



INDIANA UNIVERSITY
**SCHOOL OF INFORMATICS,
COMPUTING, AND ENGINEERING**

Science Gateway Architectures

January 11th, 2019

Suresh Marru, Marlon Pierce

smarru@iu.edu, marpierc@iu.edu

Todays Outline

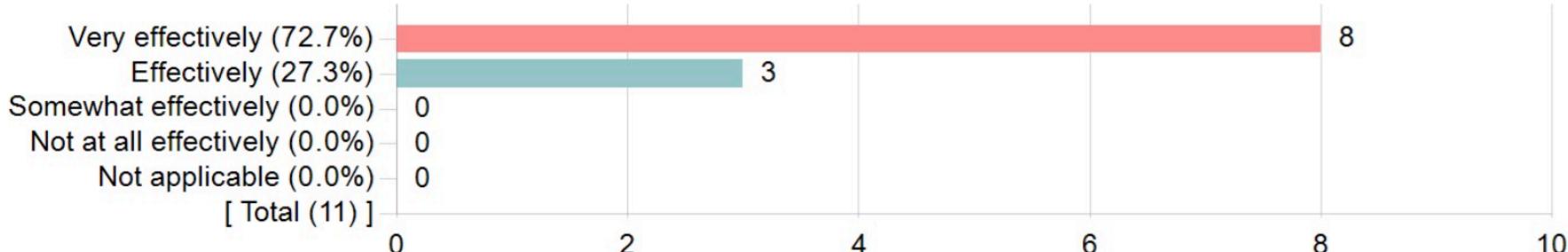
- What To Expect
- Project Teams
- Microservices
- Project Themes
- Open Discussion



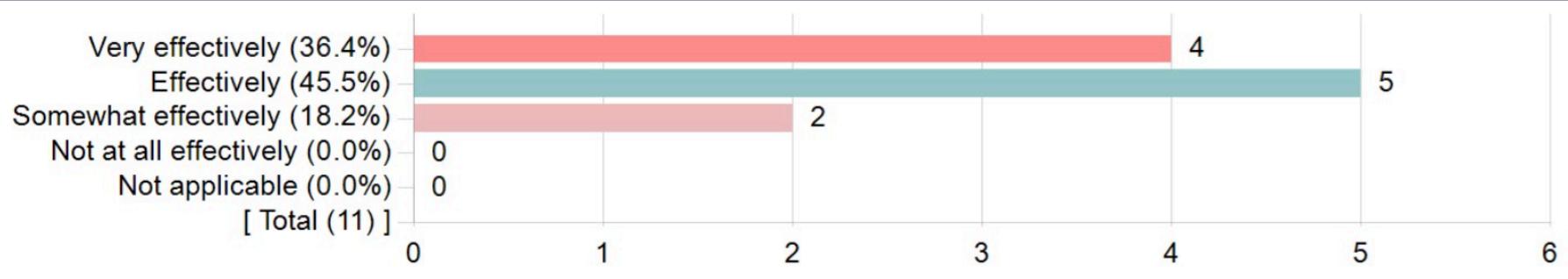
INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

How effectively did out-of-class work (assignments, readings, practice, etc.) help you learn?



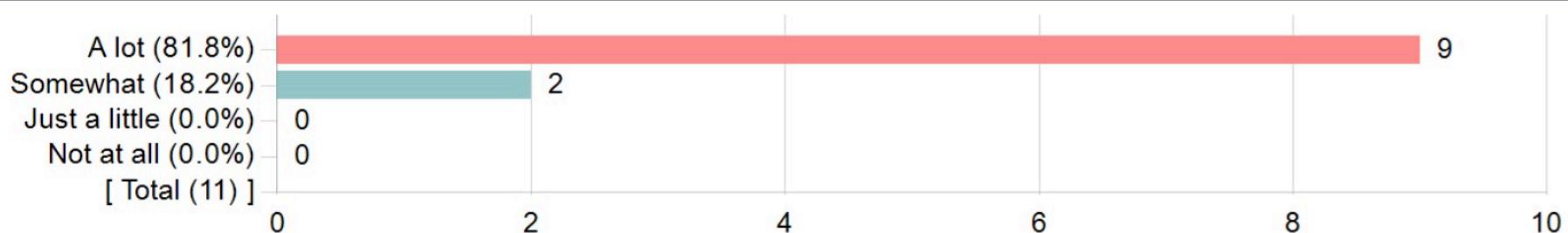
How effectively was class time used to help you learn?



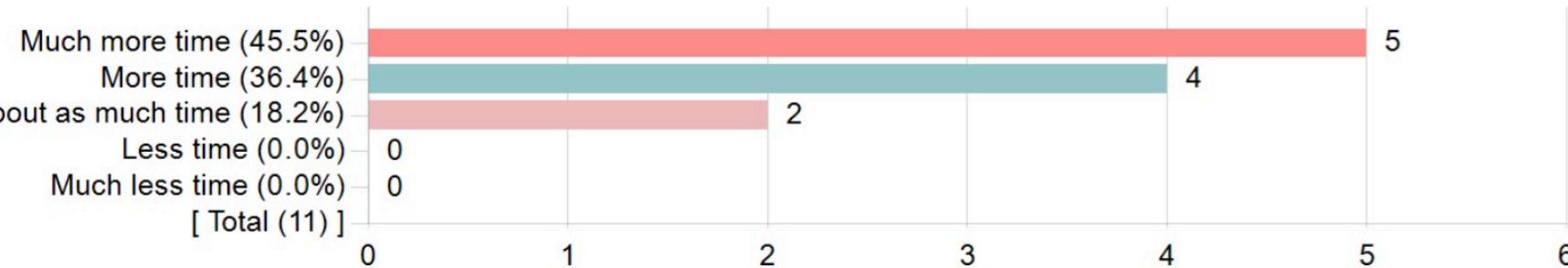
How much did the instructor motivate you to do your best work?



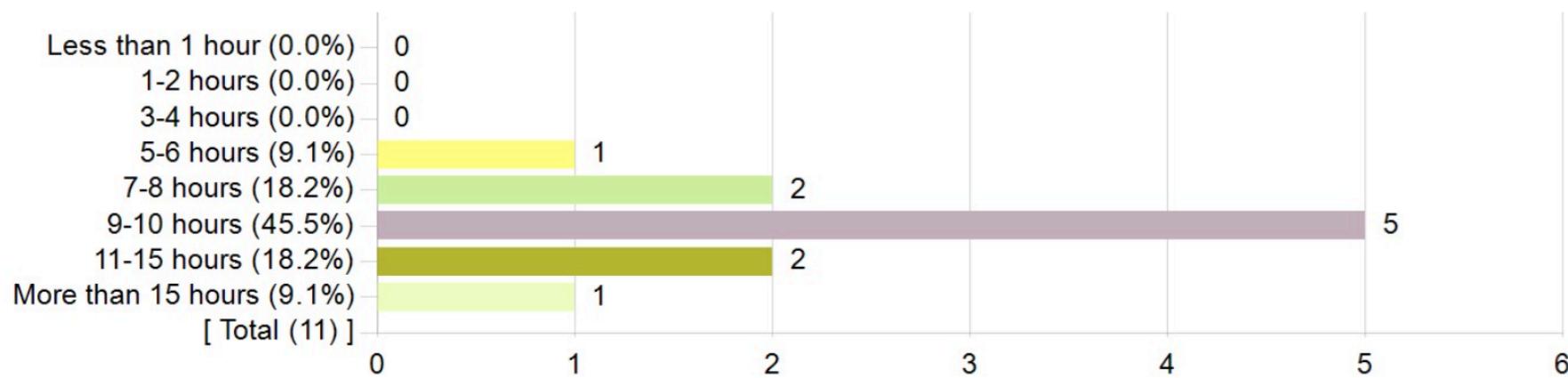
How much did the instructor emphasize student learning and development?



