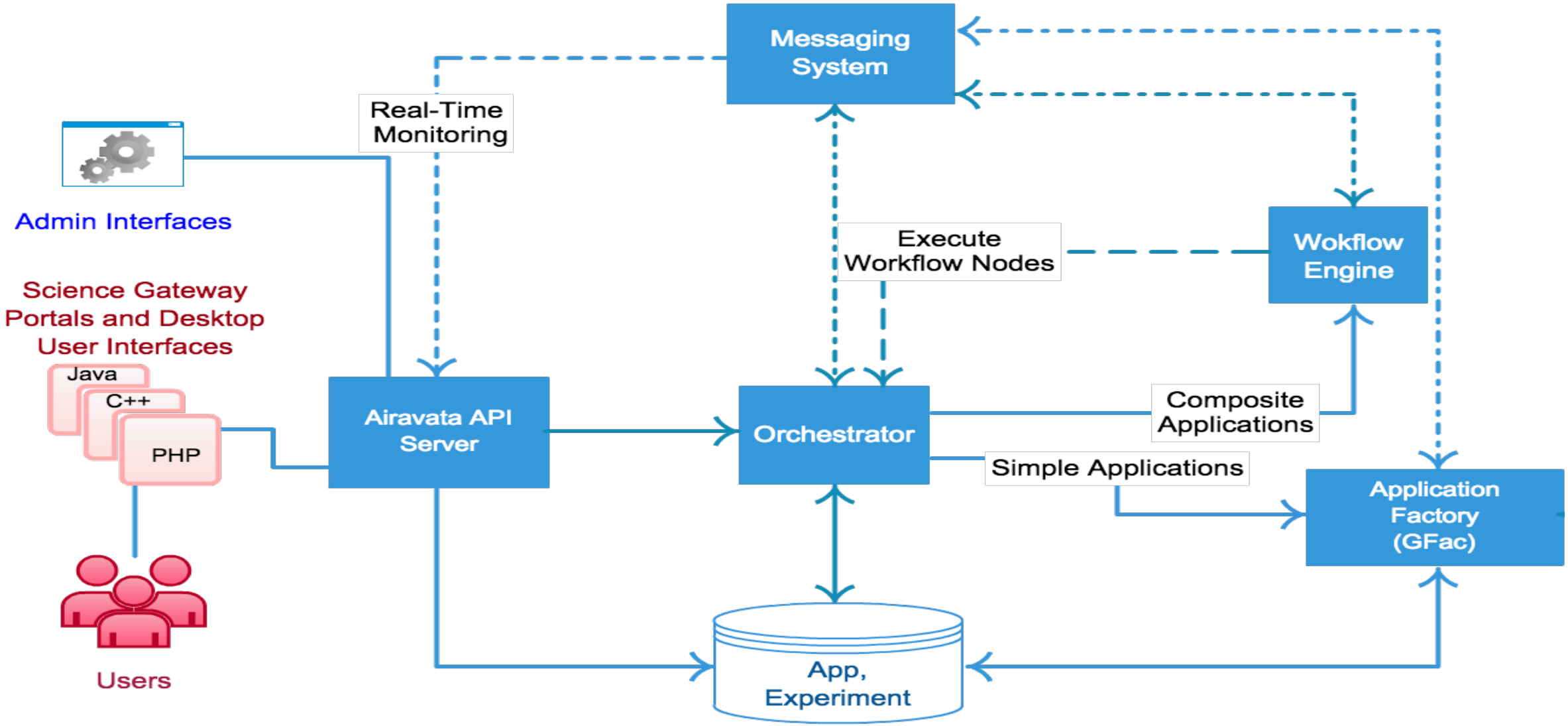
Fall 2017 Recap

Initial thoughts on topics for Spring 2018

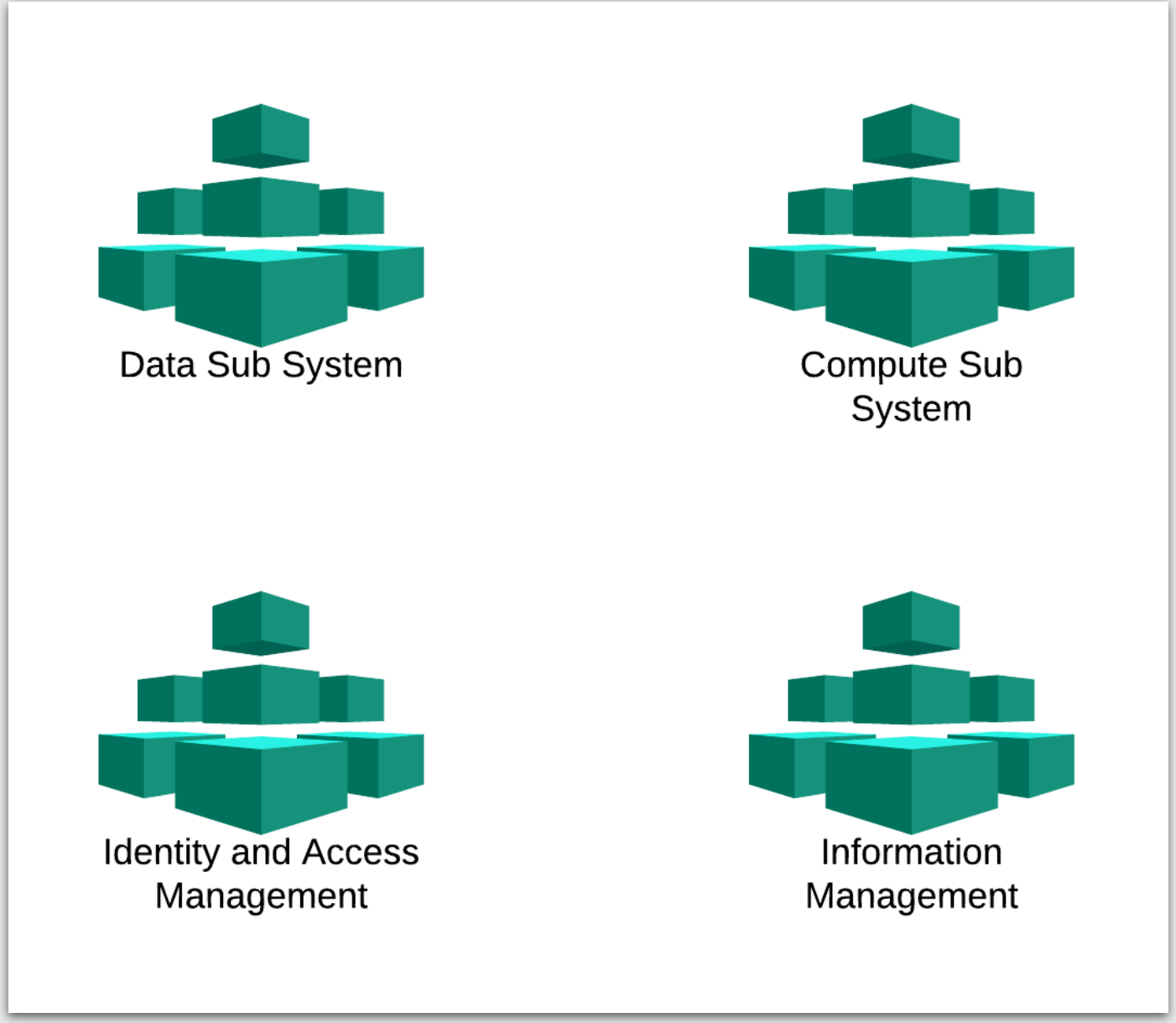
Course Recap

- Micro services
  - Programing language agnostic.
- Containers, CI/CD – DevOps in general.
- Writing software in the Open.
- Architectural choices.
- Distributed System Concepts
  - Raft, Distributed Logging, Messaging, Distributed Co-ordination
- Real-world application – Contributions to a production “Platform as a Service” and “Software as a Service” Systems.


