LUDDY
SCHOOL OF INFORMATICS,
COMPUTING, AND ENGINEERING

CSCI-B 649 Topics in Systems:
Applied Distributed Systems

Spring 2021 Course Introduction

January 19th 2021

Suresh Marru, Marlon Pierce

Cyberinfrastructure

- “Cyberinfrastructure consists of computing systems, data storage systems, advanced instruments and data repositories, visualization environments, and people, all linked by high speed networks to make possible scholarly innovation and discoveries not otherwise possible.”
  - In short, it typically refers to information technology research systems that democratizes particularly powerful and advanced capabilities such as supercomputers.

Cyberinfrastructure Integration Research Center (CIRC)

CIRC's core mission is to accelerate research, discovery and collaboration through the creation, integration and operation of user-centric cyberinfrastructure that benefits scientific communities.

Course Instructors

Marlon Pierce

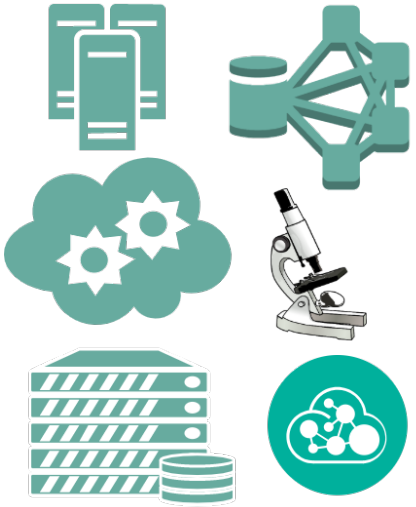
Suresh Marru

Class Introductions

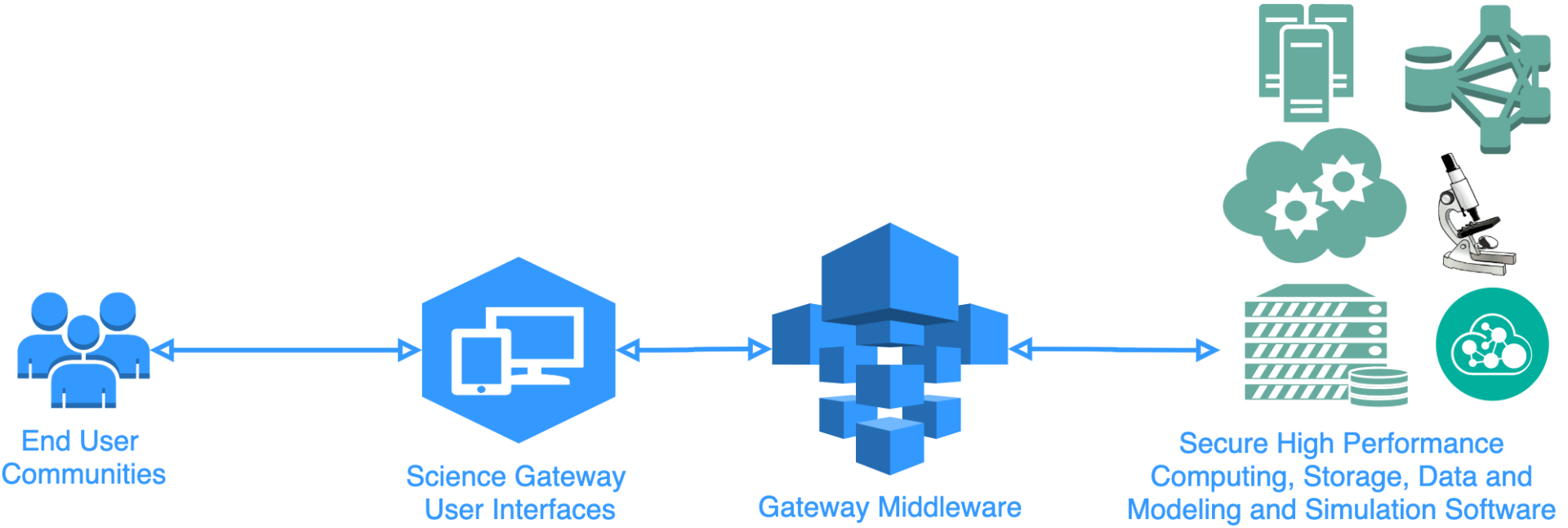
What are we expecting you to get out of this class?

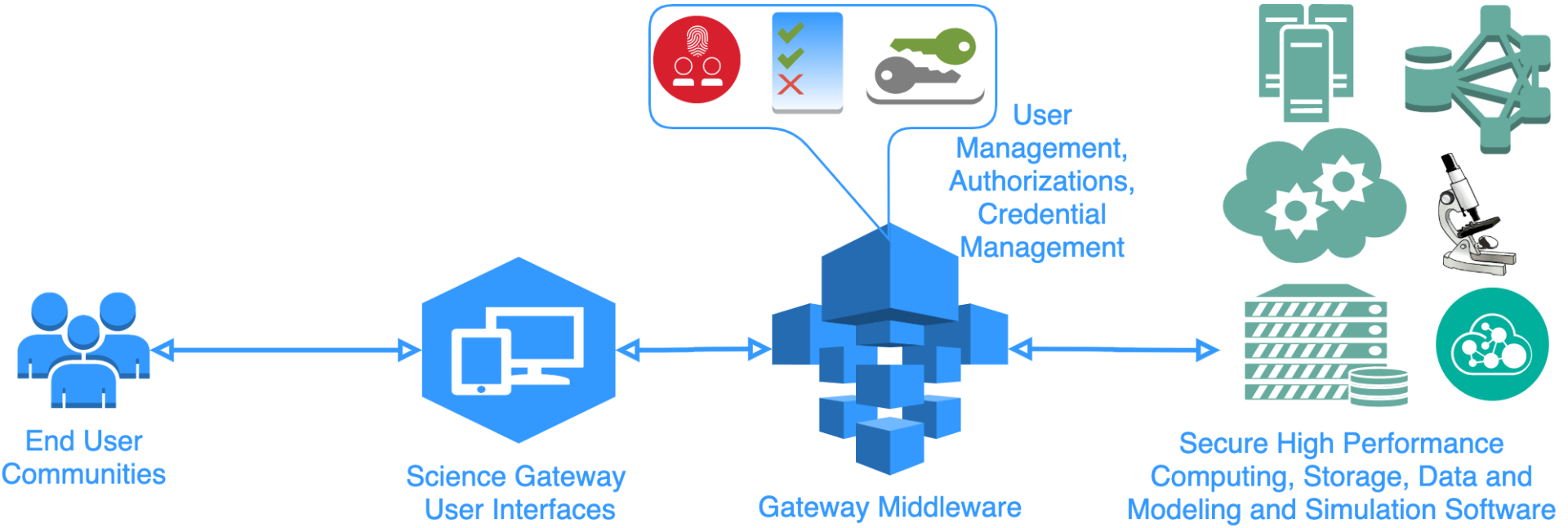
- A fusion of conceptual skills and “scientific way” of making choices.
- The course is tailored to use tools and technologies relevant in 2021 but our expectation is you will learn how to make choices not necessarily be a tutorial on a buzzy technology.
    - Our definition of a good student is someone who understand the difference between the two.