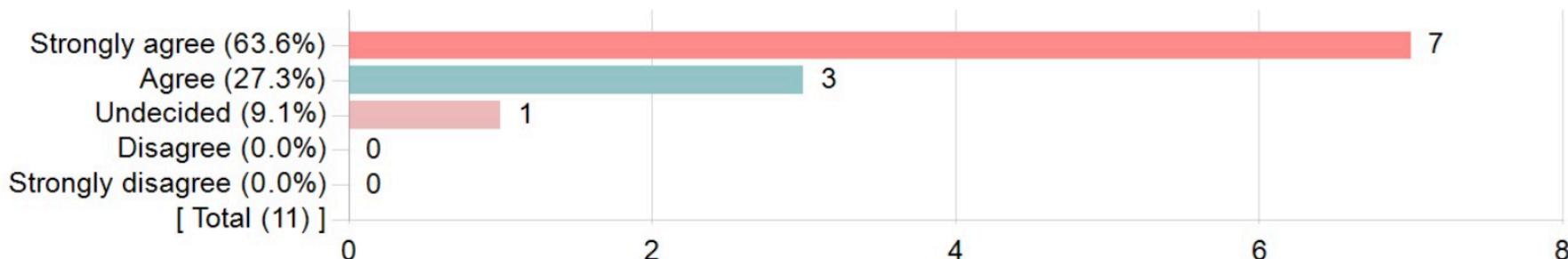
Compared to other courses you've taken, how much time did this course require?



In a typical week, about how much time did you devote to this course? (Do not count scheduled class time, labs, etc.)



I developed skill in critical thinking in this course.



What did you like most about this course and instructor?

Comments

Learning so many tools.

Best thing about projects is that I got to play around with instances on jetstream and there is no spoon feeding on how to go about setting it up. I learnt a lot in the process.

Explains in comprehensive manner

Topic Explanations were clear

Was able to learn new tools and technologies. The course was flexible with the project ideas being chosen.

Ample time was given to work on assignments

What could the instructor do to improve the course or his/her teaching effectiveness?

Comments

Great

This is one of the best class I've taken. Instead of having a tutorial series on various technologies. The course material gave high level architectural overview of the various technologies like kafka and rabbitmq. Personally I've got into the habit of reading papers in the context of distributed systems.

Please continue giving architecture and design details instead of tutorials. These are the concepts which aren't available in blog posts which can only be found by digging deep in papers

Involve more lectures about the technologies used in the projects

Teaching is great but the class can be a little more interactive

Should include more industry specific examples associated with the concepts and tools being discussed in class.

He is good, No Comments.

In Summary...

- We expect you to do a lot of work for the class
- But we expect you will find the class challenging, rewarding, and enjoyable
- Make your semester plans accordingly
 - We'll offer the class again in Spring 2020



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Science Gateways Research Center

Accelerate research, discovery and scientific collaboration through the creation, integration, and operation of systems that combine **distributed software**, data, and infrastructure resources into **user environments** that serve and enable **scientific communities**. Advance the broader cyberinfrastructure community through education, training and sharing knowledge.



INDIANA UNIVERSITY

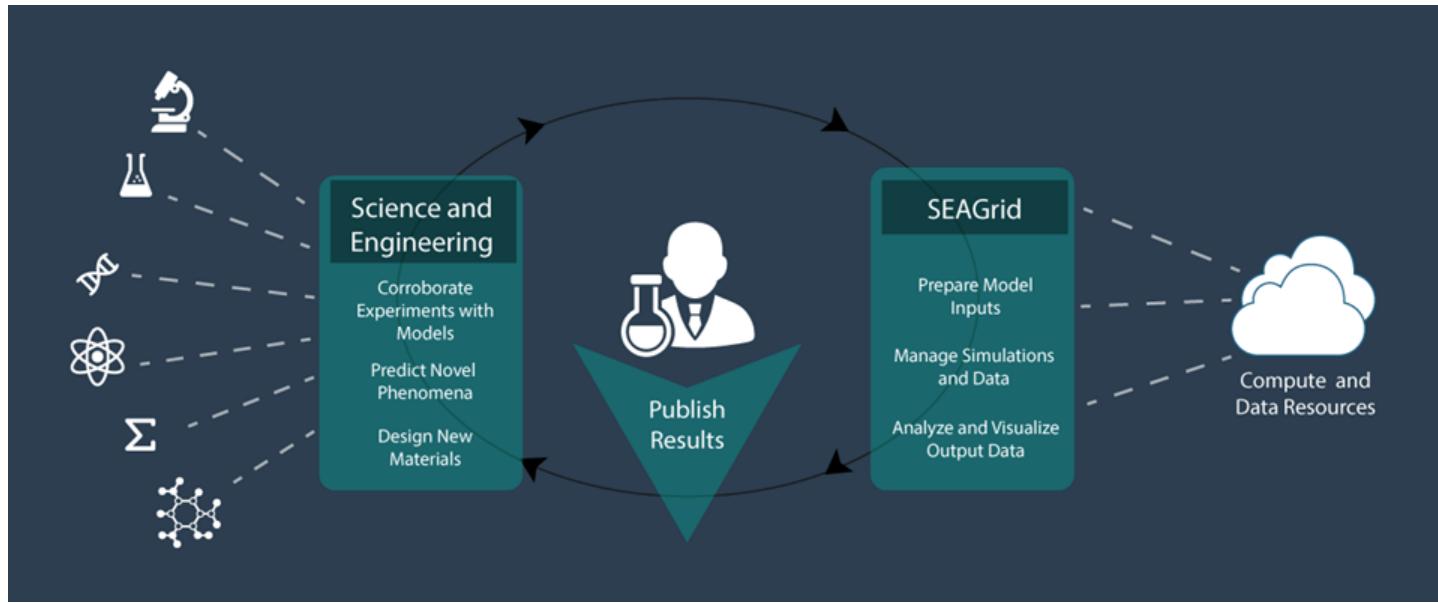
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Example Science Gateway



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING



SEAGRID.org is an Apache Airavata-powered gateway

Hydrated Calcium Carbonate in Action



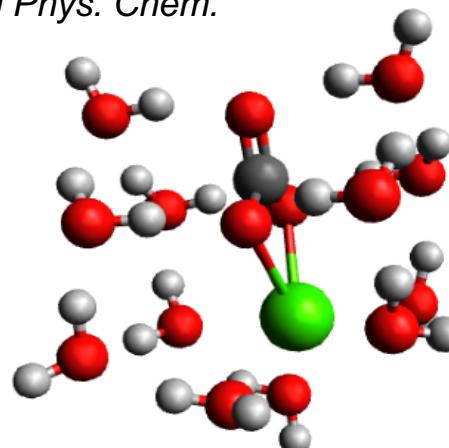
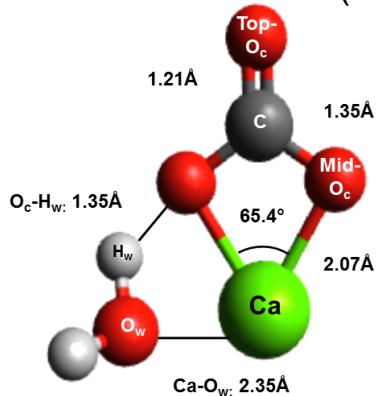
INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

What is the chemistry of hydrated calcium carbonate?