Messaging System
Real-Time Monitoring
Admin Interfaces
Science Gateway Portals and Desktop User Interfaces
Java
C++
PHP
Users
Airavata API Server
Execute Workflow Nodes
Wokflow Engine
Orchestrator
Composite Applications
Simple Applications
Application Factory (GFac)
App, Experiment

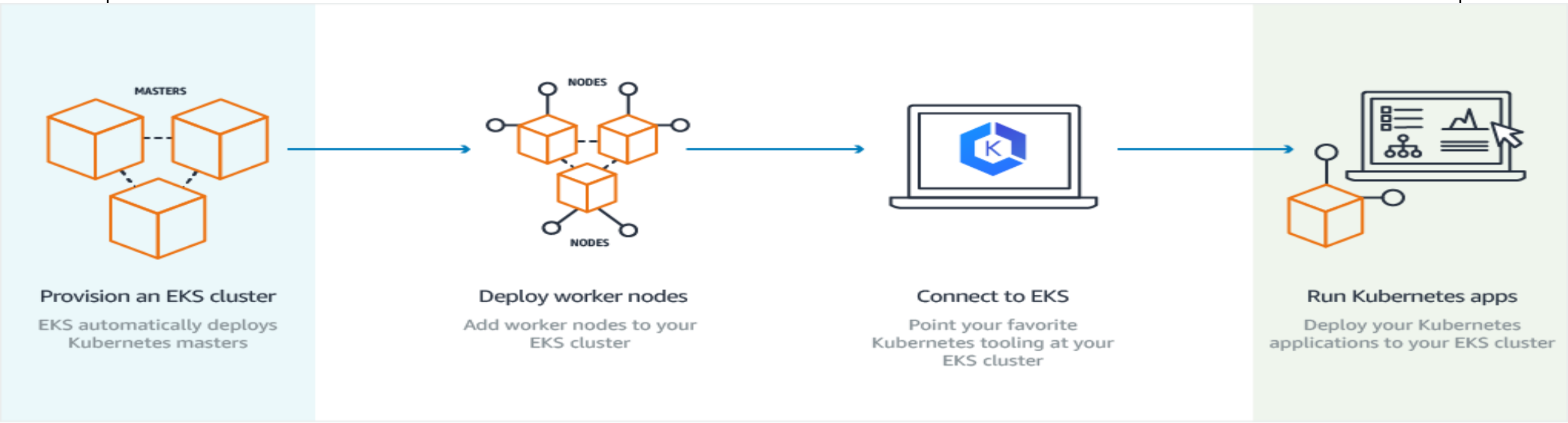
Airavata Development in Spring 2018

- Move from current monolithic API driven approach to more loosely coupled “sub systems”.
- Each sub system constitutes of many micro-services.
- Based on the context, these sub systems will be coupled.
- Moving from a centralized API to context driven SDK’s.