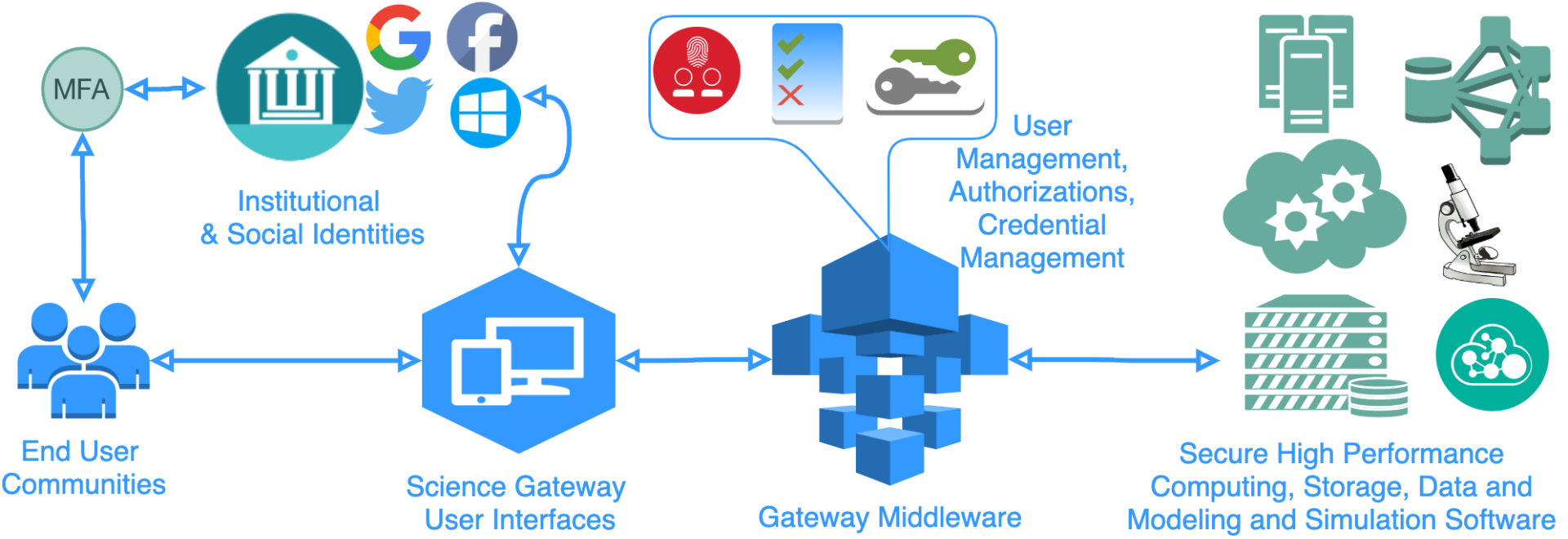
Scientists

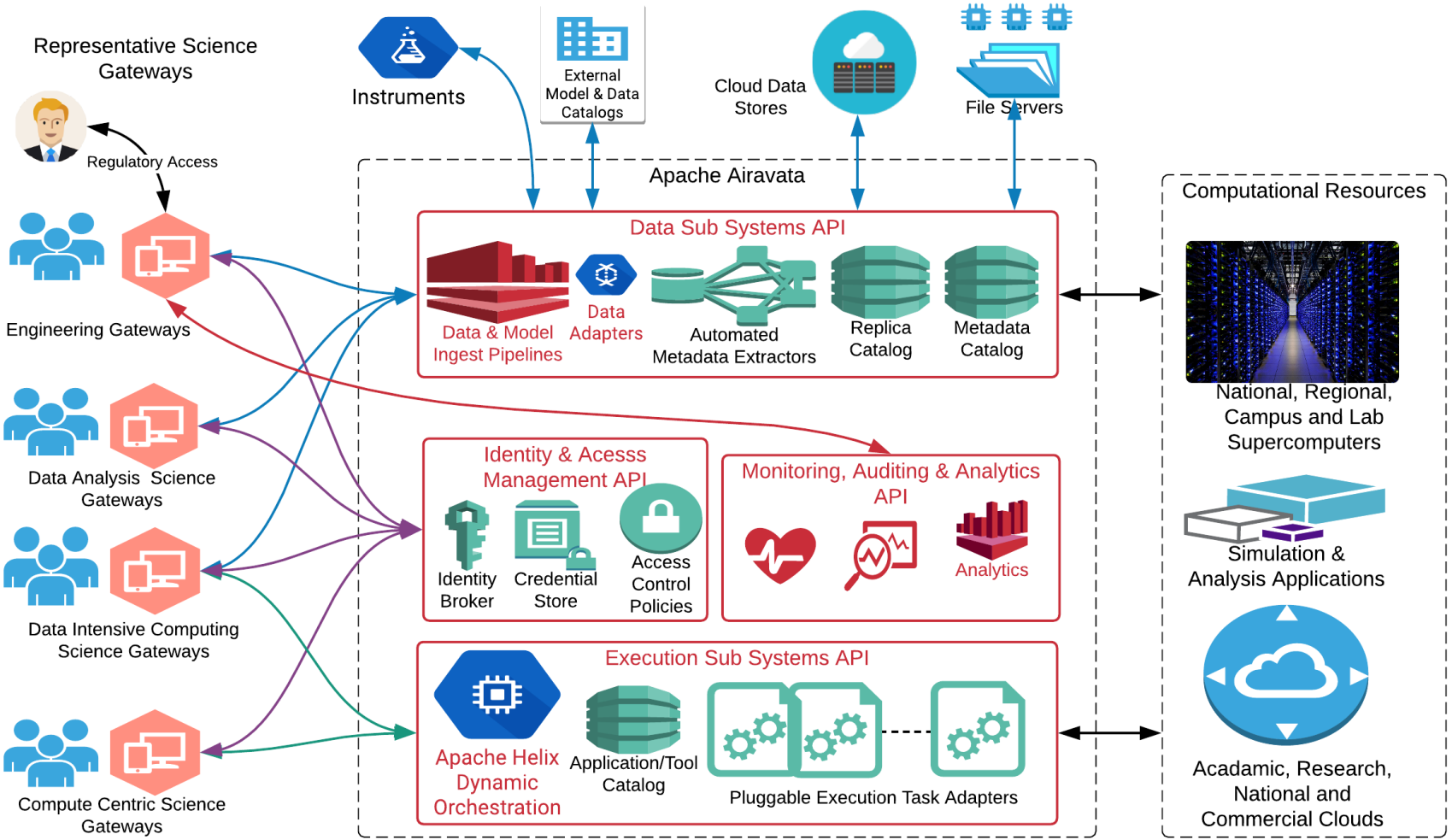


Supercomputers, Storage, Data and Modeling and Simulation Software


End User Communities
Science Gateway User Interfaces
Gateway Middleware
Secure High Performance Computing, Storage, Data and Modeling and Simulation Software


User Management, Authorizations, Credential Management
End User Communities
Science Gateway User Interfaces
Gateway Middleware
Secure High Performance Computing, Storage, Data and Modeling and Simulation Software


MFA
Institutional
& Social Identities
User
Management,
Authorizations,
Credential
Management
End User
Communities
Science Gateway
User Interfaces
Gateway Middleware
Secure High Performance
Computing, Storage, Data and
Modeling and Simulation Software


Representative Science Gateways
Regulatory Access
Instruments
External Model & Data Catalogs
Cloud Data Stores
File Servers
Apache Airavata
Data Sub Systems API
Data & Model Ingest Pipelines
Data Adapters
