

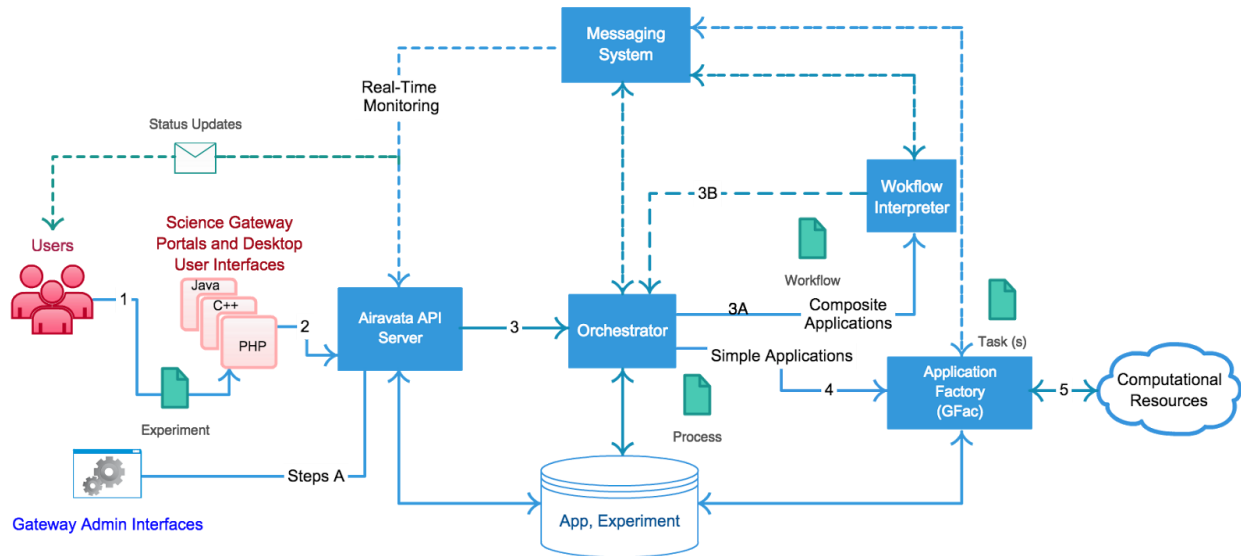
Course Goals: Advanced Science Gateway Architectures

The goal of this course is for the students, building on their experiences in the introductory “Science Gateway Architectures” course, to explore specific open problems in science gateway architectures. Working with the course instructors, students will learn how to formulate concrete problems from general research issues, design and implement solutions, and critically evaluate their solutions and the solutions of their peers.

Over the last two decades, science gateways have matured from practical Web portals designed to submit jobs to supercomputers into full fledged distributed systems cyberinfrastructure that aims to manage all aspects of a computational and data science research, from conception to exploration to publication to reproducibility. It is a research challenge to investigate distributed systems architectures for science gateways that simultaneously enable scientific research in computational and data science, adopt the best modern “DevOps” practices for managing the developing and operations cycle, and have firm foundations in distributed systems concepts. The overarching problem is to do distributed computing research to define a reference architecture that can accommodate explorations of solutions to these challenges.

In this course, we will examine the requirements for a distributed computing framework that can simultaneously address the requirements of both computational and data-centric cyberinfrastructure. Through exploratory projects, students will have the opportunity to map distributed systems concepts to science gateway requirements. Course topics will be based on the Apache Airavata framework, which will provide a laboratory for student projects; students will explore alternative approaches for implementing innovative capabilities within the Airavata. Students will gain visibility of their work through public pull requests to the Apache Airavata code repository and discussions of their work on the public Apache Airavata mailing lists. This course will require the background of the Spring 2016 (INFO-I509) and Fall 2016 (CSCI-B 649) “Science Gateway Architecture” courses. Students must have approval from the instructors to enroll.

The following diagram provides a high-level representation of Apache Airavata. The course will explore various architectural and implementation approaches to accomplish these high level goals.



Project Themes

Through exploratory themes, students will validate their understanding or learn foundation distributed systems concepts like Reliability, Scalability, Efficiency and Availability. The class will enable the students to get deeper insights into modern web scale systems and assumes understanding of DevOps strategies, microservice architecture principles and underpinning concepts like applications of CAP theorem.

Each student will contribute to one or more project themes from the list below as part of a student team. The student teams will be responsible for researching, designing, implementing, and evaluating their solutions. Each theme will be assigned a project leader, who will be an experienced Apache Airavata developer, and 2-3 additional student team members.

- Theme 1: APIs, Data Model Representation and Storage
 - API Comparisons: REST, Swagger, Thrift, ProtoBuff etc.
 - Optimal serialization, deserialization, storage and search strategies
 - Internal (CPI) versus externally exposed (API) data models
- Theme 2: Distributed Workload Distribution
 - Load Balancing Stateful vs Stateless Services
 - Message based vs hybrid message-RPC vs pure RPC.
 - Work-Queue vs Master-Worker load balancing
- Theme 3: DevOps Strategies
 - Fault tolerance testing and validation
 - Load balancing testing and validation
 - Continuous upgrades without downtime
 - Capacity testing
 - Comparison of hosting alternatives: OpenStack (Jetstream), Amazon AWS, Google, Microsoft Azure
- Theme 4: Cyberinfrastructure Integration

- Interacting with remote computing resources and data
 - Metascheduling
 - Gateway metadata and provenance
 - Alternative resource utilization
- Theme 5: Cybersecurity
 - PHI, HIPAA, etc alignment for gateways
 - DevSecOps practices
 - Auditing, monitoring, and event detection
 - Client assurance
- Theme 6: Data-Centric Computing and Gateways
 - Airavata and data stream processing
 - Internet of Things and Airavata
 - Searching and data mining structured data sets
 - Event-driven computations
- Theme 7: Scientific User Environments
 - Gateways and interactive computing
 - Application monitoring
 - Visualization
 - Modern user interfaces and environments for science
 - Notebook integration
- Theme 8: Application Toolboxes for Airavata Applications
 - Defining toolboxes: application and resource collections.
 - Toolbox templates
 - Packaging, integrating, and sharing tool boxes

Course Format and Weekly Lectures

The course will consist of 150 minutes/week of contact time with the instructors as lectures and team discussions. Specific classes will be structured around lectures on topics relevant to the projects. These will be a mixture of instructor-led and student-led presentations.

Grading Overview

This will be a project-based course with students working on team projects. Each project will be subdivided into four milestones with specific, demonstrated deliverables that the student teams will present to the class. Students will use these deliverables to evaluate their solutions and make specific conclusions on their solutions to the project challenges. All work will be in public GitHub or Apache-hosted repositories.

Each project milestone will be worth 20 points. To earn full credit, the student teams must submit the project milestone on time and demonstrate to the instructors that all required deliverables and evaluations were met.

Each student will also be assigned 5 thirty-minute presentations throughout the semester, with each presentation worth points (5 points total). Students will submit a final report (10 points), in

which the student will summarize his/her individual work and contributions. This will include pointers to the student's code contributions, developer list discussions, presentation materials, etc. The remaining 5 points are awarded based on attendance, classroom participation, and interactions with other students and with the instructors.