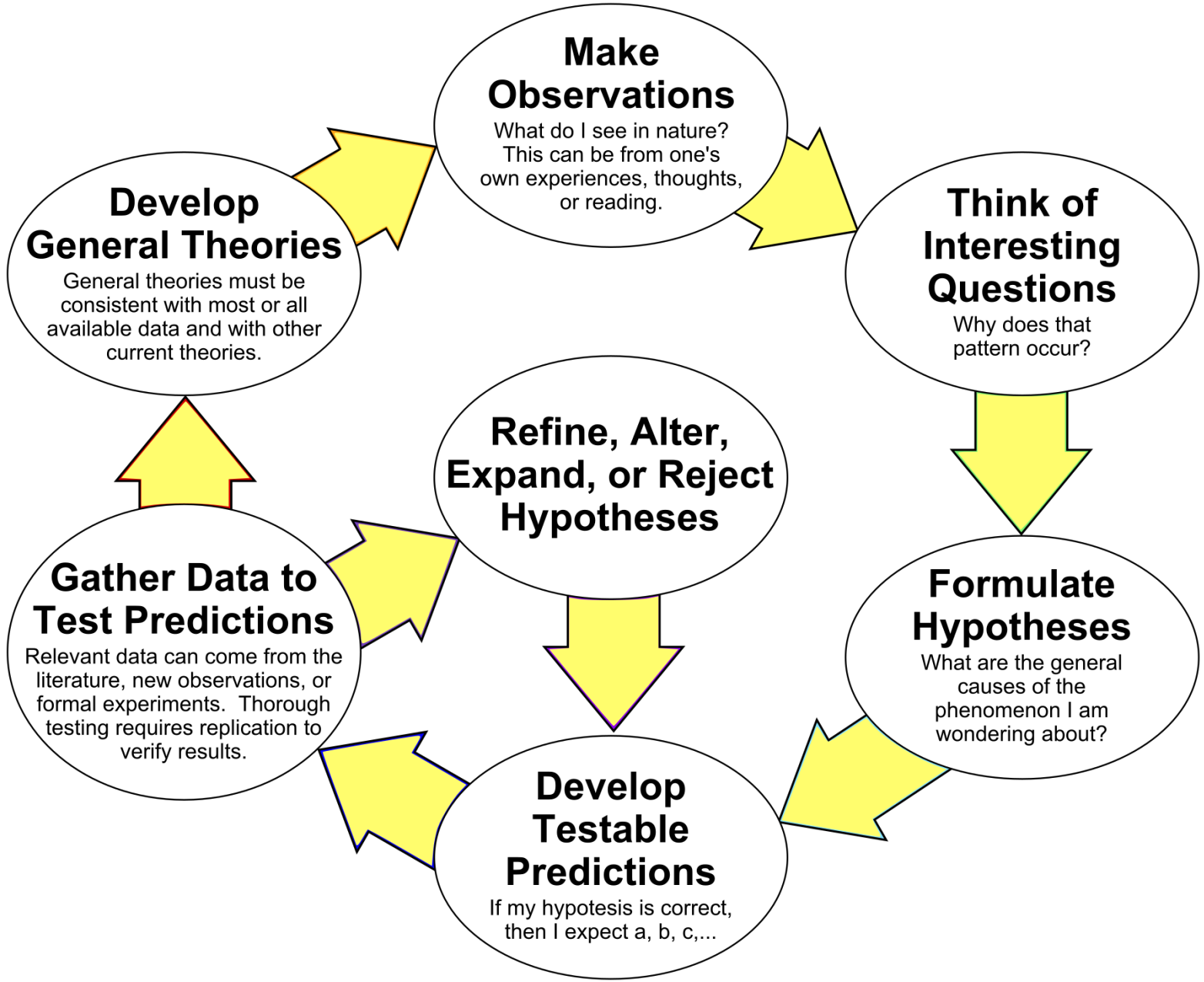


DevOps

- Everything as a Container
- Kubernetes based
- Will closely follow the new Amazon EKS Service



