Introduction & Projects Overview

Marlon Pierce, Suresh Marru

CSCI-B 649 Science Gateway Architectures

Spring 2017 – Week 1 - January 13th 2017

INDIANA UNIVERSITY BLOOMINGTON
SCHOOL OF INFORMATICS AND COMPUTING

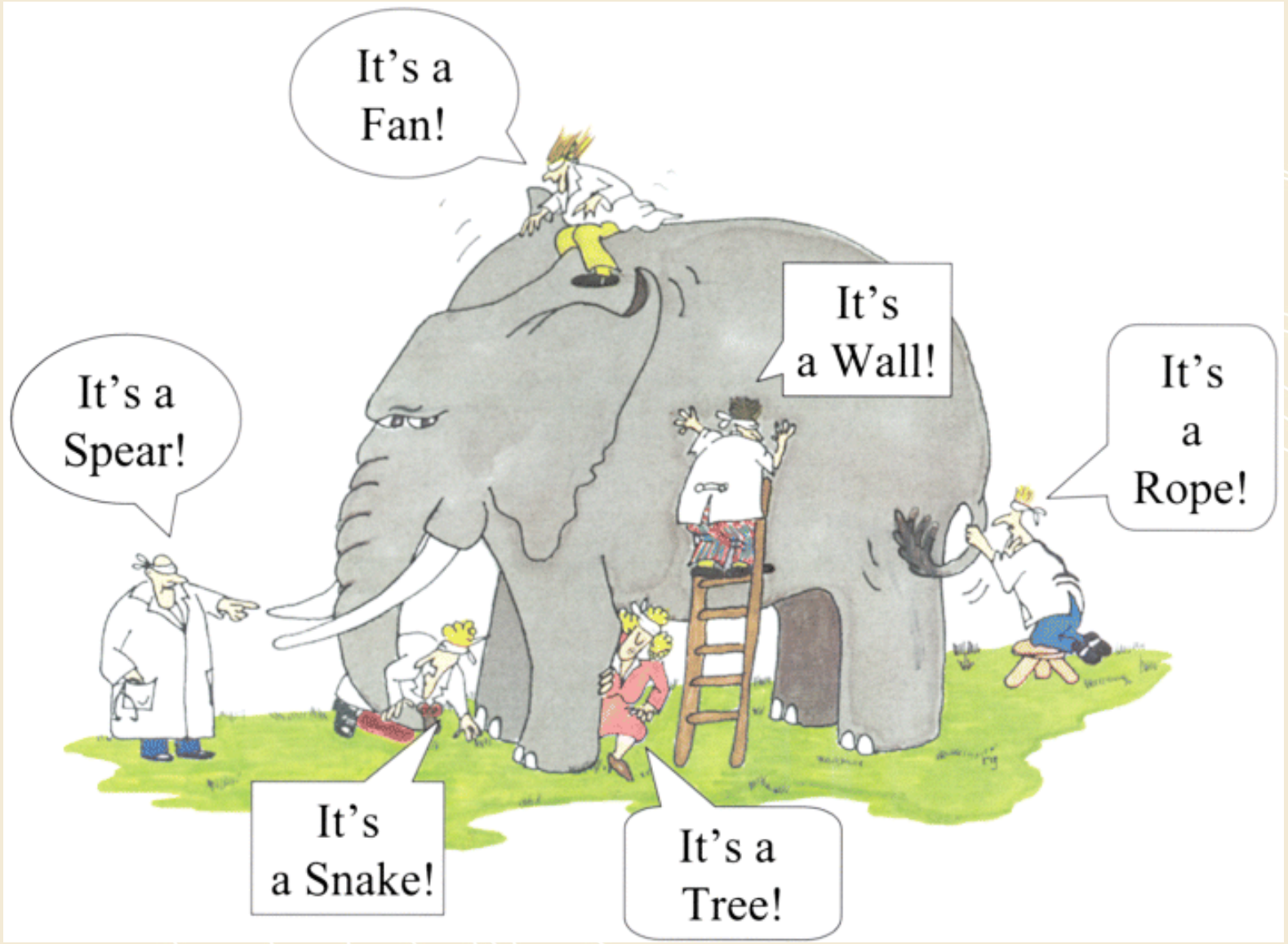
Todays Outline

- Apache Airavata Overview
- Microservice Architectures
- DevOps/NoOps Principles
- Overview of Project Themes
- Project Themes 1 & 2

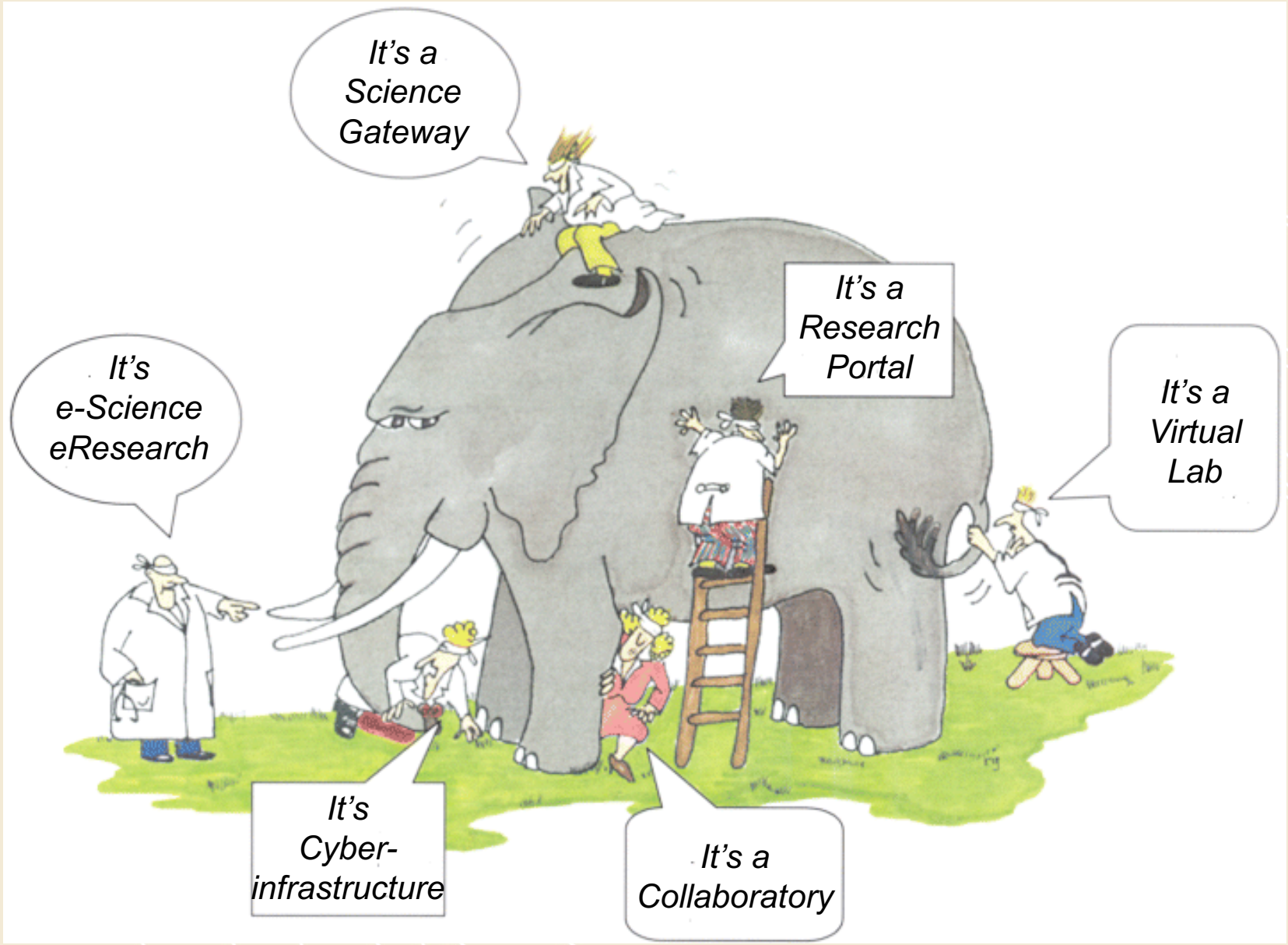
What is a science gateway?

science gateway /sī′ əns gāt′ wā′/ *n.*

1. an online community space for science and engineering research and education.
2. a Web-based resource for accessing data, software, computing services, and equipment specific to the needs of a science or engineering discipline.