Automated Metadata Extractors
Replica Catalog
Metadata Catalog
Identity & Acesss Management API
Identity Broker
Credential Store
Access Control Policies
Monitoring, Auditing & Analytics API
Analytics
Execution Sub Systems API
Apache Helix Dynamic Orchestration
Application/Tool Catalog
Pluggable Execution Task Adapters
Engineering Gateways
Data Analysis Science Gateways
Data Intensive Computing Science Gateways
Compute Centric Science Gateways
Computational Resources
National, Regional, Campus and Lab Supercomputers
Simulation & Analysis Applications
Acadamic, Research, National and Commercial Clouds

Apache Airavata

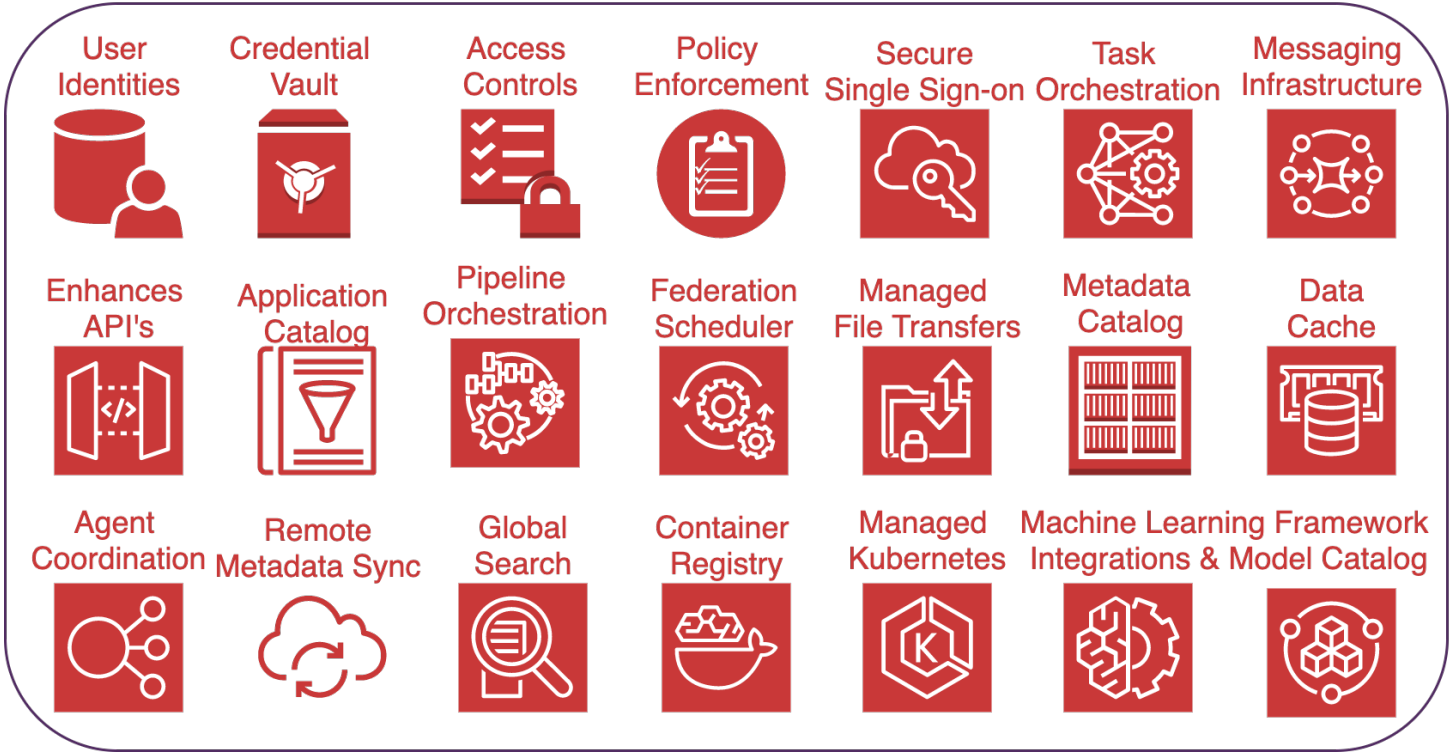
- **Gateway Developers:** Open source software for building science gateways
- **Users:** Use it to transfer data and execute remote applications and pipelines on distributed resources
- **Teams:** Create, organize, clone, and share computational experiments
- **Software Providers:** Make scientific software available as a service

