CYBERINFRASTRUCTURE INTEGRATION RESEARCH CENTER

PERVASIVE TECHNOLOGY INSTITUTE

Spring 2020 Project Overview

January 16th 2020

Suresh Marru, Marlon Pierce

Applied Distributed Systems

- We will build user-centric distributed systems that support scientific research.
  - Science gateways
  - Cyberinfrastructure
- This course will be project-based.
- You will build distributed systems.

Americans Rank A Google Internship Over A Harvard Degree

Brandon Busteed Contributor ⓘ
Education

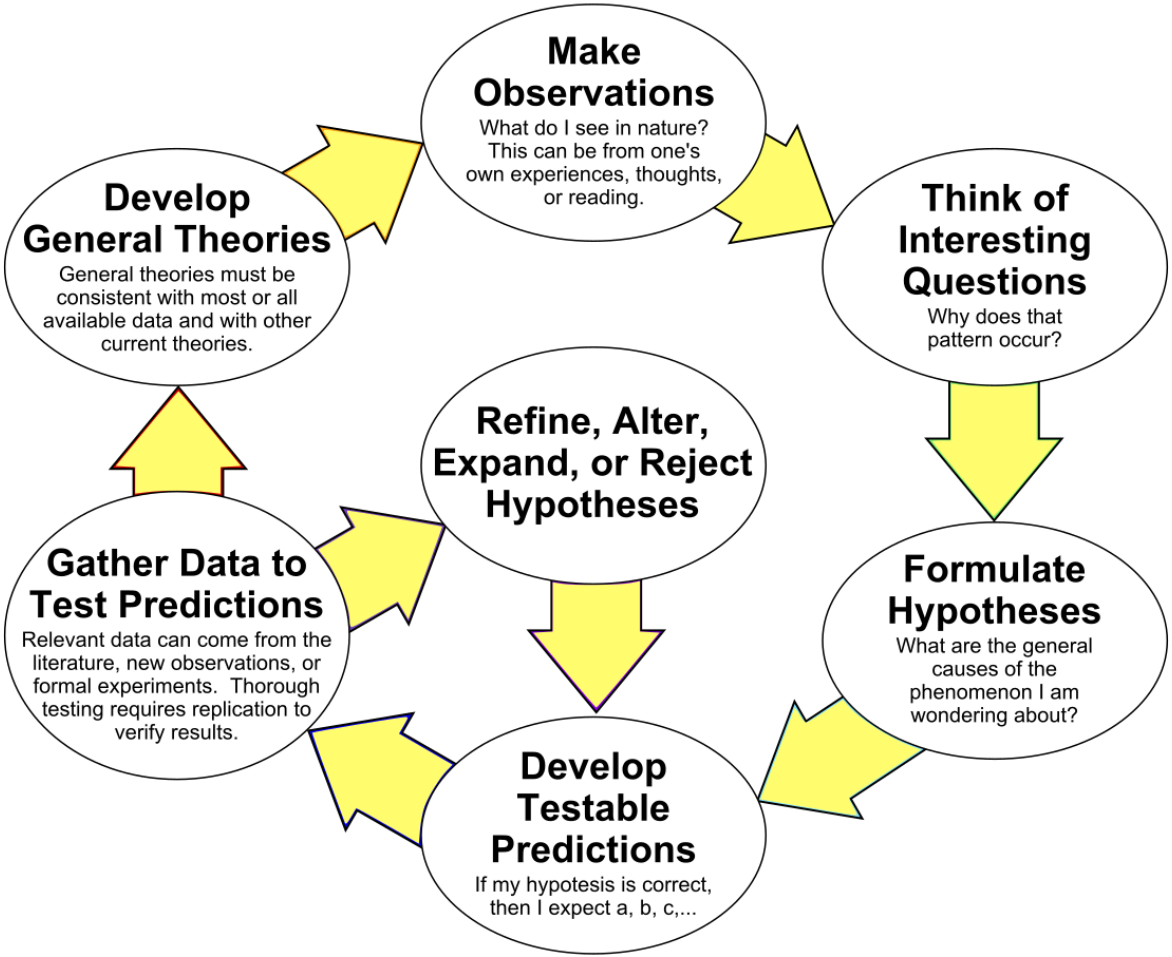
The idea of an internship at Google resonates with Americans in ways far beyond the hit comedic ... [+] GETTY

When asked what they believe would be most helpful for a high school graduate to launch a career, Americans overwhelmingly recommend an internship at Google (60%) over a degree from Harvard (40%). This latest finding from research I led at Kaplan (conducted by QuestResearch Group) is based on a survey of 2,000 U.S.

What we are expecting you to get out of this class?