Source: Nancy Wilkins-Diehr



Source: Nancy Wilkins-Diehr

For this class, Science Gateway is:

- Implement distributed systems to enable Science.
- Adopt architectural patterns used by web-scale companies.
- Usage and growth of gateways is elastic.
- Tie together diverse, heterogeneous Computational Resources (analogous to multiple cloud vendors).
- Have to be very secure (science is proprietary at least until published/patented/tenured)

Anatomy of a Science Gateway

- Gateway User Interface
  - Web Portals
  - Desktop Clients
  - Social/ Collaboration Capabilities
- Security Infrastructure
- Execution Frameworks
  - Application Abstraction
  - Workflow construction & Enactment
  - Compute Resource Management
  - Scheduling
  - Messaging System
- Data Services
  - Replica Management
  - Provenance
  - Analyses
  - Reproducibility
  - Visualization

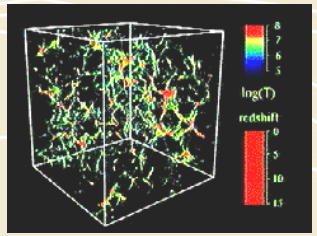
What is Supercomputing?

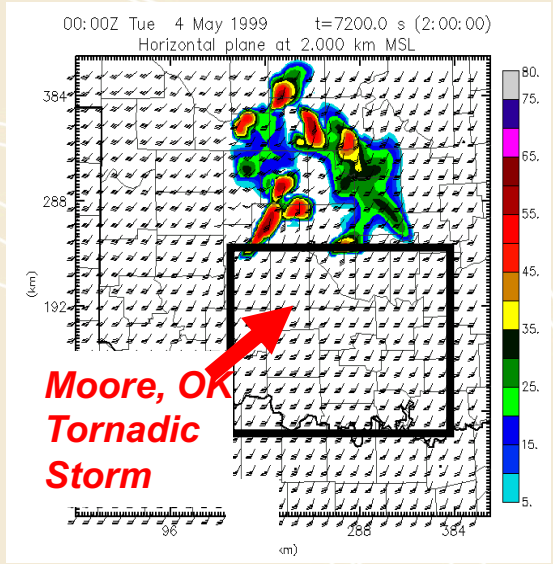
- ***Supercomputing*** is the **biggest, fastest computing right this minute**.
  - So, the definition of supercomputing is **constantly changing**.
  - GigaFLOPS, TeraFLOPS, PetaFLOPS, ExaFLOPS…...
- Supercomputing is also heard in context of :
  - *High Performance Computing* (HPC),
  - *High End Computing* (HEC),
  - Grid Computing
  - *Cyberinfrastructure* (CI).

Source: Supercomputing in Plain English, Henry Neeman, University of Oklahoma

What is HPC (hence Science Gateways) Used For?

- ***Simulation*** of physical phenomena, such as
  - Weather forecasting
  - Galaxy formation
  - Oil reservoir management





- ***Data mining***: finding **needles** of information in a haystack of data, such as
  - Gene sequencing
  - Signal processing
  - Detecting storms that might produce tornados
- ***Visualization***: turning a vast sea of **data** into **pictures** that a scientist can understand

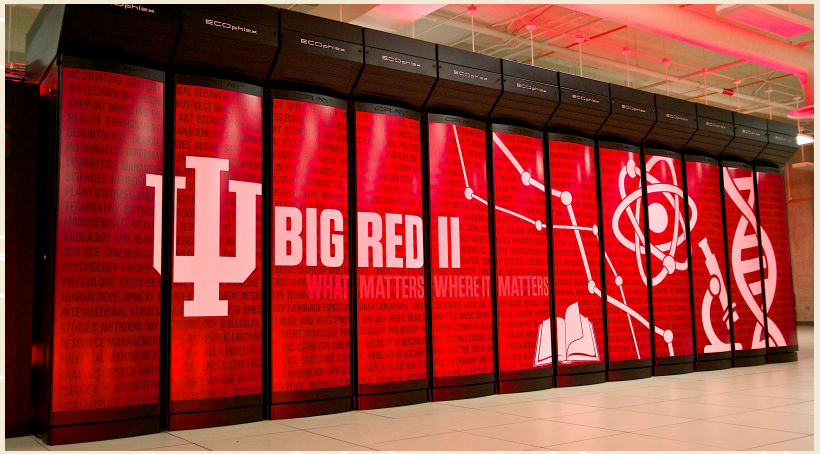
Source: Supercomputing in Plain English, Henry Neeman, University of Oklahoma

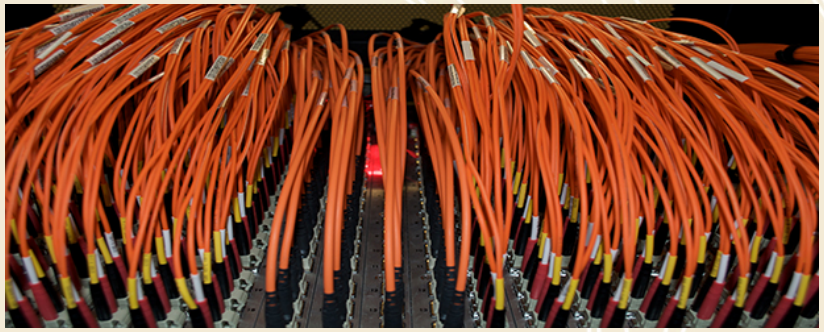
What a Cluster is ….

A cluster is a collection of small computers, called ***nodes***, hooked together by an ***interconnection network***.

It also **needs** software that allows the nodes to communicate over the interconnect.

A cluster **is** … is all of these components working together as if they're one big computer ... a **super** computer.

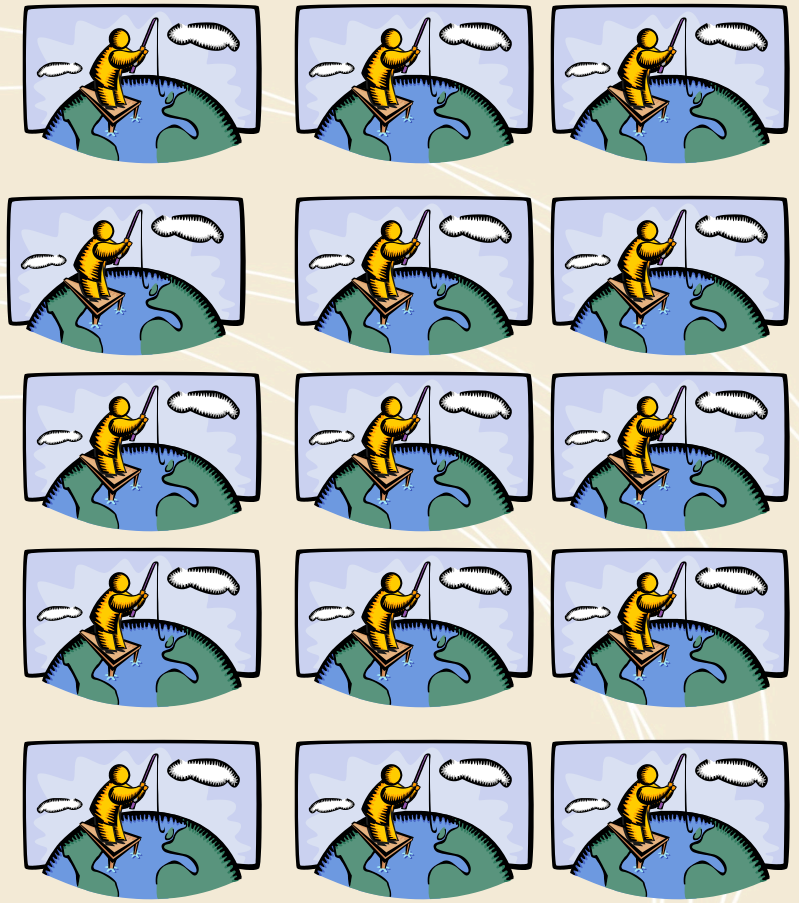




Source: Supercomputing in Plain English, Henry Neeman, University of Oklahoma

Parallelism

Parallelism *means doing multiple things at the same time: you can get more work done in the same time.*



More fish!

Source: Supercomputing in Plain English, Henry Neeman, University of Oklahoma

What is Parallelism?