Science Gateways Platform as a Service: SciGaP

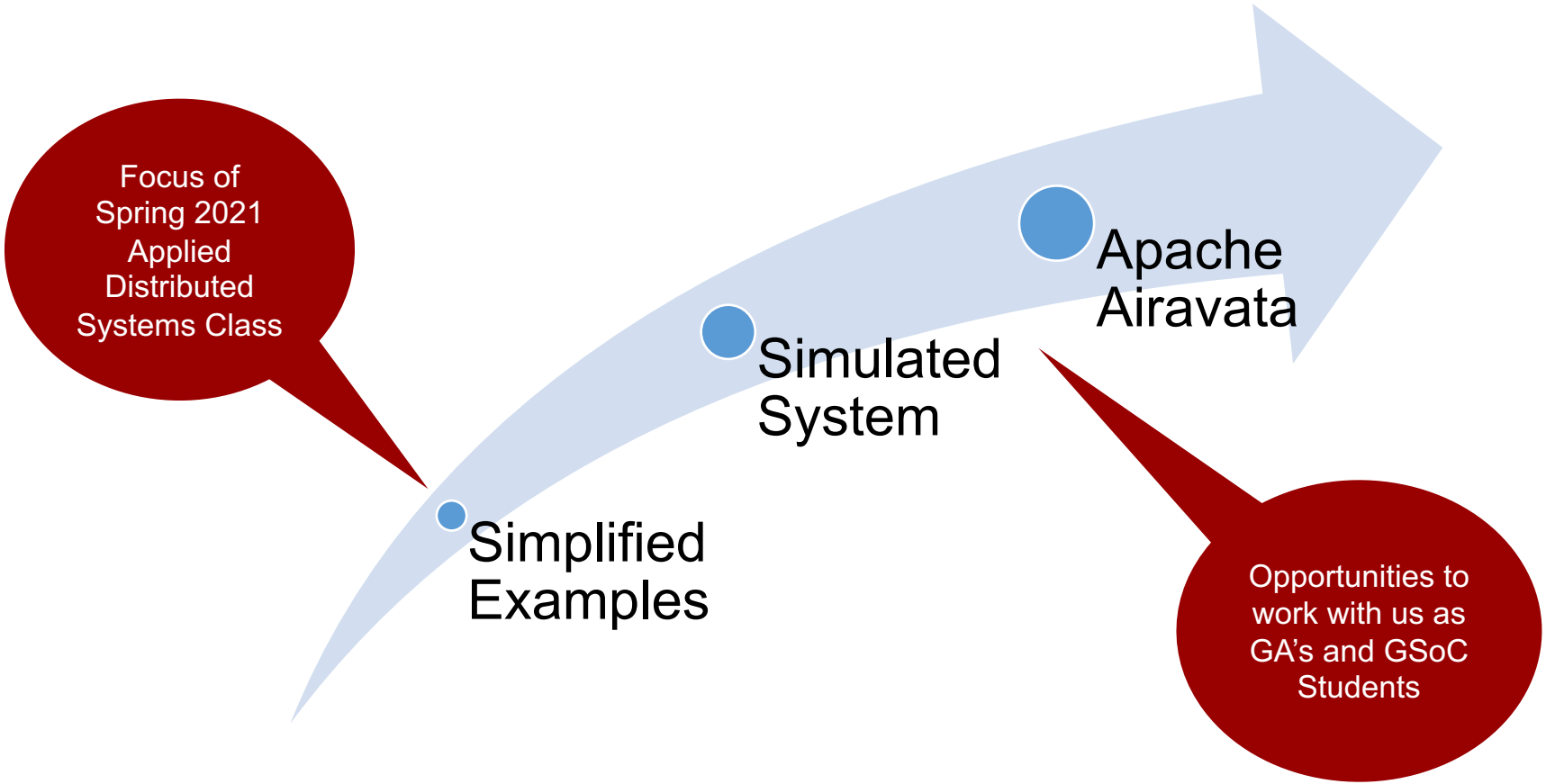
Powered By

Registered SciGaP Gateways	40+
Supported Applications	118
Integrated Supercomputers	50
Registered Users	3500+
Number of applications run (3 years)	>136,000
Computing Hours (3 years)	> 22.8 M
Resources in Countries	US, Germany, India, Australia, Canada, Finland

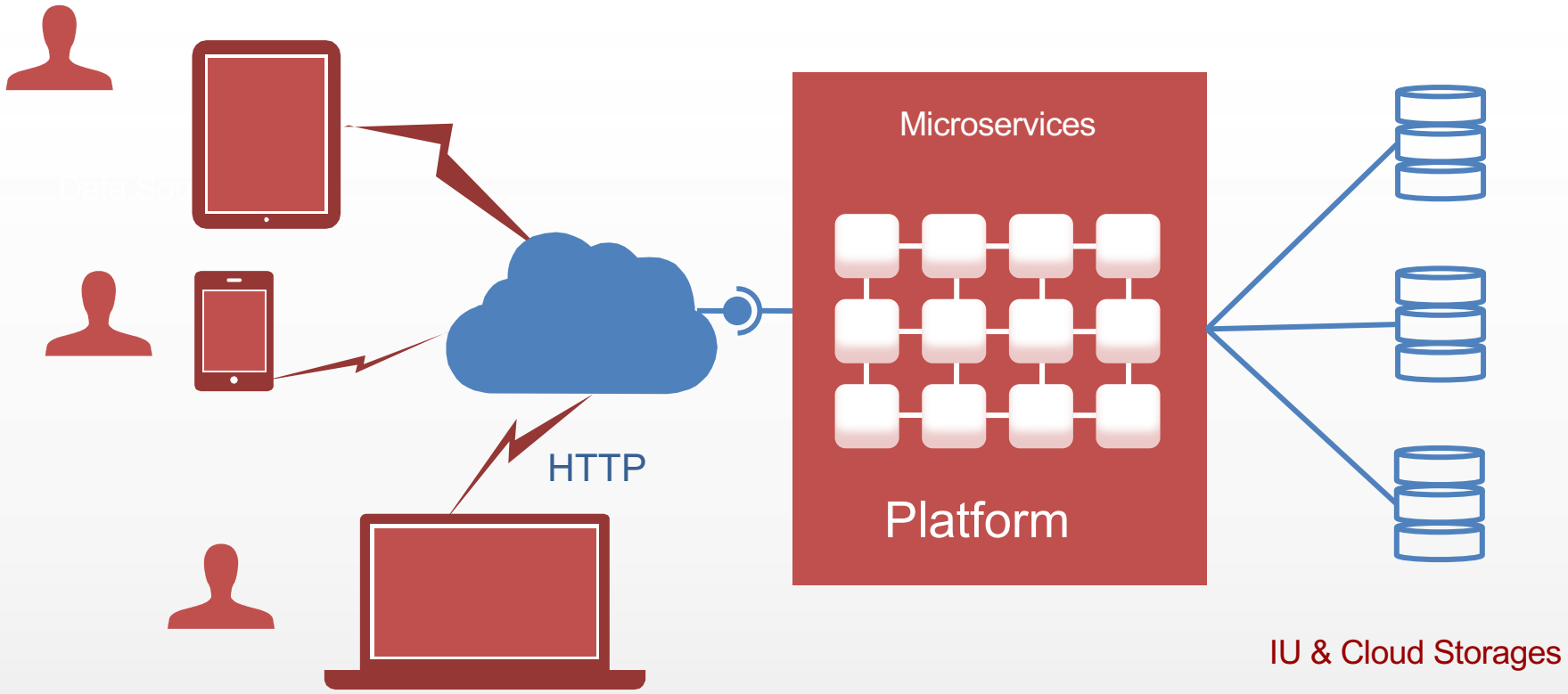
Middleware (Distributed Systems Components)