- A fusion of conceptual skills and “scientific way” of making choices.
- The course is tailored to use tools and technologies relevant in 2020 but our expectation is you will learn how to make choices not necessarily be a tutorial on a buzzy technology.
  - Our definition of a good student is someone who understand the difference between the two.

The Scientific Method as an Ongoing Process



Structure of the Class

- We will have 3 project-based assignments
  - 90% of your grade
  - 25 points/project as a team of 3-4
  - 5 points/project for peer review (individual)
- The first two assignments will be due before semester break.
  - Each team will get the same assignment to build a science gateway using distributed systems concepts
- The third assignment will be for each team to apply your understanding to open problems in Apache Airavata.
- 10% of your grade will be attendance and classroom interactions.

Characteristics of a Good Technology Base

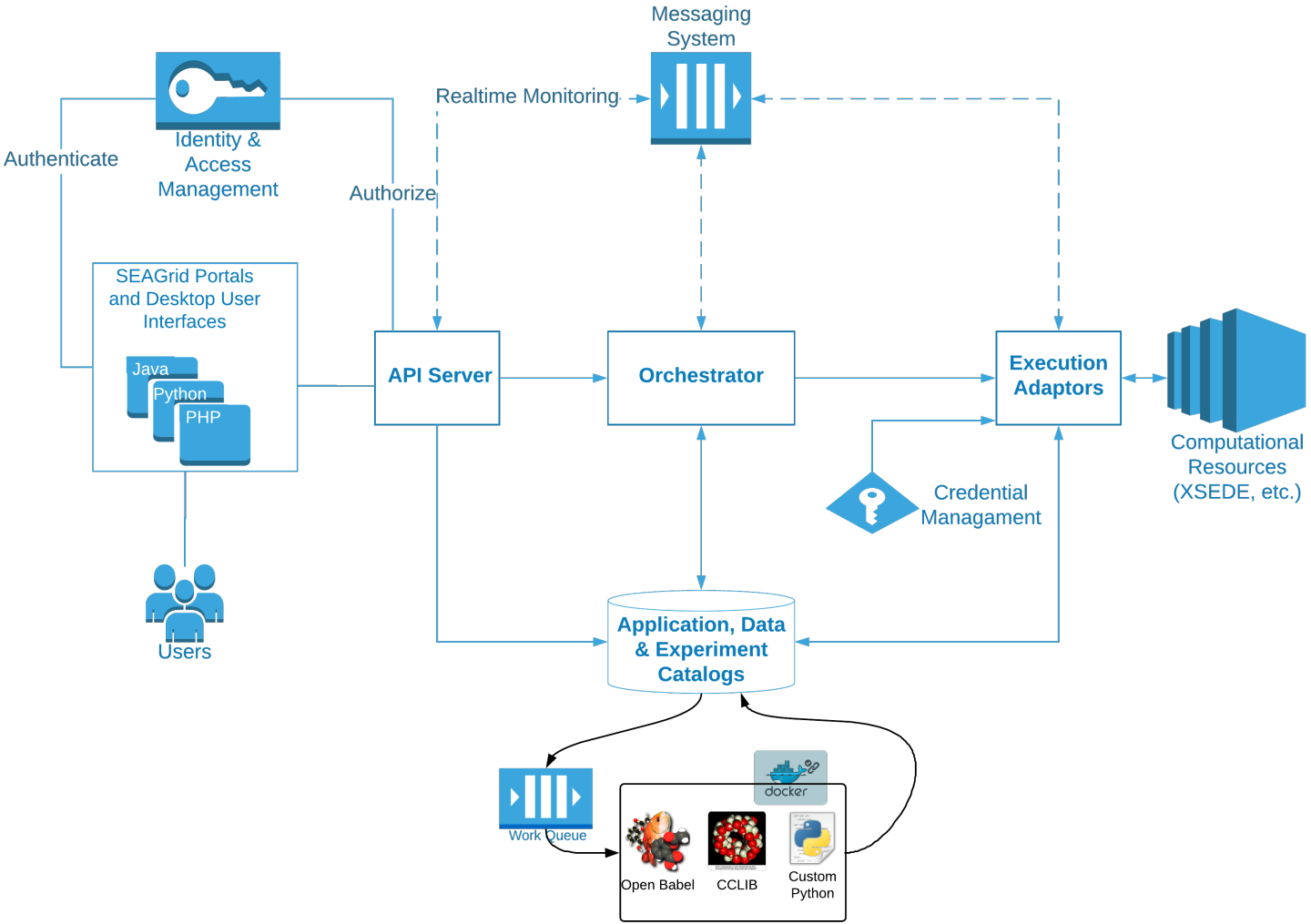
- ✓You are continually improving your code base
- ✓You are strategically adding major new capabilities
- ✓You get improvements expeditiously into production
- ✓You can replace key personnel
- ✓You get meaningful contributions
- ✓You have boring operations: the system as a whole doesn't break, security upgrades aren't a major hassle, etc.
- ✓Parts of your base get reused in other projects.