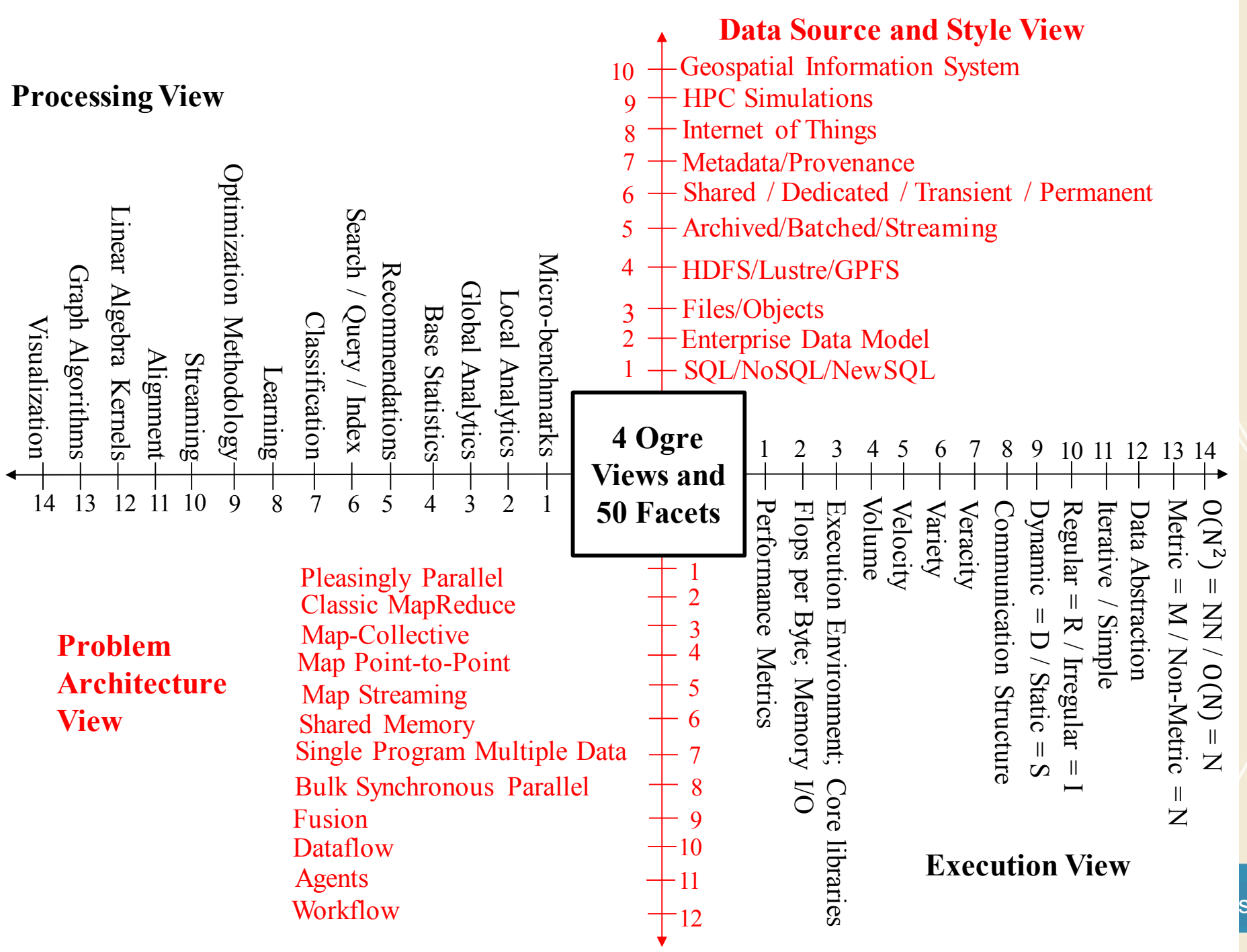
Parallelism is the use of multiple processing units – either processors or parts of an individual processor – to solve a problem, and in particular the use of multiple processing units operating concurrently on different parts of a problem.

The different parts could be different tasks, or the same task on different pieces of the problem's data.