Distributed Systems Everywhere



Spring 2021 Project: A cloud storage for Photo Sharing



System Capabilities

- Login with your IU Credentials as well as social credentials (google, github)
- Upload lots of pictures
- Organize and annotate images
- Enabled controlled shared of selected collections or pictures to a user or a group
- Allow single picture download and bulk download
- Enable copying pictures to IU accessible cloud storages

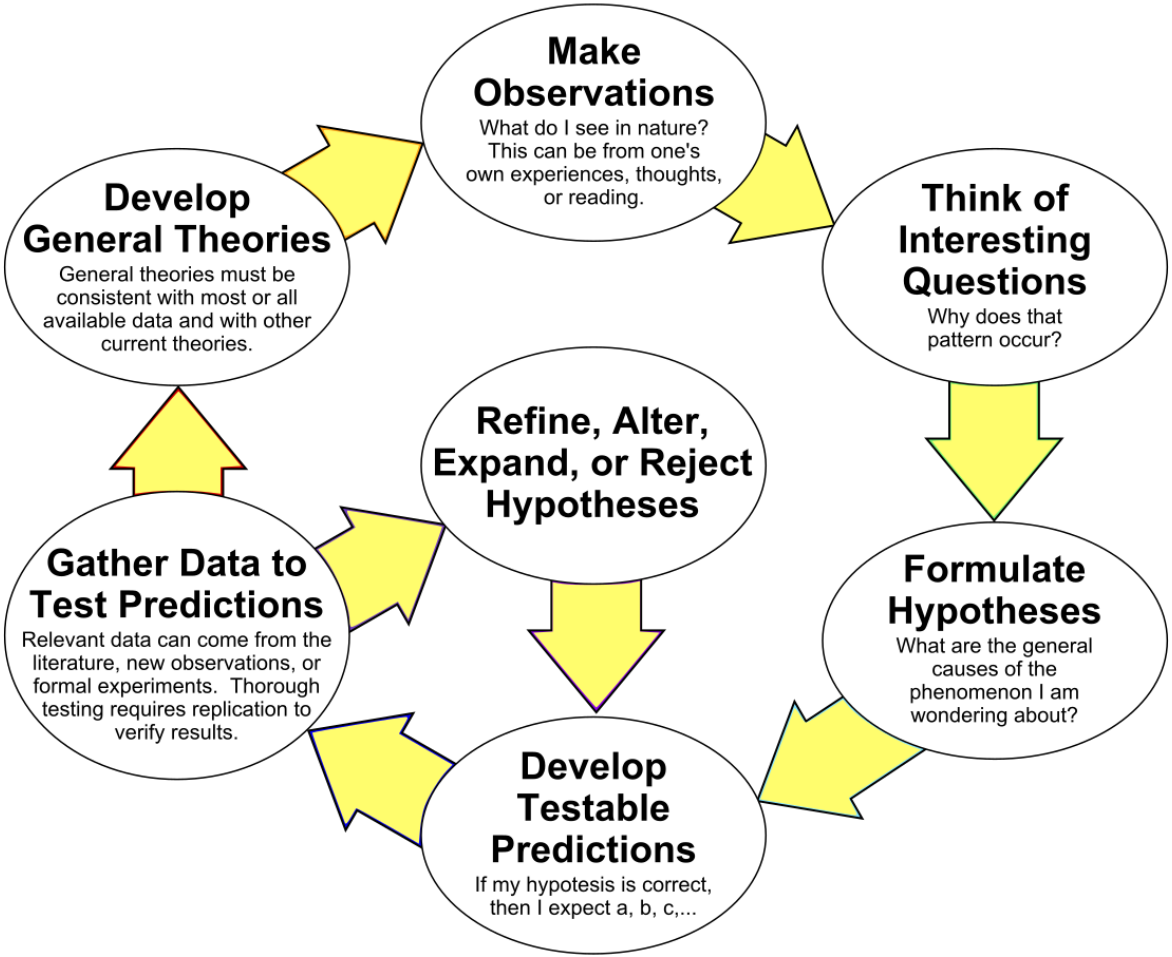
Advanced Capabilities

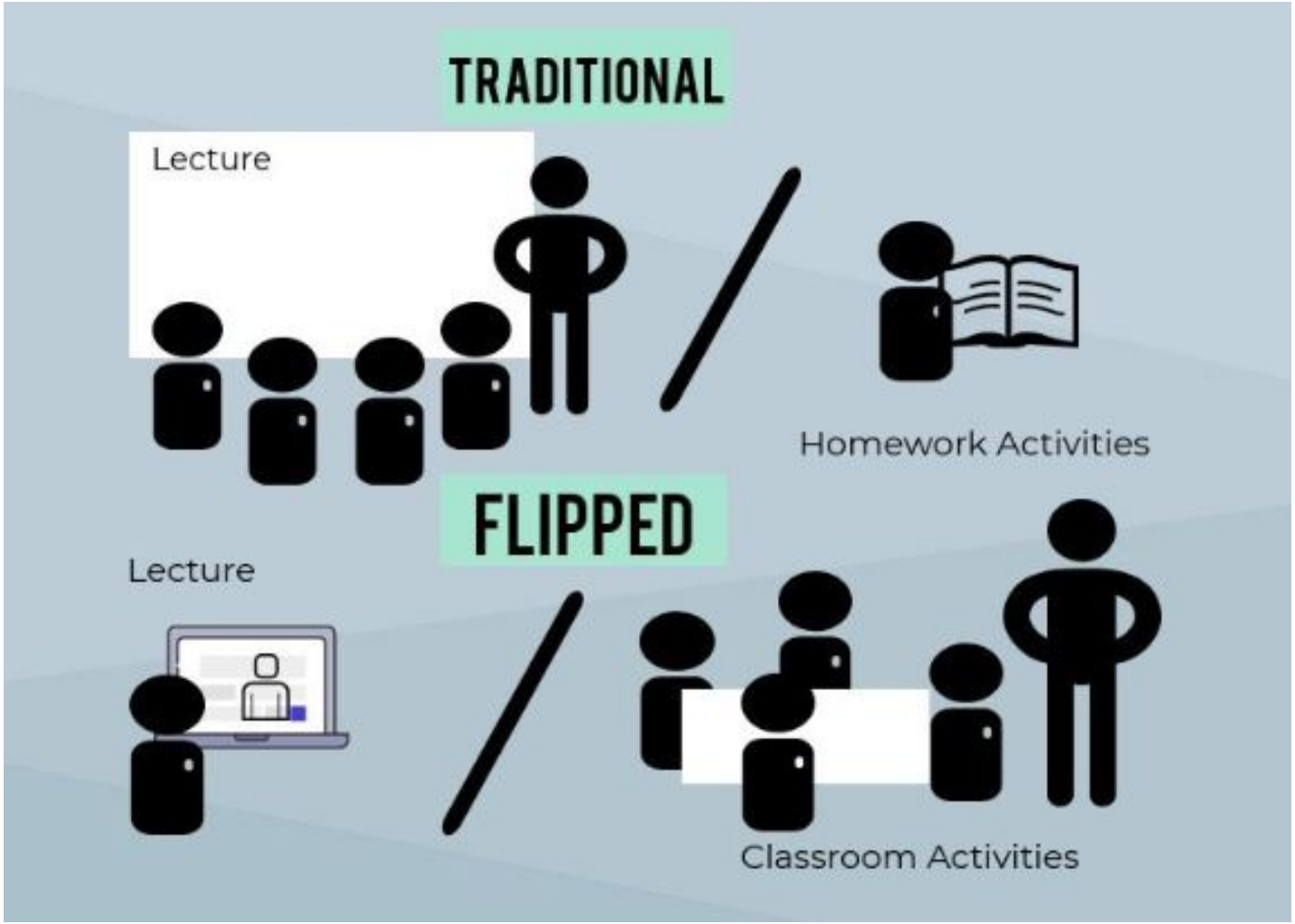
- Extract image based information/metdata
  - Use open source image parsing libraries
    - Example - https://github.com/drewnoakes/metadata-extractor
- Catalog the metadata
- Allow discovery of images with metadata search
  - Search for images with a text query