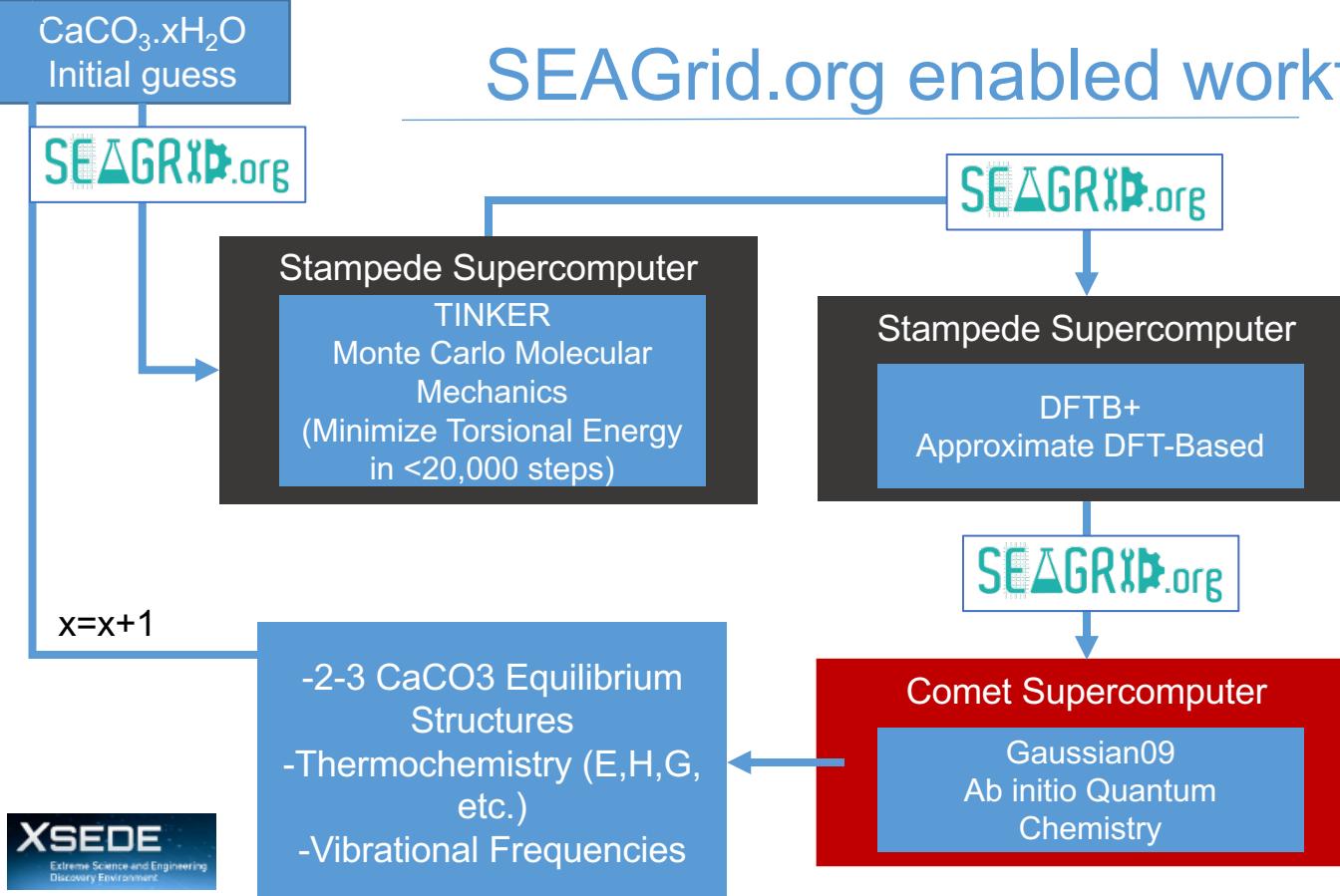
- Bio-mineralization of skeletons and shells
- Geological CO₂ sequestration
- Cleanup of contaminated environments

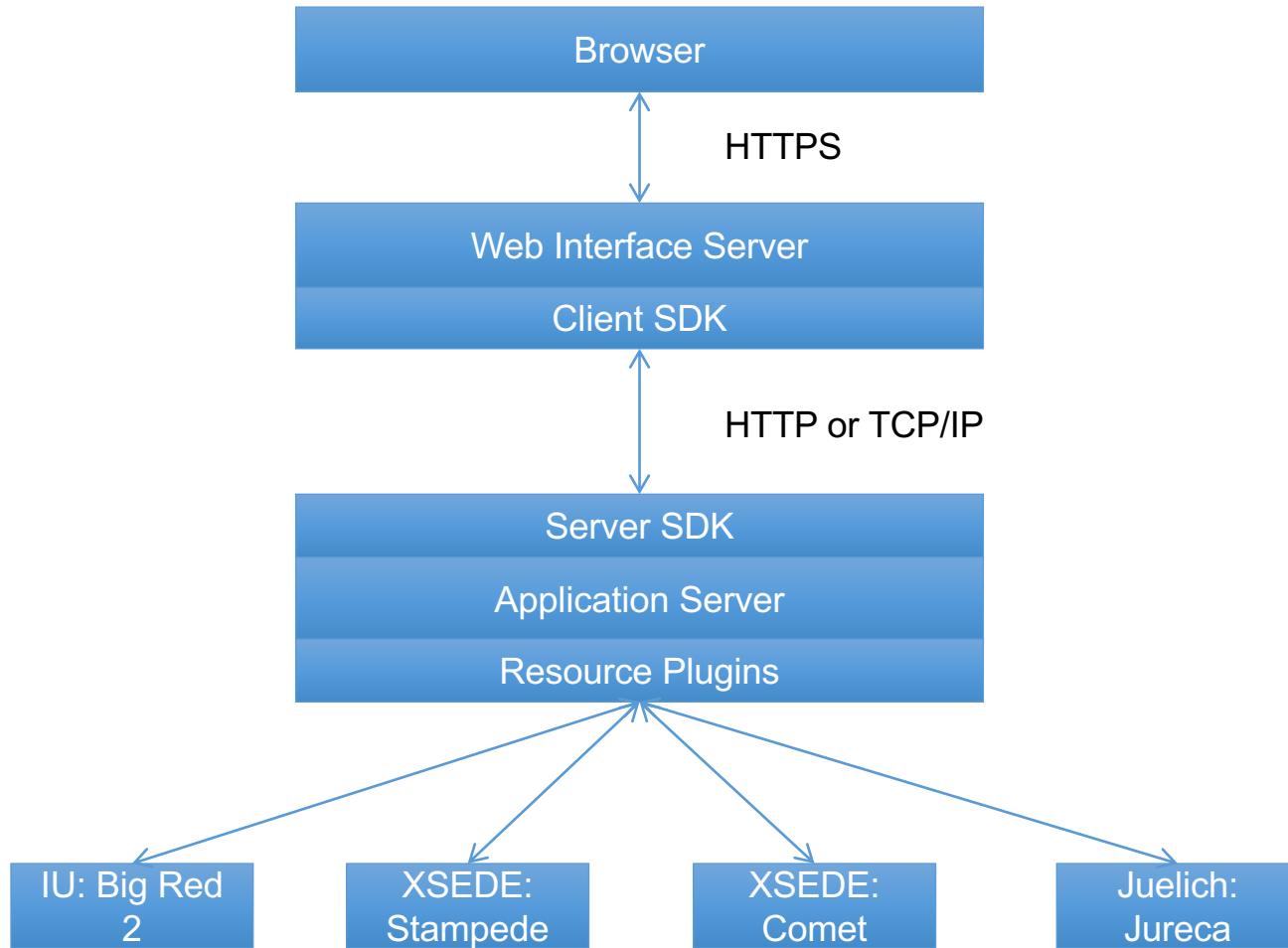
Lopez-Berganza, et al. *J Phys. Chem. A*(2015)



$\text{CaCO}_3 \cdot x\text{H}_2\text{O}$
Initial guess

SEAGrid.org enabled workflow





Challenges for Science 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



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Goal 1: Apply basic distributed computing concepts to Science Gateways.