Project Mechanics

- Create your project team.
- We will populate your team repo
- Use all GitHub software engineering tools to start working on your project.
- Make 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 Deadlines

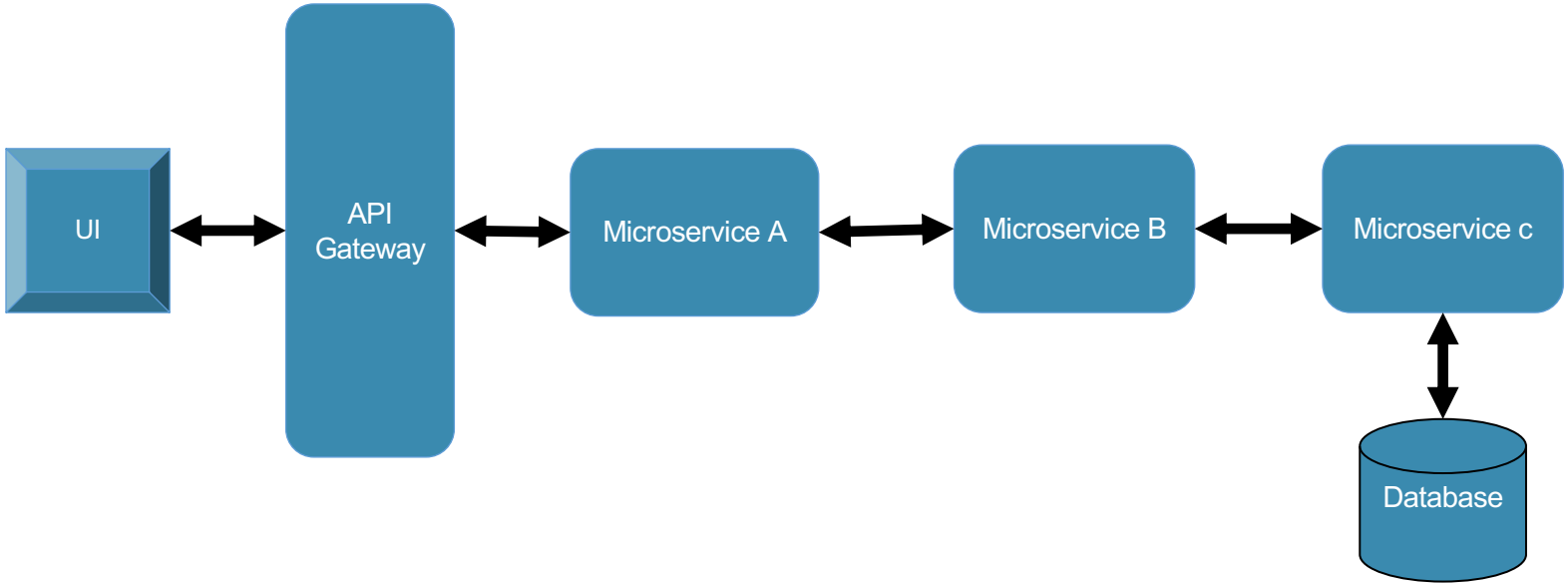
- Project 1 due February 13th
  - Peer reviewers will be assigned and reviews start on February 4th
- Project 2 due March 10th
  - Peer reviews start March 3rd
- Project 3.1 due April 2nd
- Project 3.2 due April 23rd
  - Peer reviews start April 16th


Messaging System
Realtime Monitoring
Identity & Access Management
Authenticate
Authorize
SEAGrid Portals and Desktop User Interfaces
Java
Python
PHP
API Server
Orchestrator
Execution Adaptors
Computational Resources (XSEDE, etc.)
Credential Managament
Users
Application, Data & Experiment Catalogs
Work Queue
docker
Open Babel
CCLIB
Custom Python