Source: Supercomputing in Plain English, Henry Neeman, University of Oklahoma

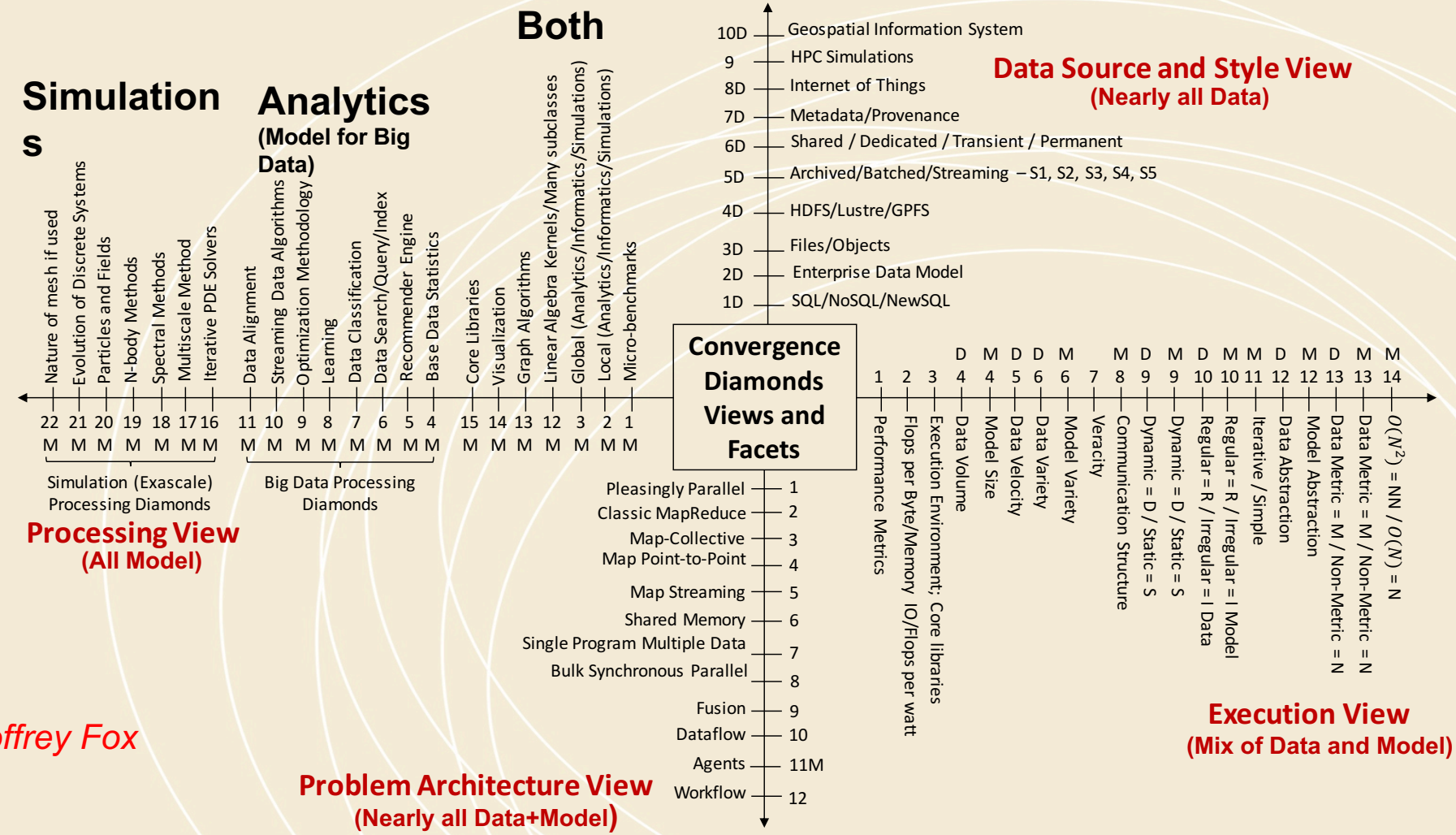
Berekely Dwarfs

- https://www2.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.pdf



Source: Prof. Geoffrey Fox

64 Features in 4 views for Unified Classification of Big Data and Simulation Applications



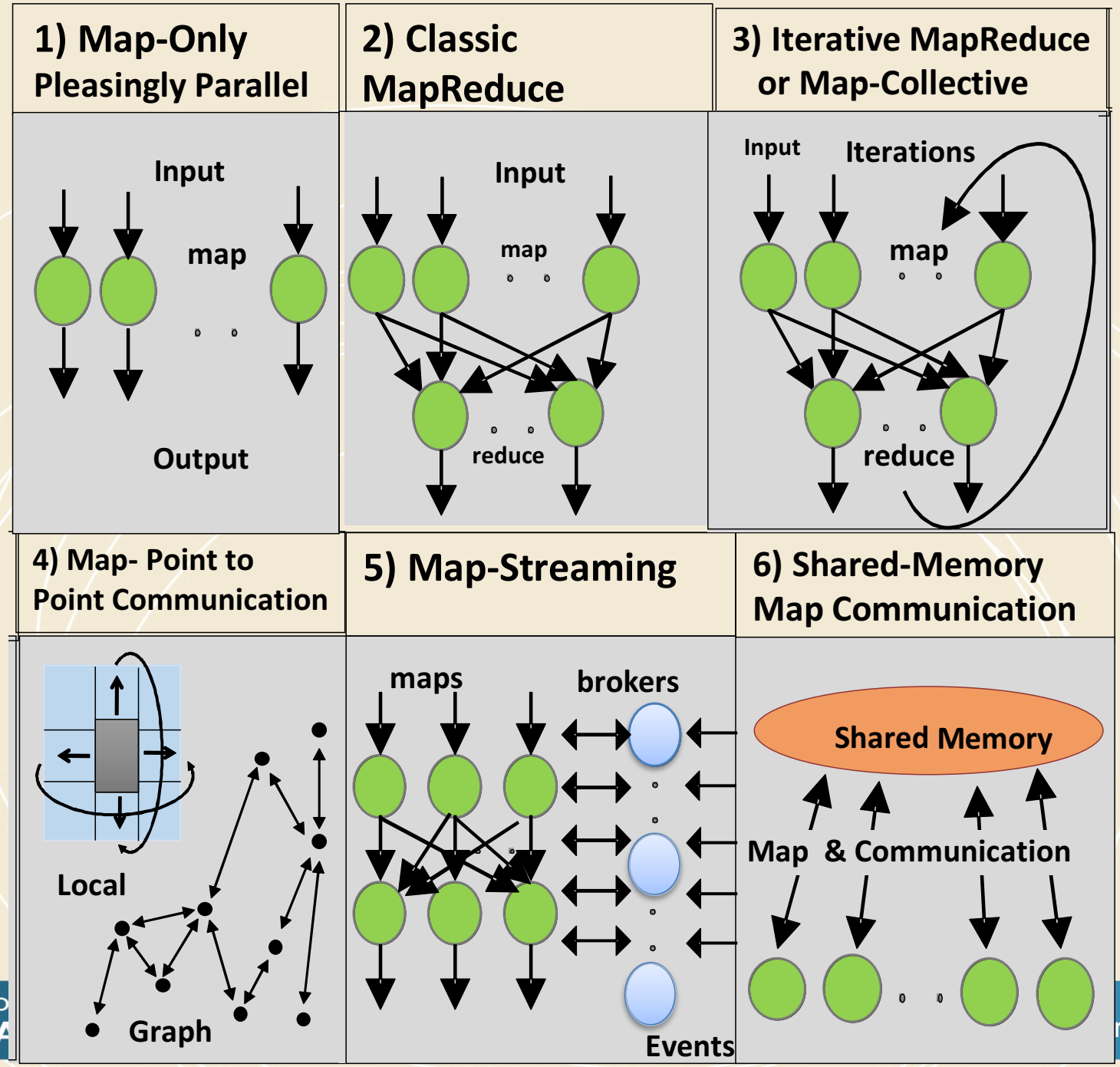
Source: Prof. Geoffrey Fox

6 Forms of MapReduce

Describes Architecture of
- Problem (Model reflecting data)
- Machine
- Software

2 important variants (software) of Iterative MapReduce and Map-Streaming
a) "In-place" HPC
b) Flow for model and data

Source: Prof. Geoffrey Fox



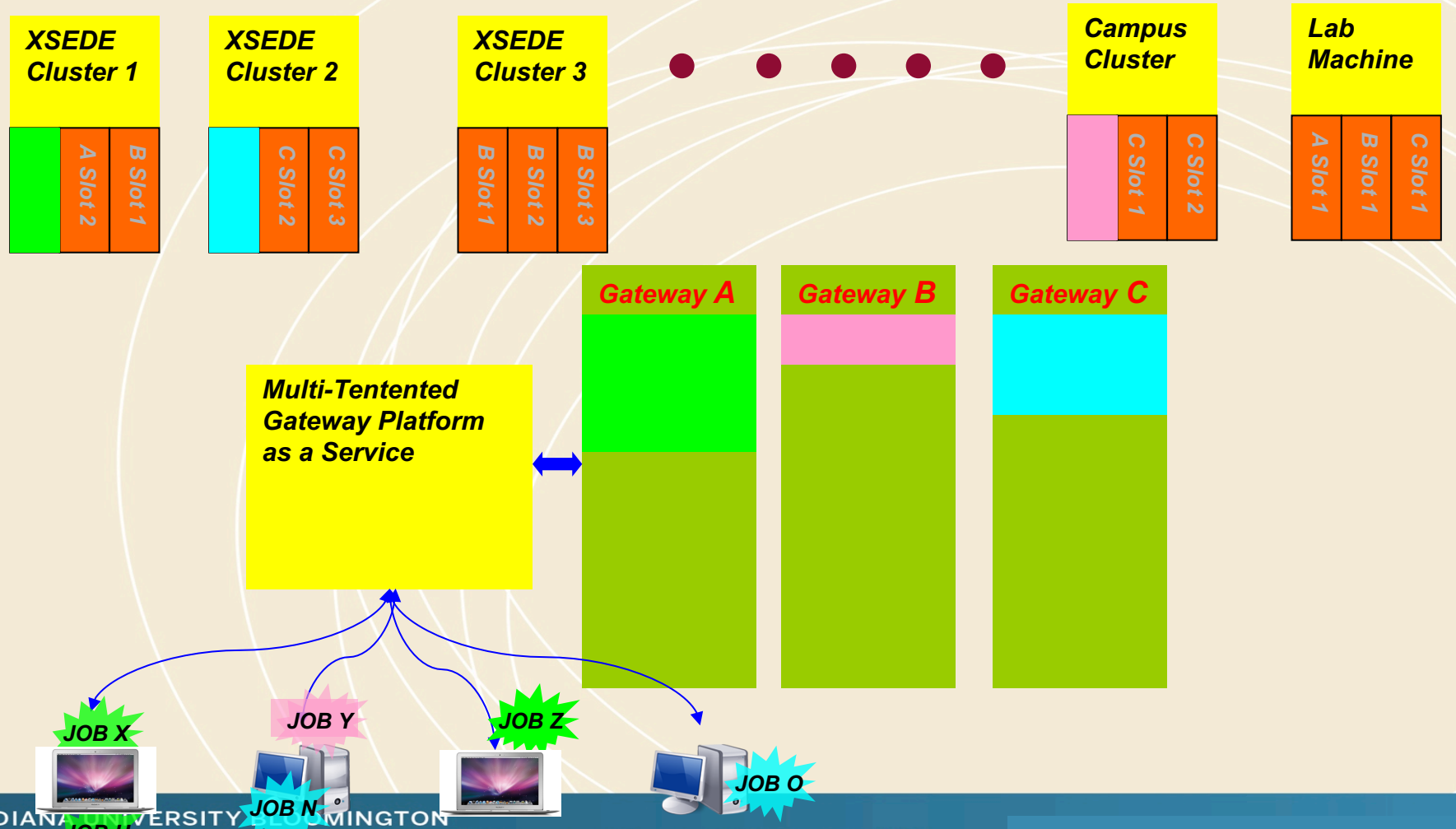
Gateways *abstract* these details from Users

- Parallelism Types
  - Instruction Level Parallelism
  - Shared Memory Multithreading
  - Distributed Memory Multiprocessing
  - GPU Parallelism
  - Hybrid Parallelism (Shared + Distributed + GPU)
- Gateways should pick the right computations paradigm based on the problem.

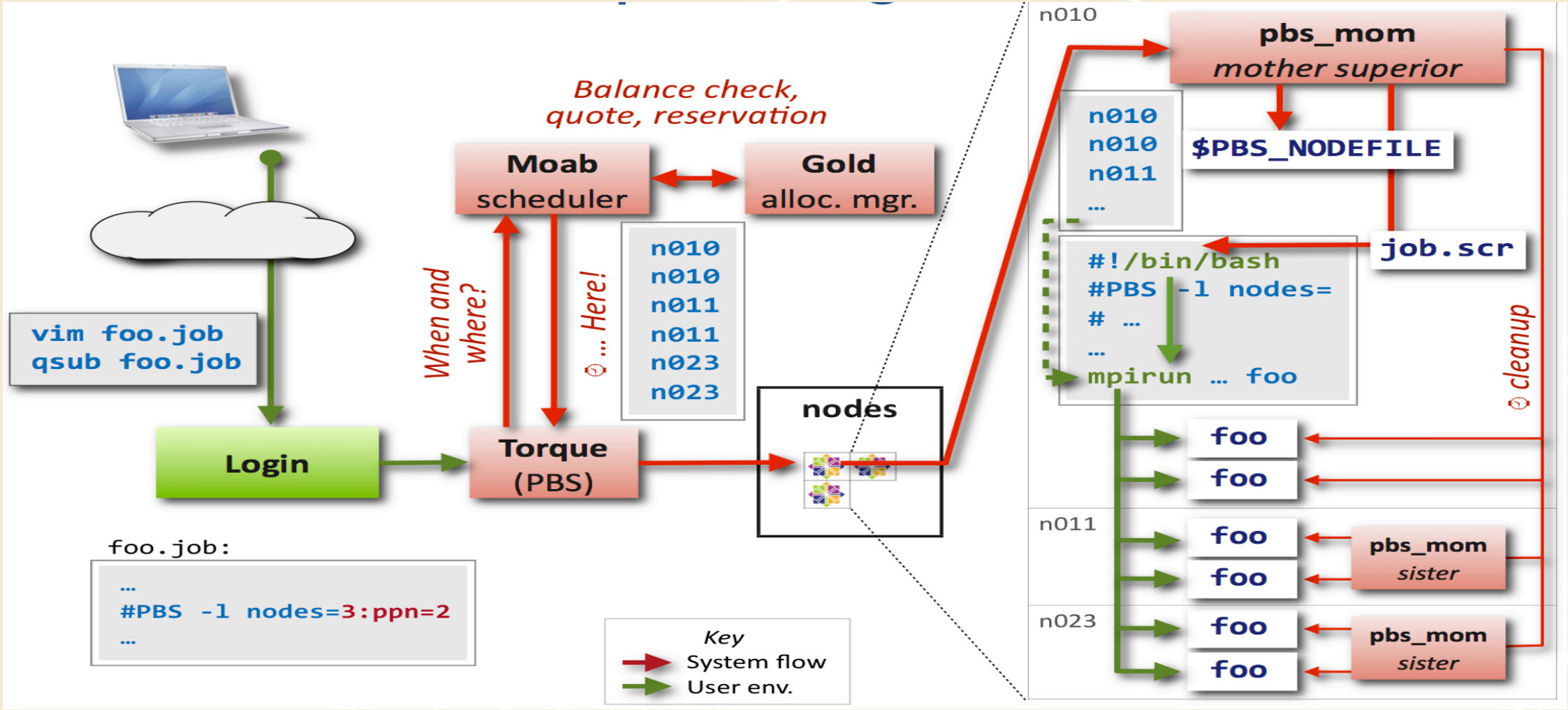
Science Gateways

Unleashing Supercomputing and Cloud to broad range of Science and Engineering users