- ADVANCED: ADVANCED: enable image based searching
  - Search images to reveal copies with a known image

Applied Distributed Systems

- We will build user-centric distributed systems that mimic scientific research.
- This course will be project-based.
- You will build distributed systems.

The Scientific Method as an Ongoing Process





Source: https://teach.ufl.edu/resource-library/flipped-classroom/

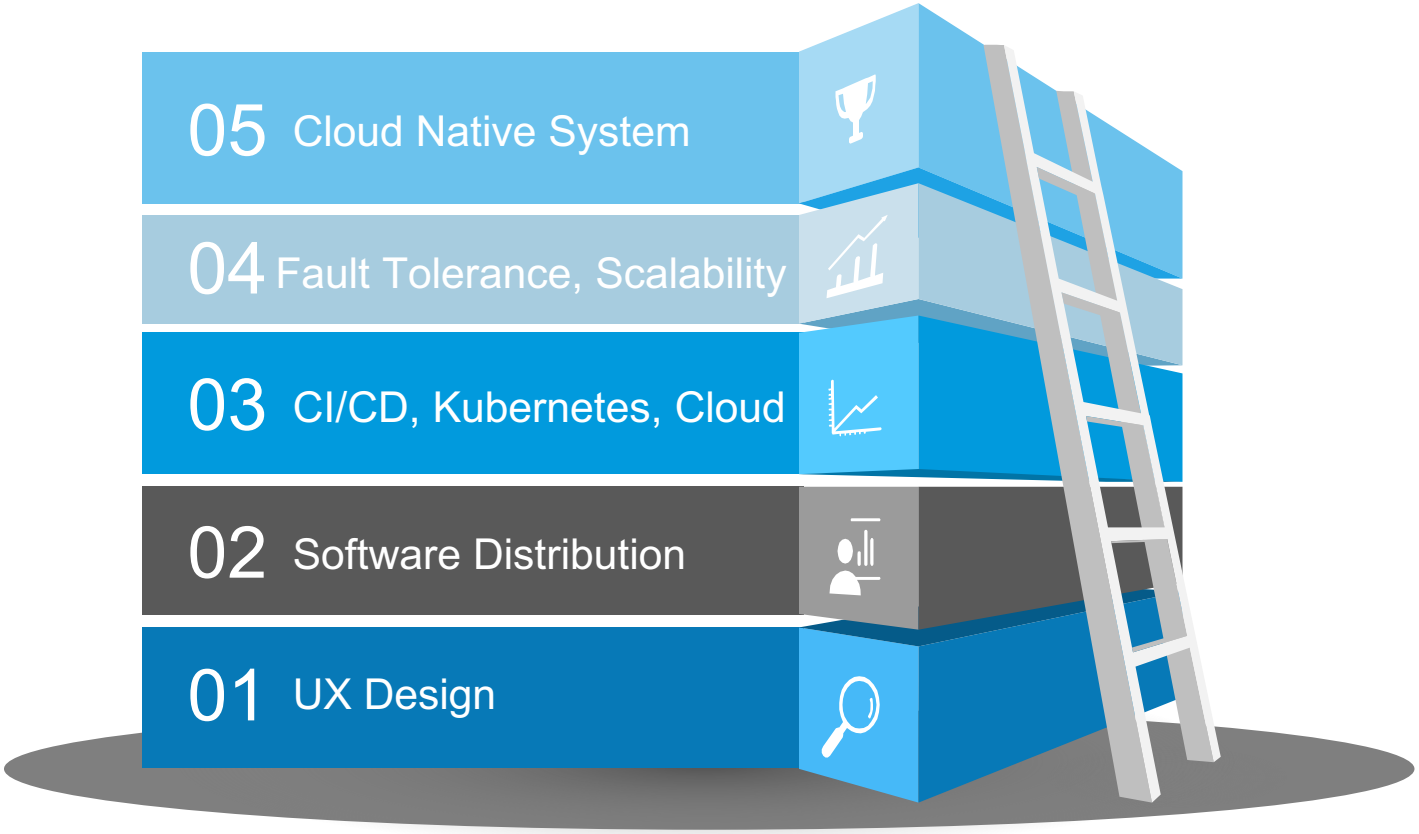
Structure of the Class

- We will have 5 project-based assignments
  - 90% of your grade
  - Projects will done by a team of 3 but graded individually.
  - 5 Bonus points/project for peer review (individual)
- 10% for mid-term and final presentations
- 10% (bonus) will be attendance, classroom interactions and peer project interactions (auditable github communications, whatsapp msgs do not count).

