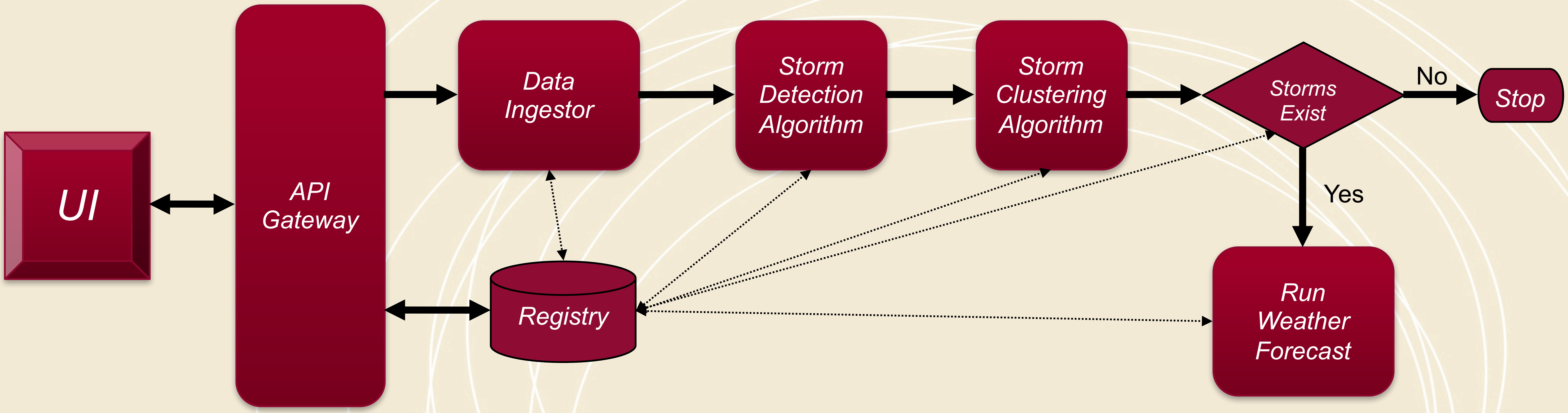
Three levels of open source licenses

- Three levels of open source licenses
  - **Level 1**: Gimme credit (APL, BSD, MIT)
    - You can use, modify and redistribute my code in your product but give me credit (actually, modern BSD doesn't even require credit).
  - **Level 2**: Gimme fixes (MPL, CDDL, LGPL)
    - You can use, modify and redistribute my code in your product but give me the source for any fixes you make to it.
  - **Level 3**: Gimme it ALL! (GPL)
    - You can use, modify and redistribute my code in your product but give me your entire product's source code.

Source: http://rollerweblogger.org/roller/entry/gimme_credit_gimme_fixes_gimme

We recommend to use Level 1 licensed software

Implement “mock” services



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 queriable 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.

Questions

Marlon Pierce, Suresh Marru

{marpierc, smarru}@iu.edu