Rethink if this course is right for you

Implement a small full stack “micro service” architecture

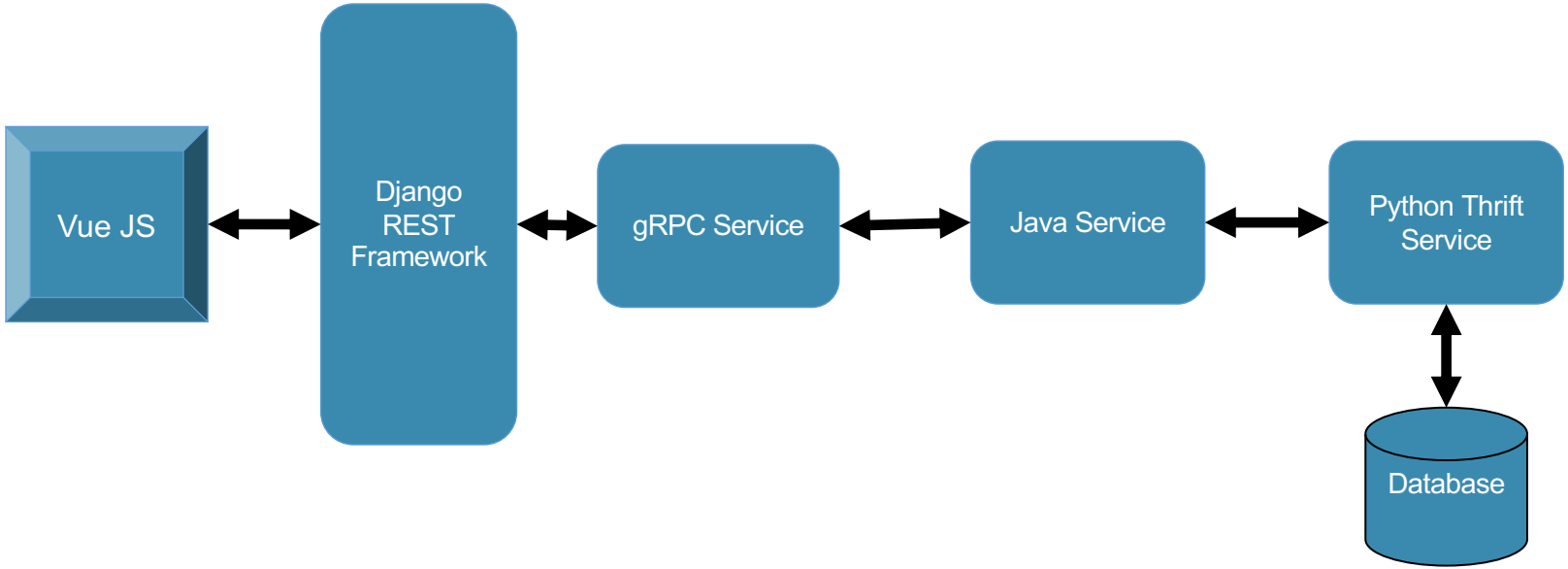
Sample Architecture



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



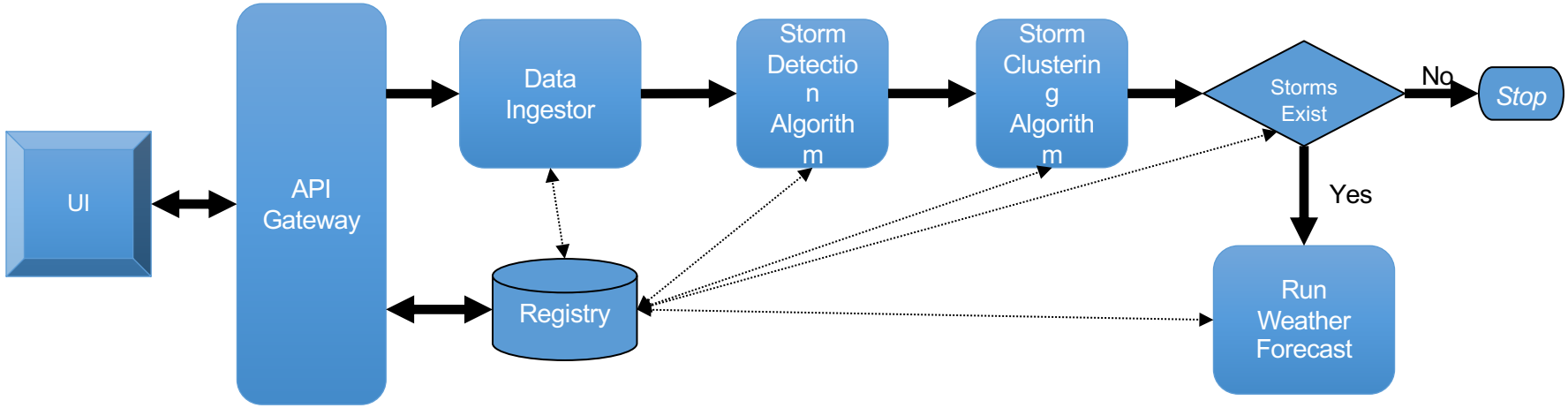
Weather Forecasting Summary

- 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.

Assignment 1 Preparation

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

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.

Implement “mock” services