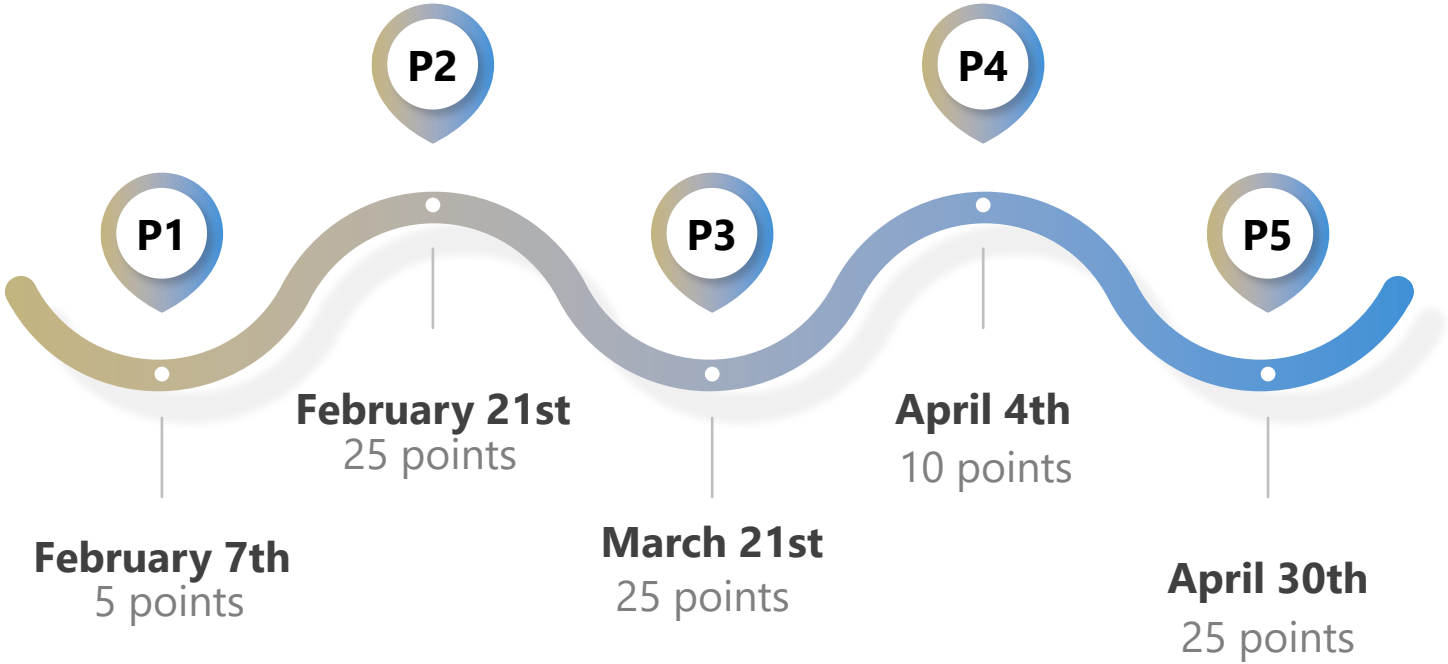
Characteristics of a Good Technology Base

- ✓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.