The Scientific Method as an Ongoing Process



Task Execution Framework

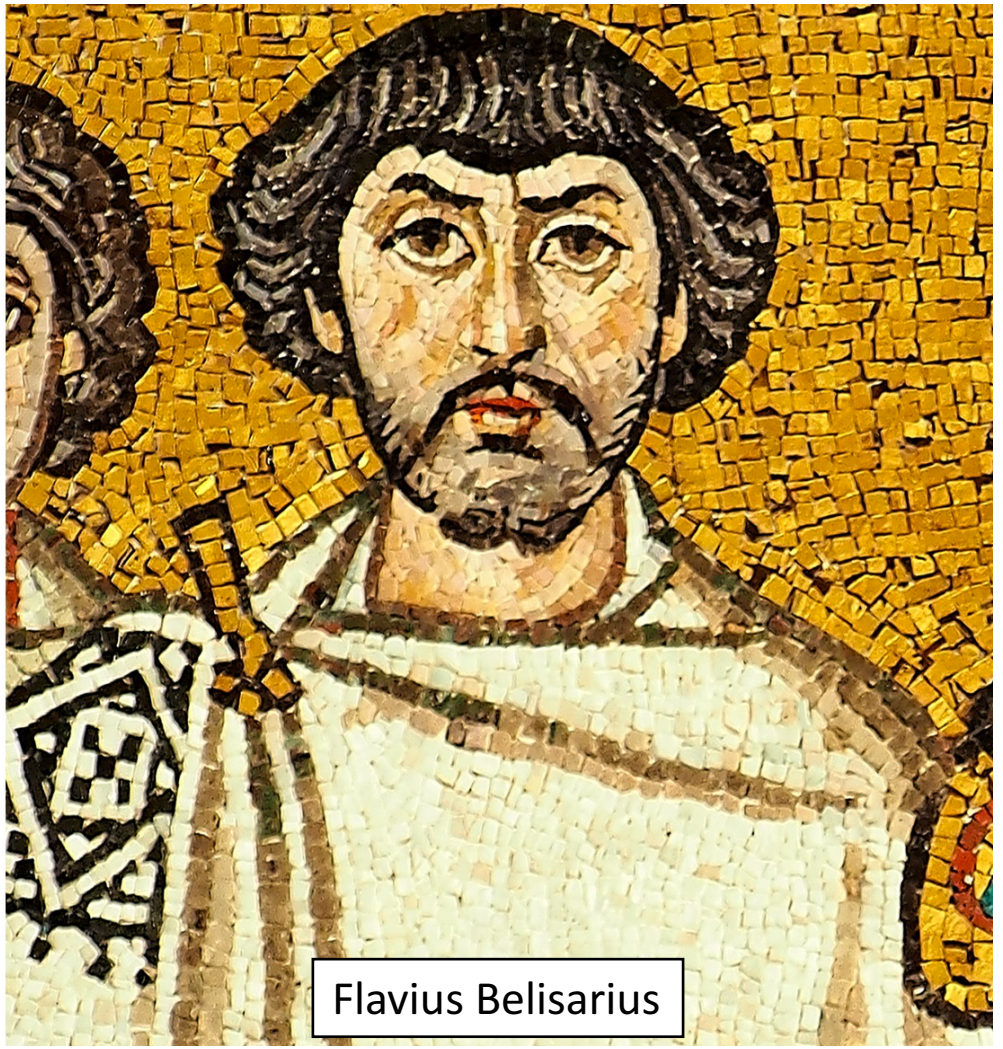
- Gourav Shenoy and Team
  - https://figshare.com/articles/Evaluating_Big_Data_Frameworks_To_Simplify_Distributed_Task_Execution_In_Apache_Airavata/5483743
- Dimuthu's Proposal – Kafka based system
- Final conclusion – Use Apache Helix
  - Browse through Airavata Dev List discussions for the rational and also the provenance on how such decisions are made.
- In the Spring 2018 course you can make similar recommendations and follow through.

Data-Centric Computing

- Wrap up Registry Refactoring
- Craft a first class Data Service Provider API (internally uses current components).
- Integration with Globus, DropBox like tools for seamless data movement.
- Integrating with large scale observational and experimental data catalogs.
- Build upon simulation output parsing system and search capabilities
  - https://scnakandala.github.io/assets/papers/2016/gce-2016.pdf

Raft - Byzantine Failures

- What if a Raft cluster member doesn't behave like it is supposed to?
- Byzantine failure sources
  - Bugs in software
  - Hardware and networking issues
  - Malicious cluster members
- These types of problems are known academically as "the Byzantine Generals Problem"



Log-Centric Airavata

- Help with prototyping log-centric Airavata
  - See Dimuthu's postings
- Develop and evaluate prototypes that use Raft-like strategies for updating stateful services like the Airavata Registry
- Develop and evaluate prototype approaches like Practical Byzantine Fault Tolerance and Blockchain that can be used in log-centric Airavata
  - Decide if this is worth the effort

Security Topics

- What is the best way to distribute OAuth2 public clients that will directly access the API
  - Desktop applications
  - Jupyter
  - Other scripts, developer keys, etc
- Issues with public clients
  - The clients themselves need client IDs and client secrets
  - Users of the clients must further authenticate themselves
  - This should work with non-browser clients
- What are the best approaches for internal (microservice-to-microserivce) security in Airavata?
  - Byzantine Fault Tolerance
  - DevSecOps
  - Black hat hacking