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Goal 2: Apply new architectures, methodologies, and technologies to Science Gateways: Microservices, DevOps

You Don't Choose Chaos Monkey...
Chaos Monkey Chooses You

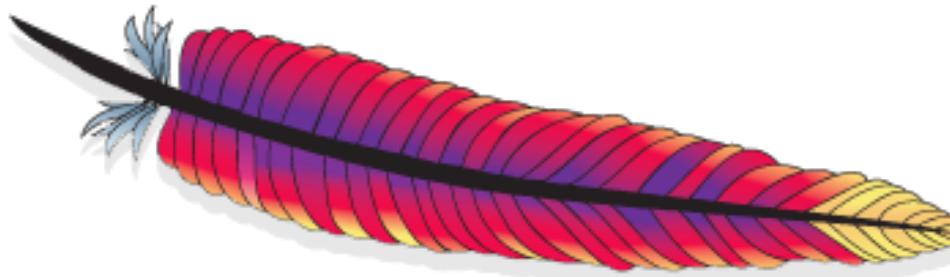


@RealGeneKim, genek@realgenekim.me



INDIANA UNIVERSITY
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Goal 3: Teach open source software practices



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Why Do We Teach This Class?

We are looking for students who like what we do and want to contribute to Apache Airavata.



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

What Is Apache Airavata?

- Open source middleware to support Science Gateways
 - Compose, manage, execute, and monitor distributed, computational workflows
 - Wrap legacy command line scientific applications with Web services.
 - Run jobs on computational resources ranging from local resources to computational grids and clouds
 - Record, preserve, search, and share metadata about computational experiments
- Hosted version of Apache Airavata provides multi-tenanted Platform as a Service.
 - SciGaP



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

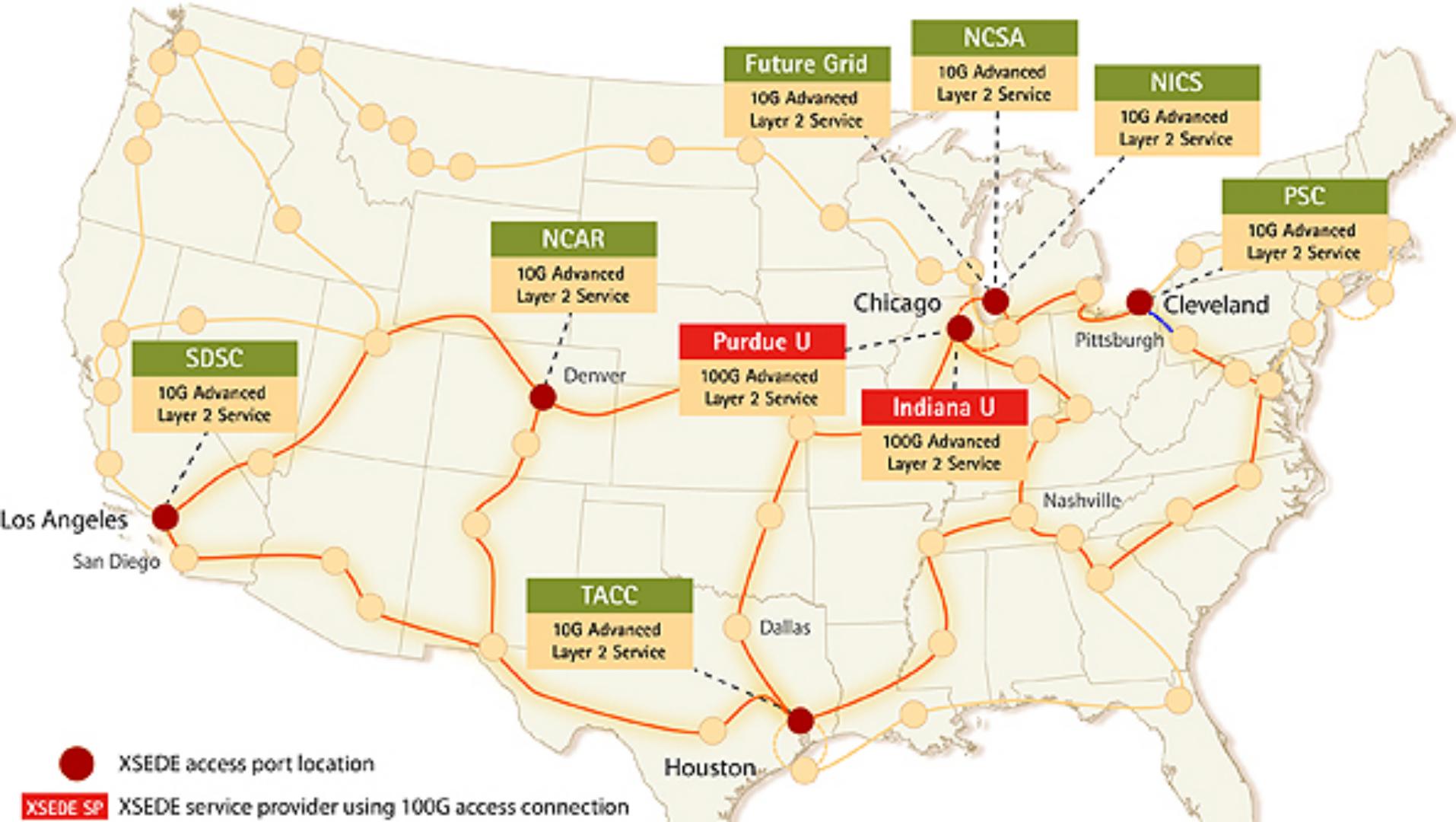
What You Need to Do Before Next Class

- Create a GitHub account if you don't have one
- Send us your GitHub user name so that we can add you to the Airavata Courses GitHub organization
- Go to <https://portal.xsede.org/> and create an account
 - Send us your user name
- If possible, please make these user names consistent with your IU network ID.
- Submit this through Canvas (0-point assignment)



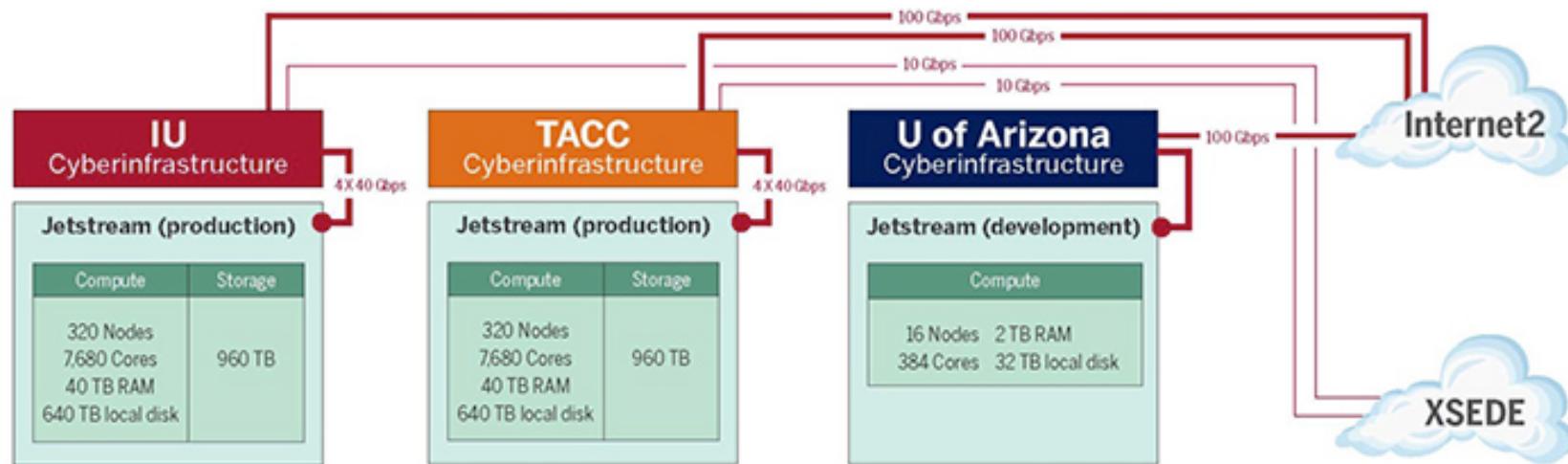
INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING





Science Engineering Cloud based on OpenStack



Monolithic Applications: Traditional Software Releases

- Software releases occur in discrete increments
- Software runs on clients' systems
- Releases may be frequent but they are still distinct
 - Firefox
 - OS system upgrades
- Traditional release cycles
 - Extensive testing
 - Alpha, beta, release candidates, and full releases
- Extensive recompiling and testing required after code changes
- Code changes require the entire release cycle to be repeated



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

MicroServices: Software as a Service

- Does your software run as a service?
- Do you run this service yourself? Or does another part of your organization run the service that your team develops?
- Traditional release cycles don't work well
 - May make releases many times per day
 - Test-release-deploy takes too long
- You can be a little more tolerant of bugs discovered after release if you can fix quickly or roll back quickly.
- Get new features and improvements into production quickly.



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Science Gateways and Microservices



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

What Is a Microservice?

- Develop a single application as a suite of small services
- Each service runs in its own process
- Services communicate with lightweight mechanisms
 - “Often an HTTP resource API”
 - But that has some problems
 - Messaging and hybrid approaches
- These services are built around business capabilities
- Independently deployable by fully automated deployment machinery.
- Minimum of centralized management of these services,
 - May be written in different programming languages
 - May use different data storage technologies.



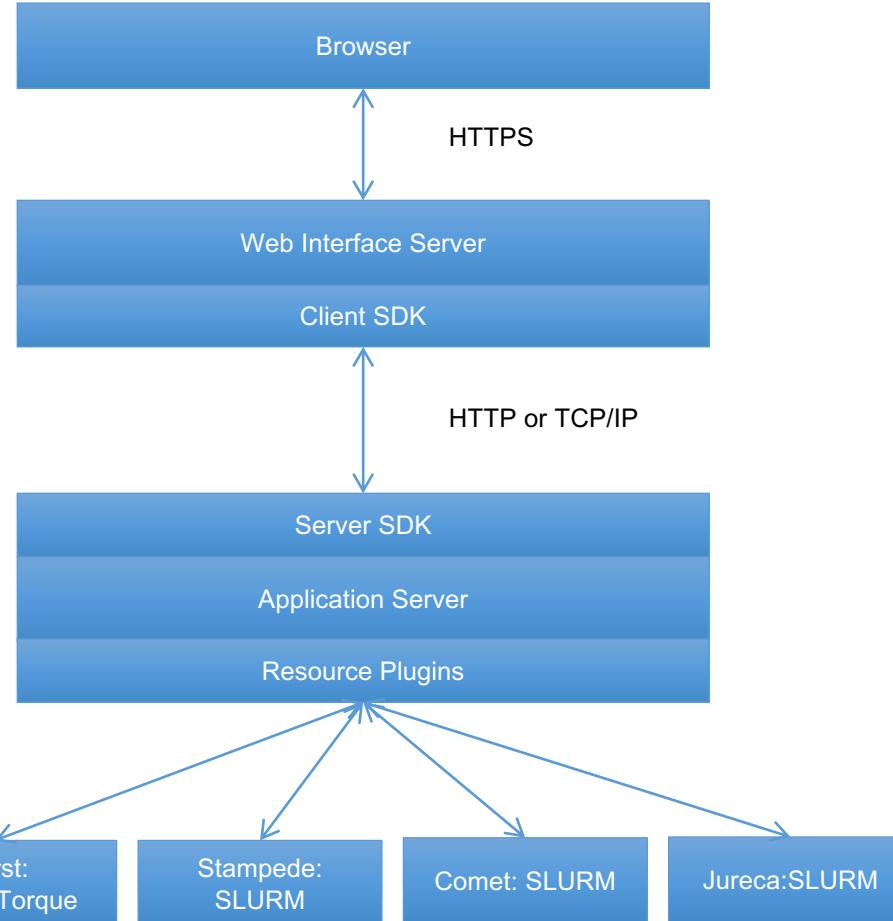
INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

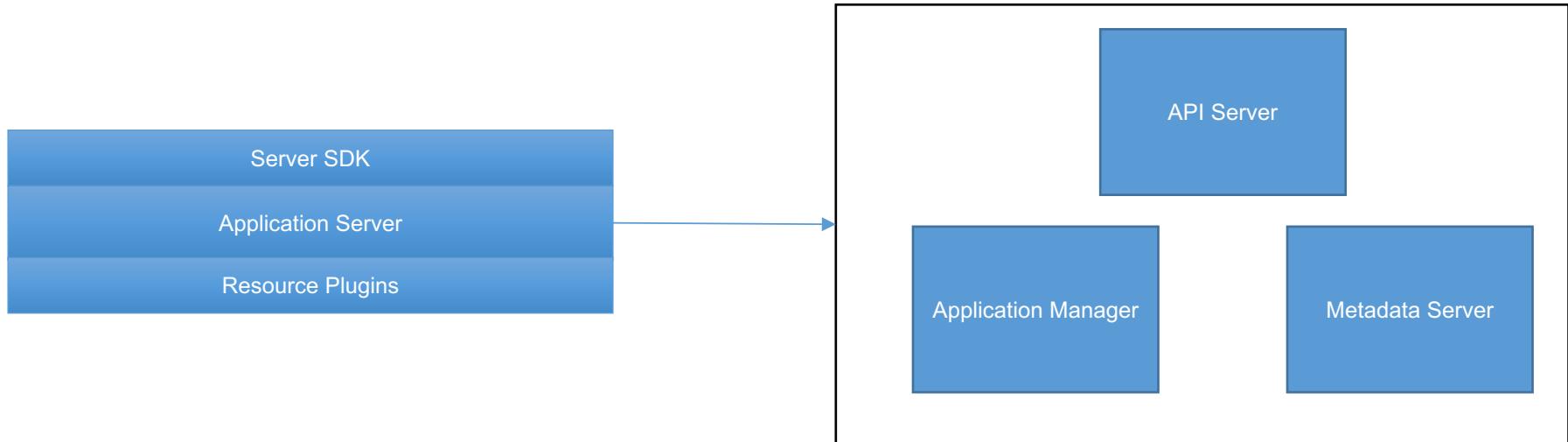
<http://martinfowler.com/articles/microservices.html>