Example: Science Gateways Federate Jobs across clusters



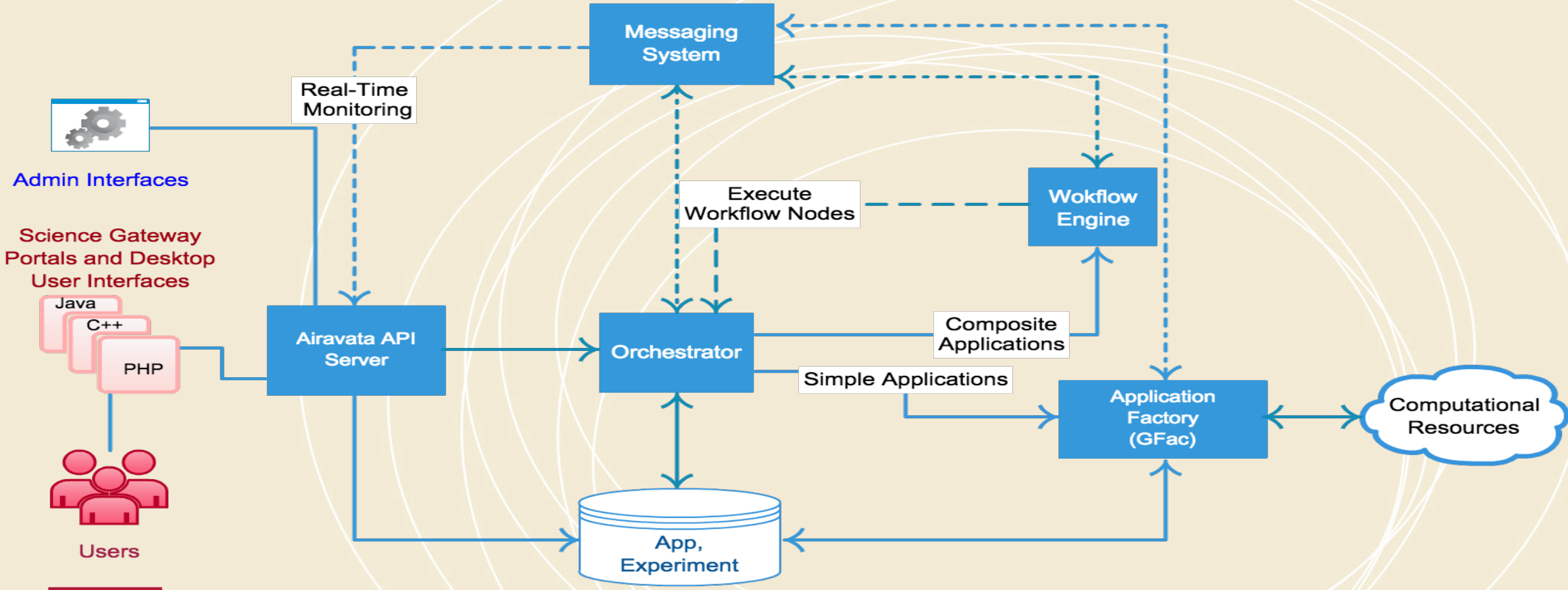
Job Managers Flow



Source: https://wiki.anl.gov/cnm/HPC/Submitting_and_Managing_Jobs

Apache Airavata

Airavata: Multi-Tentanted Gateway Middeware



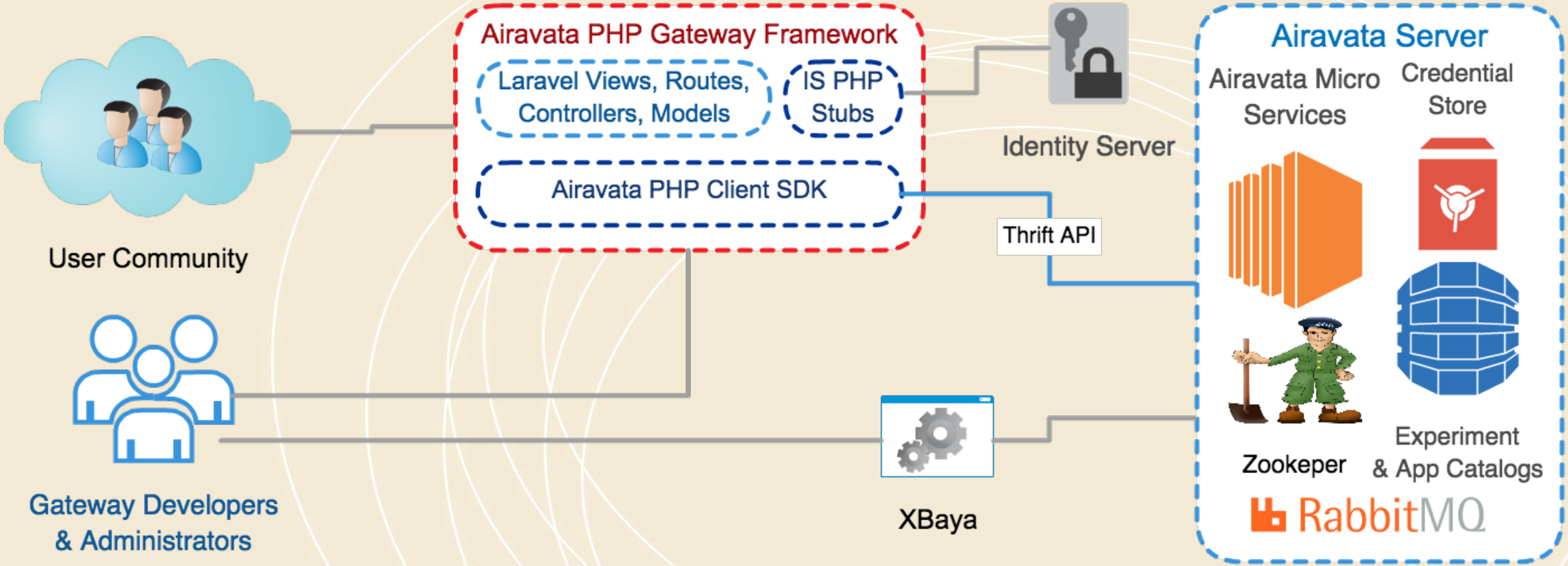
Airavata Overview

- Airavata is a general purpose distributed system software framework build on micro-service and component based architecture principles.
- Airavata provides capabilities to compose, manage, execute and monitor large scale applications and workflows on distributed computing resources.
- Airavata supports executions on local clusters, national grids, academic and commercial clouds.
- Airavata is inherently multi-tenanted.
- Airavata is increasingly evolving to support 4th Paradigm – “Data-Intensive Scientific Discovery”.