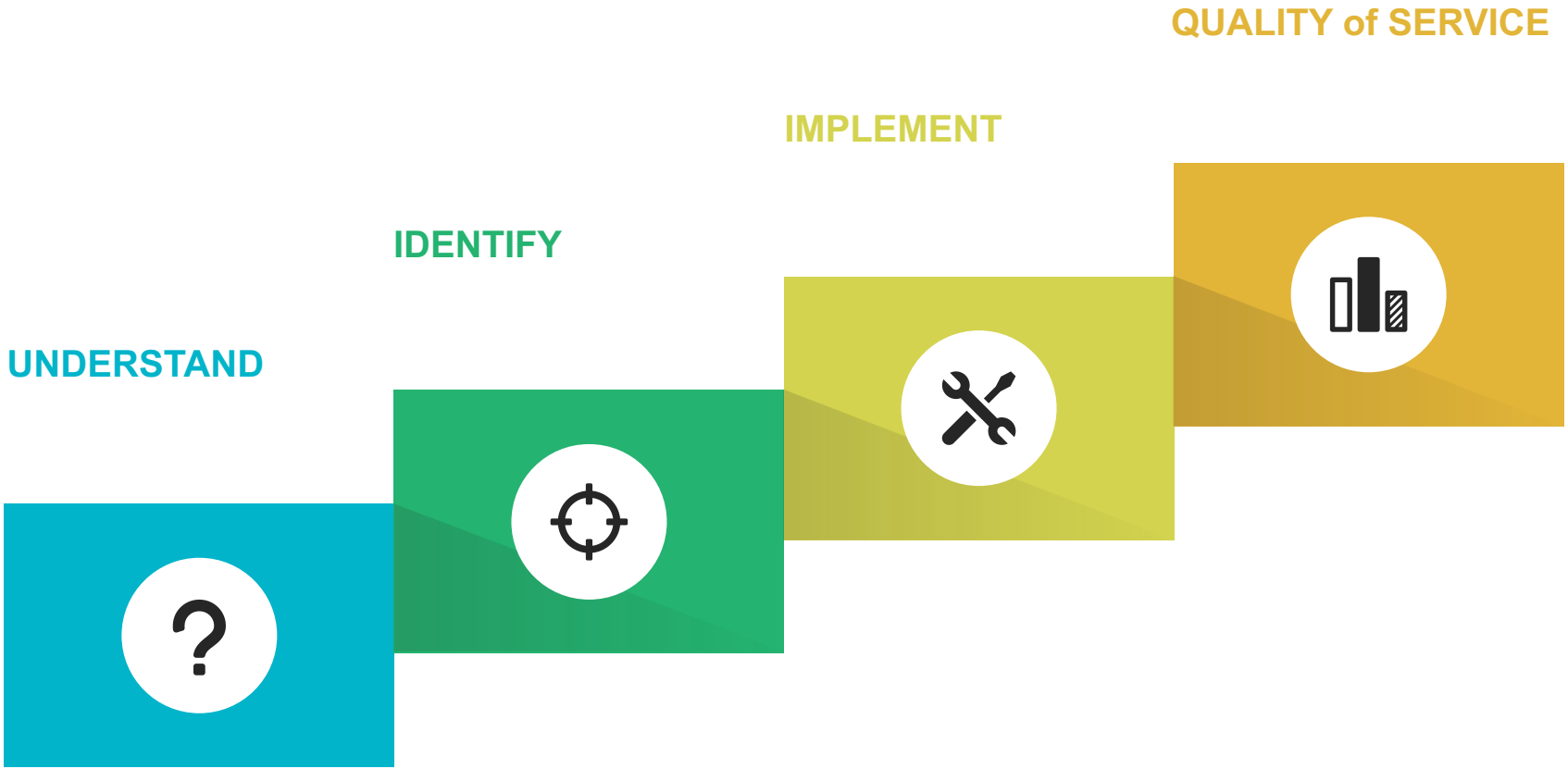
Project Milestones



Project Due Dates



Will start with User Experience



Traditional Product Development Process

Concept

Idea Generation

- Requests
- Customer Pain
- Market Studies
- Legislation
- Competitors

Key deliverables

Products concept docs

Research

Assess Market

- Segment & size
- Growth potential
- Customer needs
- Legal issues
- Competition

Key deliverables

Market Research Report
Market Req. Document
Product Definition Statement

Analysis

Business Analysis

- Cost/Benefit
- Resources Required
- Capital Expenses
- Profitability/Margin
- Anticipated Sales

Key deliverables

Business Case
Profitability Analysis
Product Req. Document

Develop

Product Development

- Technical Specs
- Prototyping
- Trial Production
- Testing & QA
- Test Market Selling

Key deliverables

Product Dev. Schedule
Product Testing Report
Product Req. Document

Launch

Go to Market

- Marketing Plan
- Sales Training
- Distribution Plan
- Collateral Design
- Set Launch Date

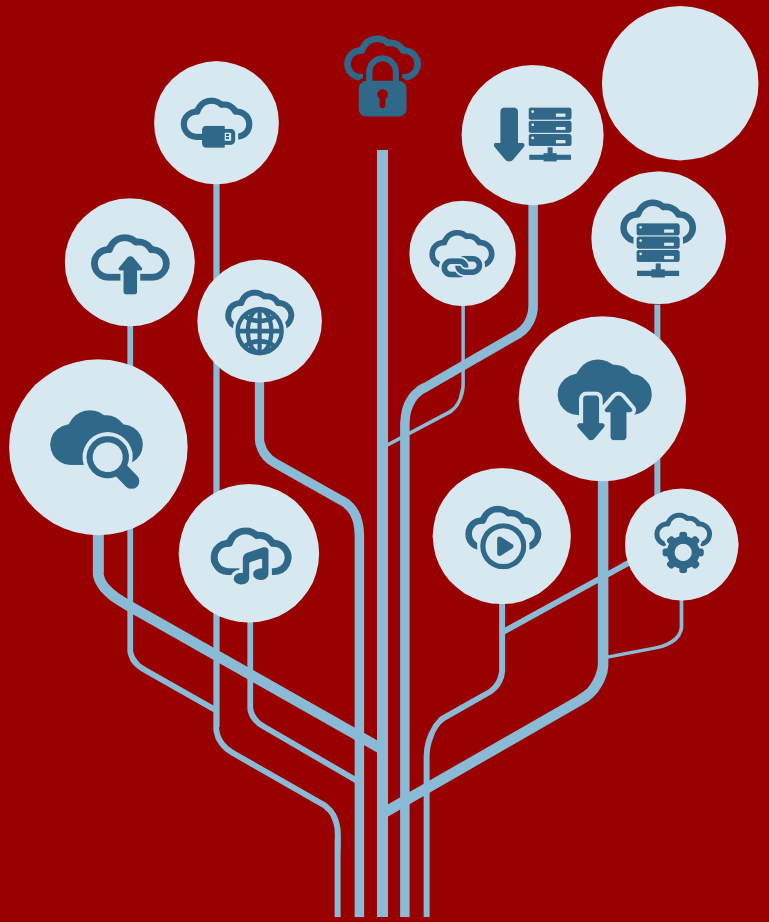
Key deliverables

Product Launch Plan
Product Launch Budget
Product ROI Forecast
Target Launch Date Set

Microservice Architecture Principles

Each service is broken
by a functional capability

Services should be able
to evolve independently,
scale independently.



Polygot – use at least 3 programming languages

CODING

All components (including UI) need to use a build framework: Make, Maven, Bower…

Cyber Security at all layers:

Go beyond Authentication and Authorization

Securing all communications

Project Overview

- Project 1 (5 points); Due February 7th
  - Deliverables: UX Designs, Sketching, Wireframes, Task Flows
- Project 2 (25 points); Due February 21st
  - Develop at least 5 services running as separate process
    - API Service
    - Image Metadata Extraction
    - Browse
    - Indexing and Search Service
    - User Management
  - Only microservice can connect to DBMS
  - Deliverables: Github README which points to all buildable software.

Project Contd..

- Midterm presentations - March 16th, 18th
  - Group presentations with individuals taking turns
- Project 3: 25 points - Due March 21st
  - Containerize each service
  - Establish CI/CD
  - Use Kubernetes to Manage Deployments on Jetstream Cloud
  - Deliverables: Reproducible CI/CD pipeline to move from code to a running service.

Projects Contd.

- Project 4: (10 points) Due April 4th
  - Using JMeter evaluate the performance of your system and plot throughput under incremental loads
  - At what point does your system fail
  - Inject failures and describe the operational limits of your system
- Final Presentations – April 27th & 29th
- Project 5: (25 points) Due April 30th
  - Add elastic resource management using Cloud Native technologies such as Service Mesh's.
  - Re-Test with 1, 3 and 5 replicas
  - Re-Test with elastic resource management (as system changes under load, resources scale up and down)
  - Manually kill a VM and see the recovery of the system

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.