Recall the Gateway Octopus Diagram

We will focus
on this piece



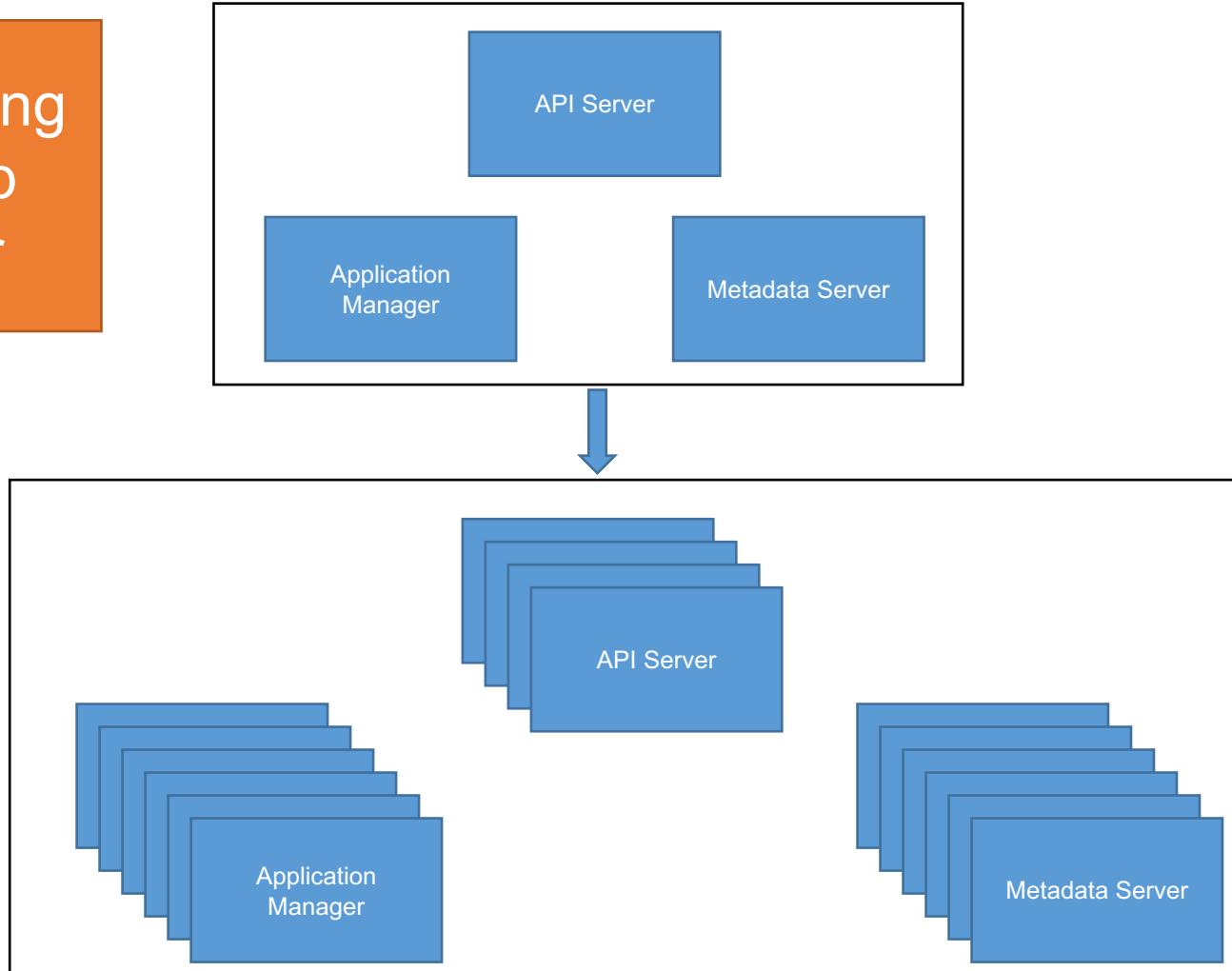
Basic Components of the Gateway App Server



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Decoupling the App Server



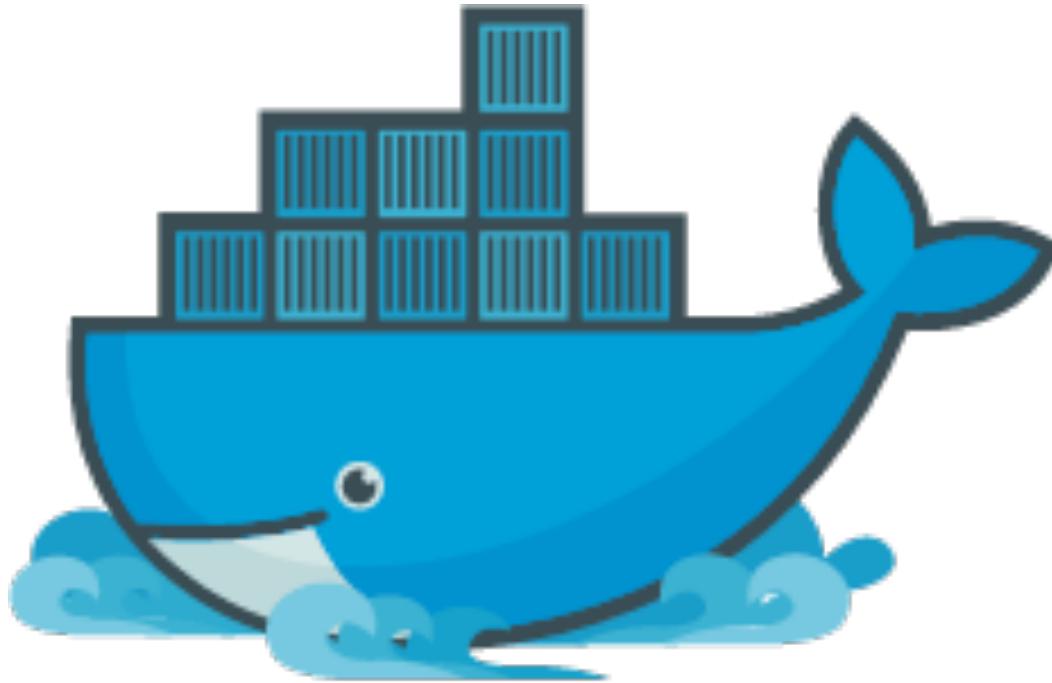
How Do We Package and Where Do We Run All Those MicroServices?

On the Cloud? In the Matrix?



INDIANA UNIVERSITY
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Virtualization, Containers, Docker

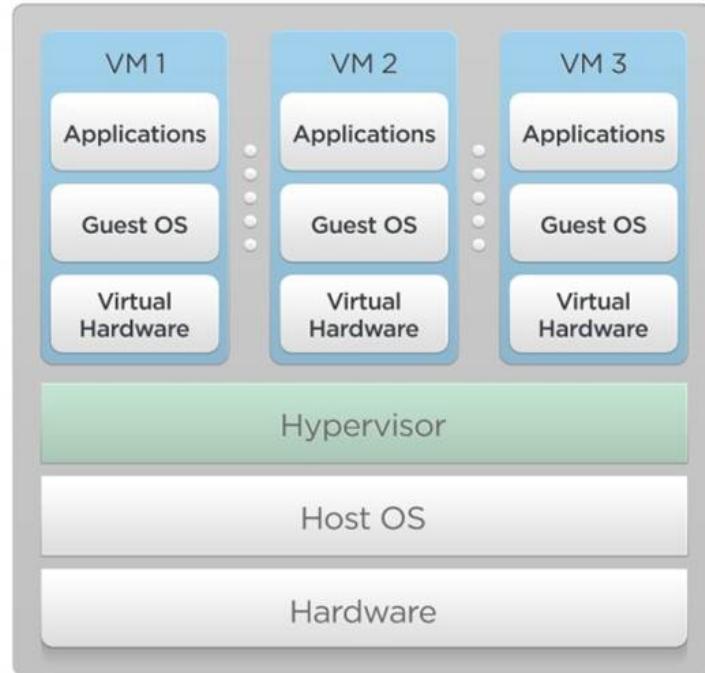


INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Hypervisor Virtualization

- Hypervisors provide software emulated hardware and support multiple OS tenants.
- Type 1 Hypervisors run directly on the hardware
 - Kernel-based Virtual Machine (KVM)
 - Xen: Amazon uses (or used) this
- Type 2 Hypervisors run as another program in the HostOS
 - Virtual Box
 - VMWare



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Virtualization Benefits and Drawbacks