Airavata as a Science Gateway Middleware

- Airavata is dominantly used to build science gateways.
- Airavata supports secured communications to HPC resources and empowers gateway operators to administer and monitor long running executions.
- A reference PHP based gateway is provided to illustrate the Airavata capabilities and can be used to customize science-centric gateways

Reference Gateway <-> Airavata Services



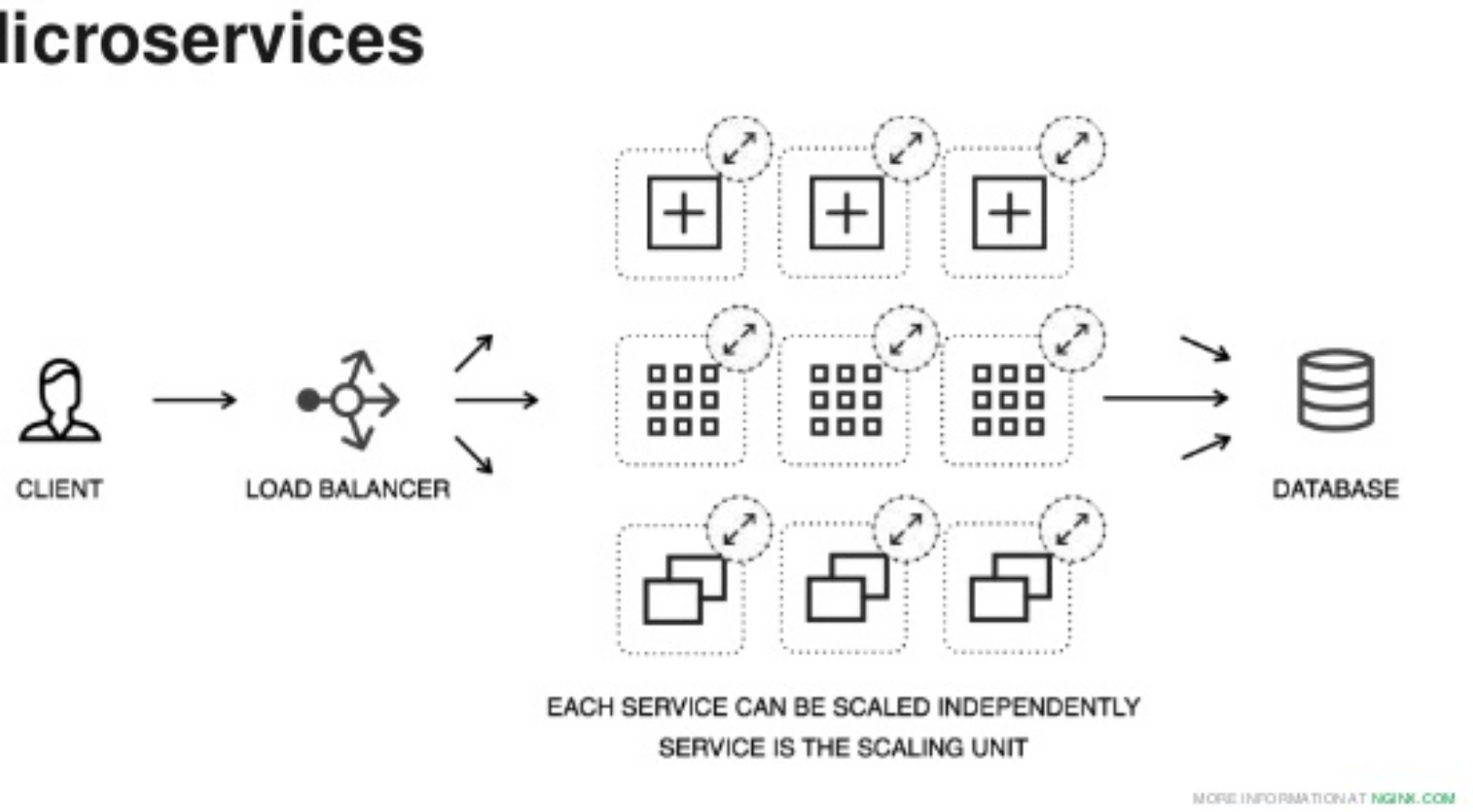
SciGaP – Powered by Airavata

- Science Gateway Platform as a Service (SciGaP) provides application programmer interfaces (APIs) to hosted generic infrastructure services that can be used by domain science communities to create Science Gateways.
- SciGaP hosted service platform is powered by Apache Airavata.
- SciGaP helps gateway developers to concentrate their efforts on building their scientific communities and not worry about operations.

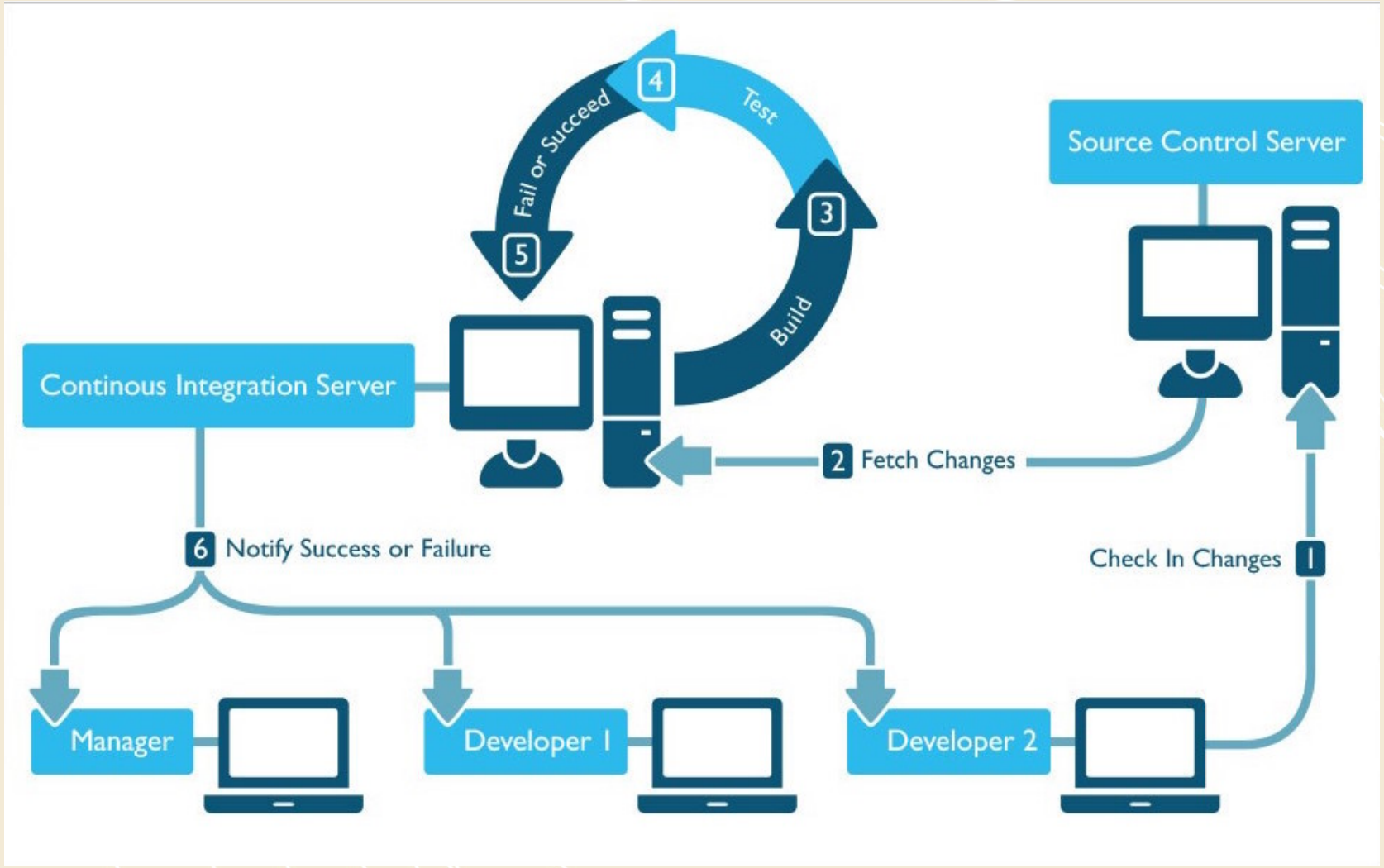
Challenges for Gateways

- Providing a rich user experience
- Defining an API for the application server
- Defining the right sub-components for the application server.
- Implementing the components, wiring them together correctly.
- Supporting multiple gateway tenants
- Fault tolerance for components
- State management
- Continuous delivery
- Security management
- Supporting full scientific exploratory cycle

Component Microservices

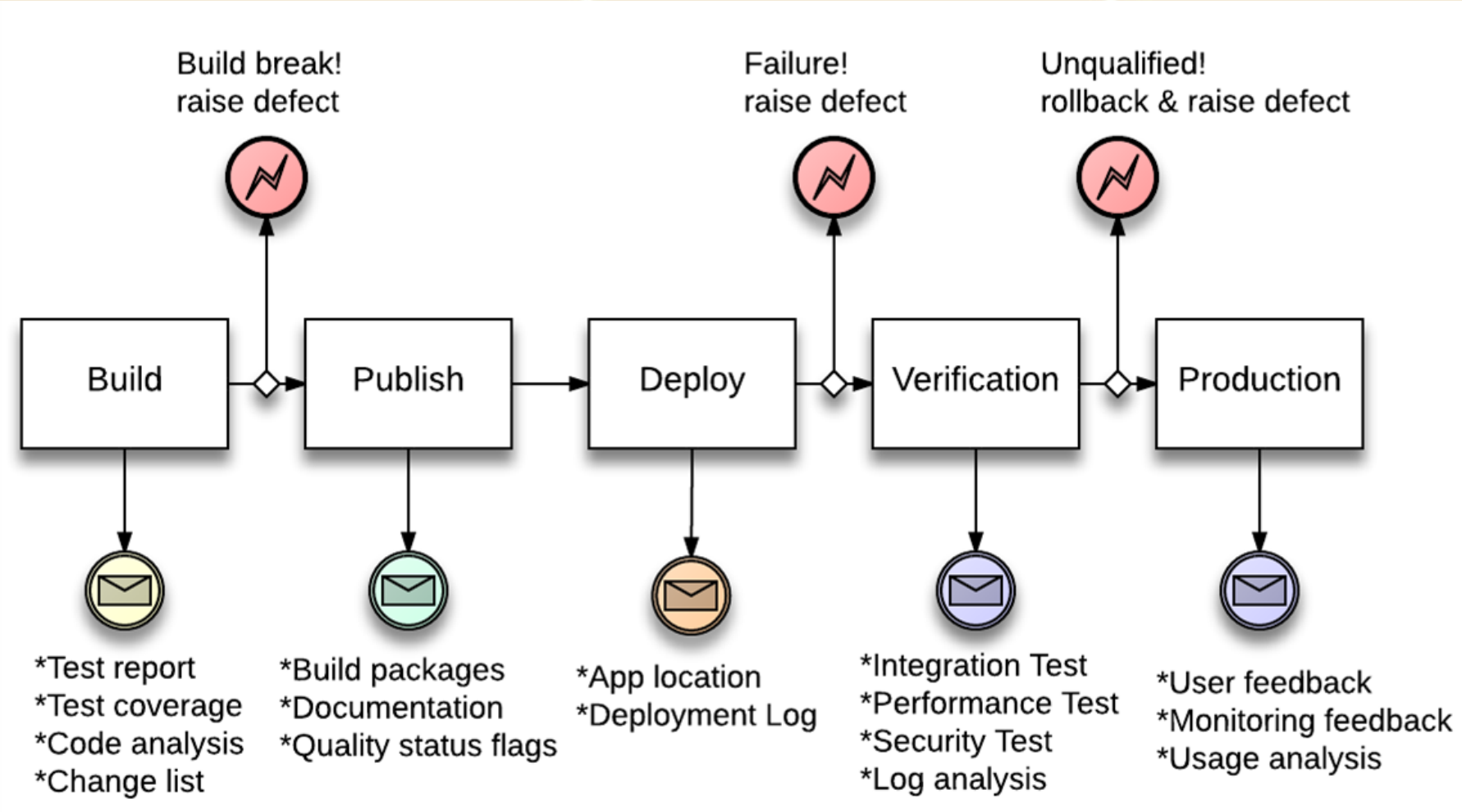


Continuous Integration



Source: https://insights.sei.cmu.edu/devops/2015/01/continuous-integration-in-devops-1.html

Continuous Deployments/Delivery



Source: https://www.ibm.com/developerworks/community/blogs/c914709e-8097-4537-92ef-8982fc416138/entry/devops_in_practice_best_practices_for_adopting_continuous_delivery?lang=en