Benefits

- Makes your development, testing, and deployment environments the same
- Supports Continuous Integration and Deployment

Drawbacks

- Adds overhead
 - Relatively slow to create and destroy VMs
 - Virtualized hardware and networking are slower than real systems
 - Uses memory, disk, and CPU resources

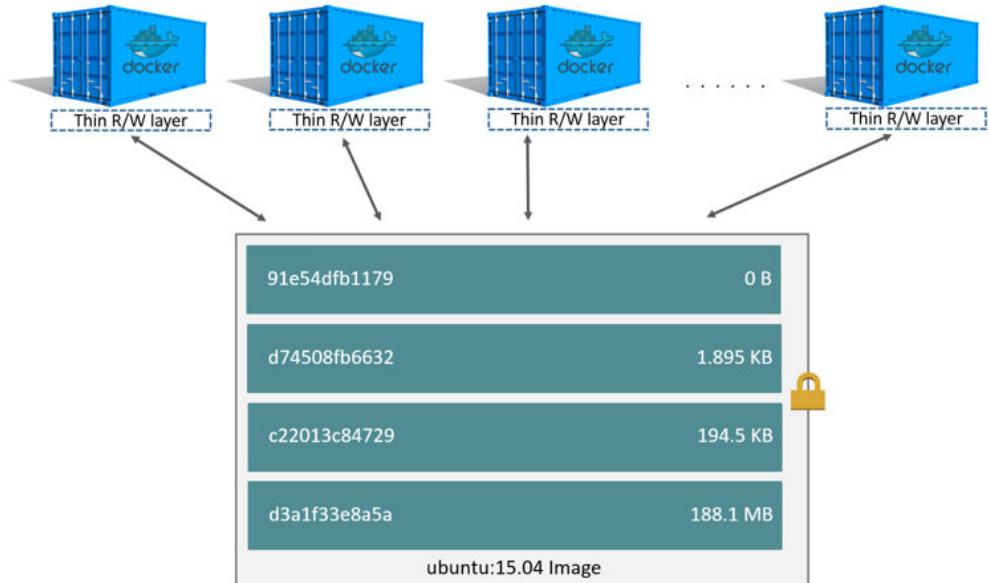


INDIANA
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

You can run 1 VM per microservice, but there is a better way.

Docker and Other Linux Containers

- Docker replaces the need for DIY containers
- Docker uses LXC containerization + AuFS
 - AuFS lets Docker containers share common files
 - AuFS also allows you to have efficient version control of container images



Because each container has its own thin writable container layer, and all changes are stored in this container layer, multiple containers can share access to the same underlying image and yet have their own data state.



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

How Do Microservices Communicate?

Push, Pull e.t.c



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

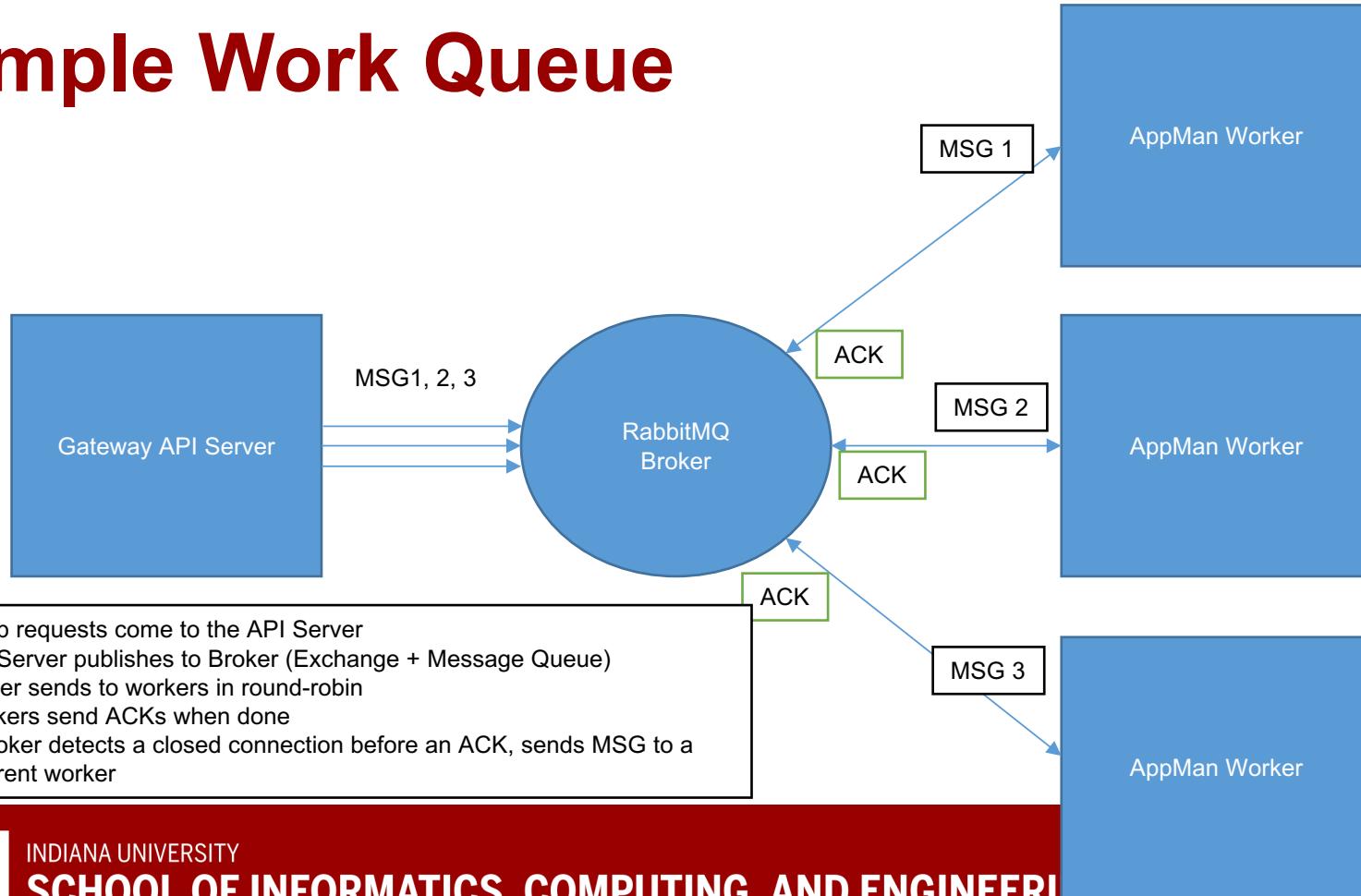
Messaging Systems: RabbitMQ, Apache Kafka



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Simple Work Queue



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Expose their APIs and Data Models to Other Components?

And can we make this programming language
independent?



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

API and Metadata Model Design

Apache Thrift™

RESTful API

GET PUT POST DELETE



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

How Can I Discover, Monitor, and Manage Services?

Can we learn some lessons from distributed systems research?



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

How Can I Discover, Monitor, and Manage Services?

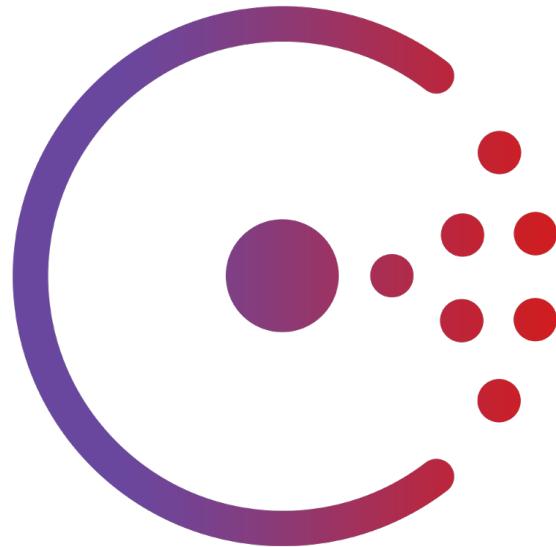
Can we learn some lessons from distributed systems research?