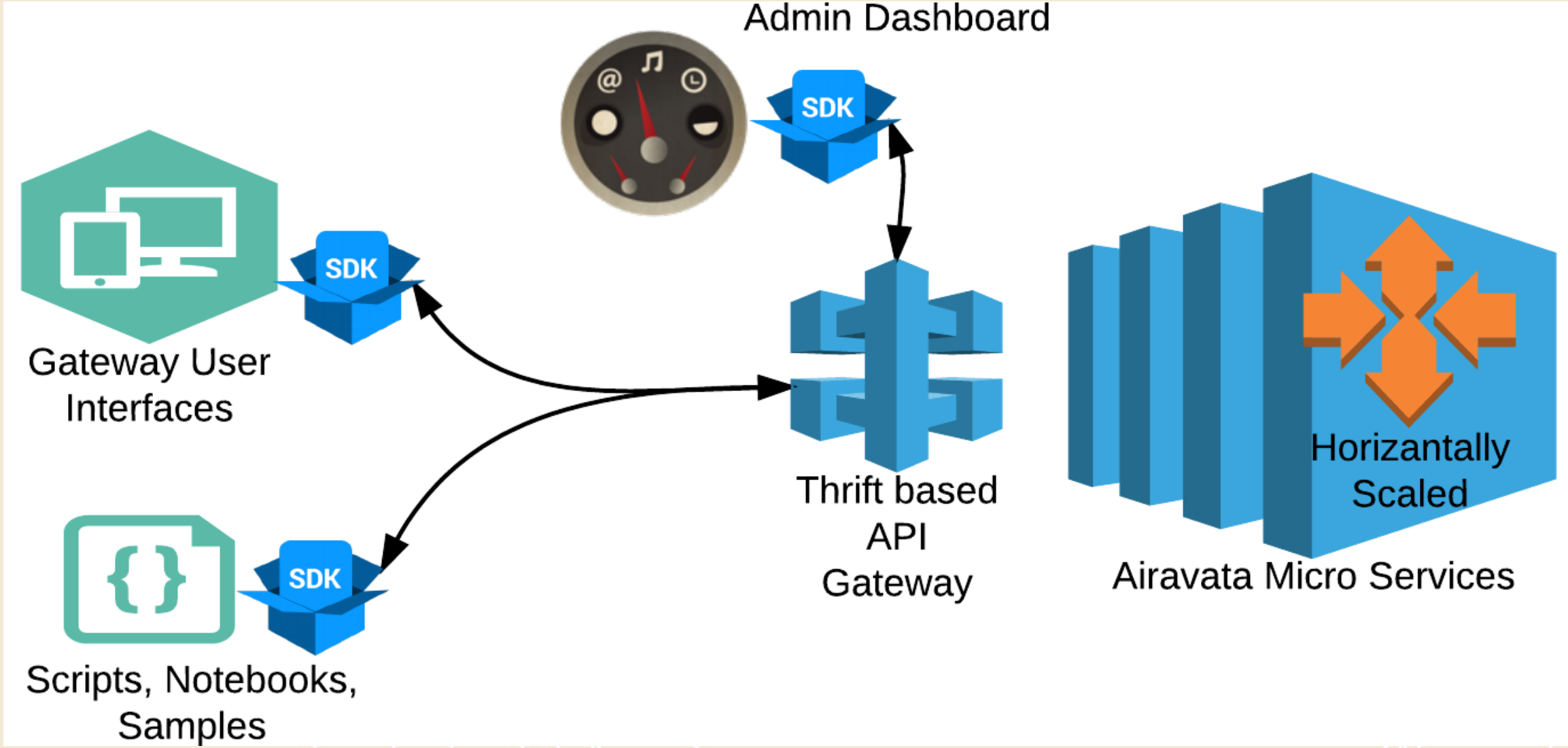
Projects Overview

First 2 week Projects

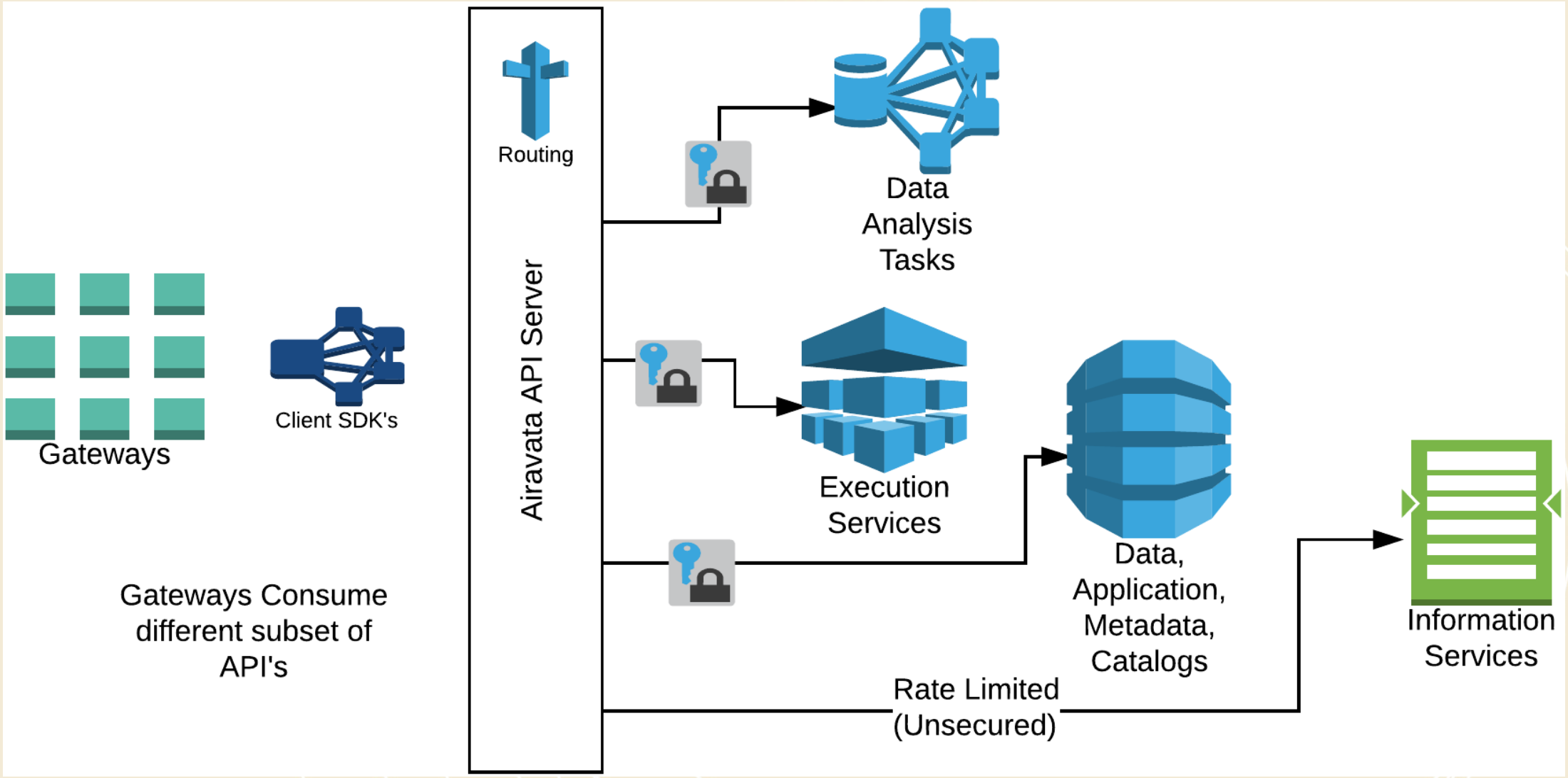
Will have two themes

January 13th 2017 – January 27th 2017

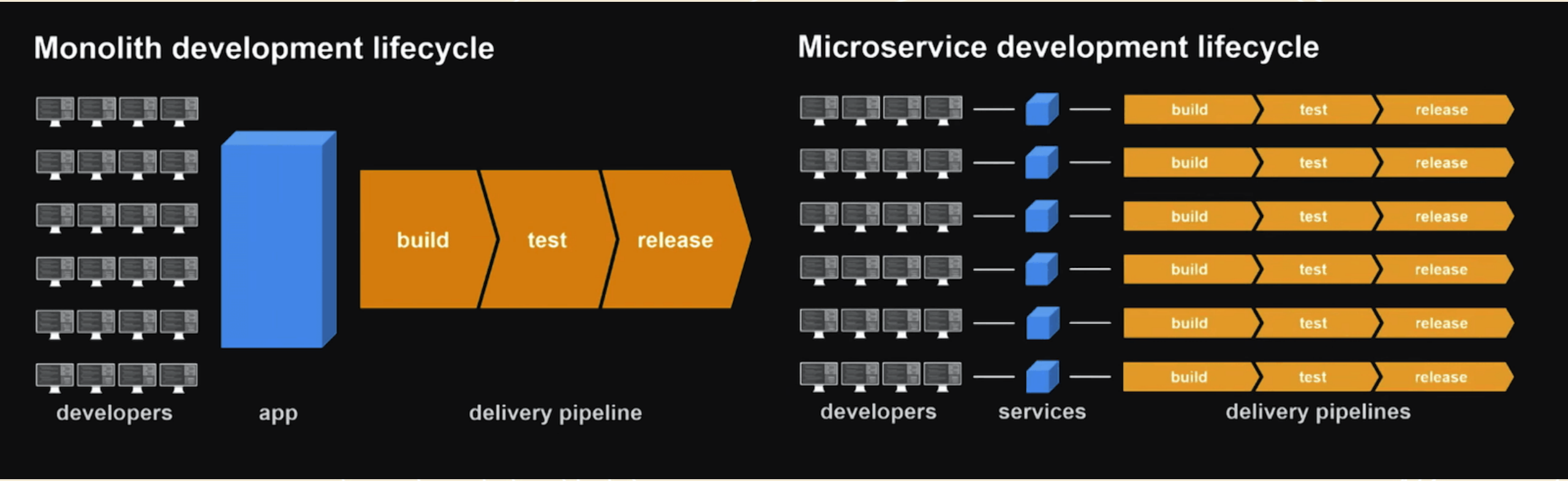
Phase 1 Projects



Phase 2 Projects



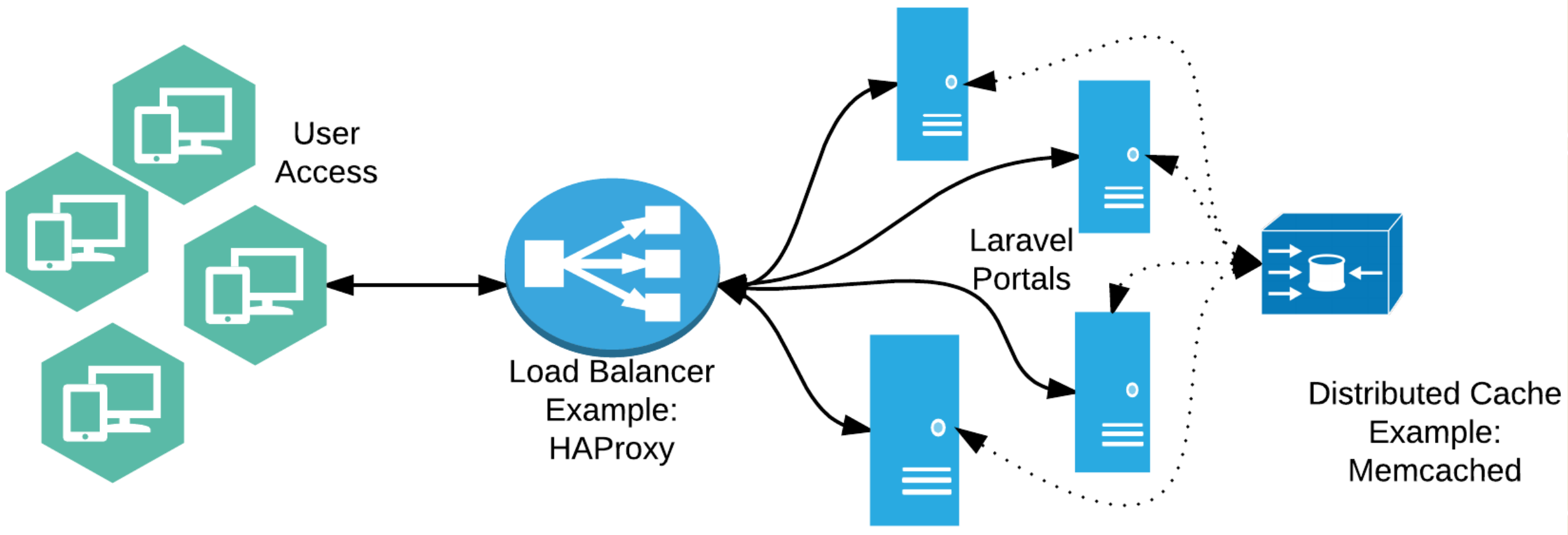
Phased Releases



Project Theme 1: Scalable Web Portal

- Load balance Laravel Portal.
- Fail Over to other regions in Amazon.
- Use distributed cache for session and security management.
- Easily Deployable.
- Have a Build and make it IDE and Developer friendly
- Facilitate micro-feature releases
- Connect laravel Portal with Thrift API Server

First 2 weeks Goal 1:



Example Test Case:

- Have multiple gateway instances:
  - sneha.scigap.org, marcus.scigap.org, mayank.scigap.org, eldho.scigap.org, ameya.scigap.org, anuj.scigap.org, sagar.scigap.org……..
  - Each of these portals have their own look and feel and functionality.
  - They all can be hosted from one web server.
- How can we scale the number of gateways.
- How can we not be nervous about hardware failures?
- How can we move from “pets to cattle’s”?

"Pets vs Cattle" (Yes, again)

Scale Up

- Servers are like pets.

Pets are given names, are unique, lovingly hand raised and cared for. When they get ill, you nurse them back to health

- Servers are like cattle.

Cattle are given numbers and are almost identical to each other. When they get ill, you get another one.

> "Future application architectures should use Cattle but Pets with strong configuration management are viable and still needed"
> - Tim Bell, CERN

The above adapted from Tim Bell, CERN
http://www.slideshare.net/noggin143/20121017-openstack-cern-accelerating-cience

Project Theme 1: Load Balance API Gateway

- Thrift based multiplexed API Gateway
- Fault Tolerant
- Load Balanced
- Scalable
- Pluggable
- Facilitate Overlaid Features

Timeline

- Next Friday Jan 20th
  - Present design choices
  - Discuss Hurdles
- Following Friday Jan 27th
  - Demonstrate Scalable, Reliable, Fault Tolerant Portal and API Server

Thoughts?

Marlon Pierce, Suresh Marru

{marpierc, smarru}@iu.edu