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Distributed State Management: Zookeeper, Consul



INDIANA UNIVERSITY
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

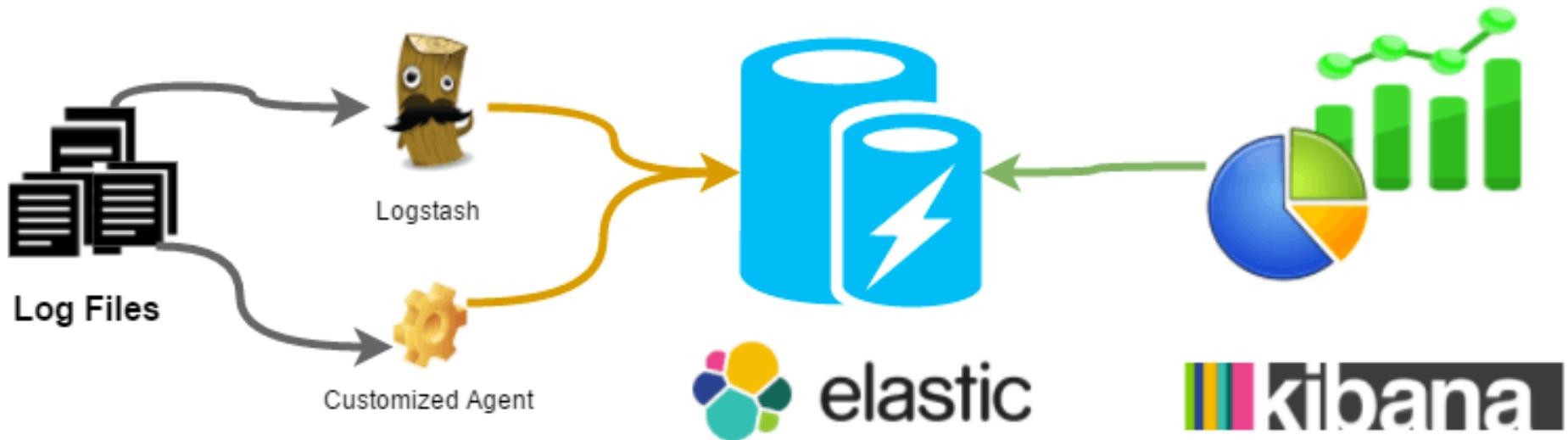
How Do I Manage Logs from Microservices

And detect if there are problems



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING



How Can I Secure my API Server and Microservices?

How do I manage user identities, authentication and authorization?



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Security: OAuth2 and OpenIDConnect



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Interact with Computing Clouds and Supercomputers?

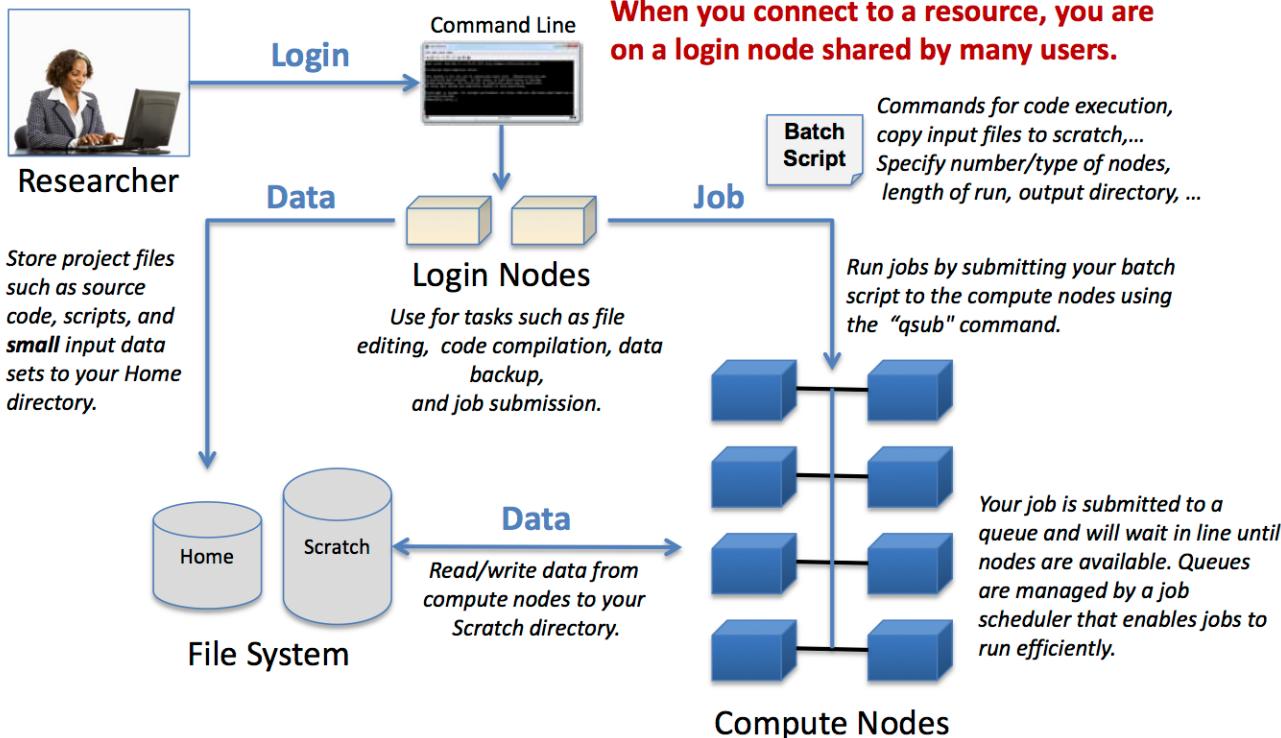
How can we work reliably with unreliable resources?



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Running Jobs Overview



Resource Management and Scheduling: Mesos, Aurora, and Torque



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

How Can We Automate All of This?

How can we make our infrastructure reproducible?



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Course Logistics



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Course Communication

- For syllabus information and presentation material, see the course website
 - <http://courses.airavata.org>
- For project assignments and other communications, use Canvas
- For projects, use GitHub



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Grading

- Four team projects per semester
 - 18 points each
 - 2 points for peer reviews
- Mid-term and finals week team presentations/demos
 - 10 points each



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Project Teams

- The class will be divided into 3-person teams
 - More (4 person) or less (2 person) as needed
- Get some diversity of expertise
 - Programming languages, UI expertise, service/middleware development, DB expertise, systems admin/linux expertise
- This is also your chance to learn new stuff
 - If you spend the entire semester only doing UI work (for instance) because that's what you know, then you are not getting anything from the class
- **We will form teams next week after we have the final roster**
 - You can pick your teammates.
 - We'll make this a 0-point Canvas assignment



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Course Projects

- Project Milestone 1: Microservices with DevOps
- Project Milestone 2: Security, Testing, Distributed Coordination
- Project Milestone 3: Reliability & Scaling
- Project Milestone 4: Project team choice



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Project Repository

- All team work will be in GitHub
- Each team will have a separate project under <https://github.com/airavata-courses/>
- You will use GitHub to manage all your code and releases (project assignments)
- You will use GitHub's Wiki to document your project and your releases
- You will use GitHub issues to track the work you do
 - Attach all commits to issues
- You will use GitHub issues also for peer reviews
- **Create a GitHub account if you don't have one**
 - Send us your user name via a 0-point Canvas assignment



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Class Time

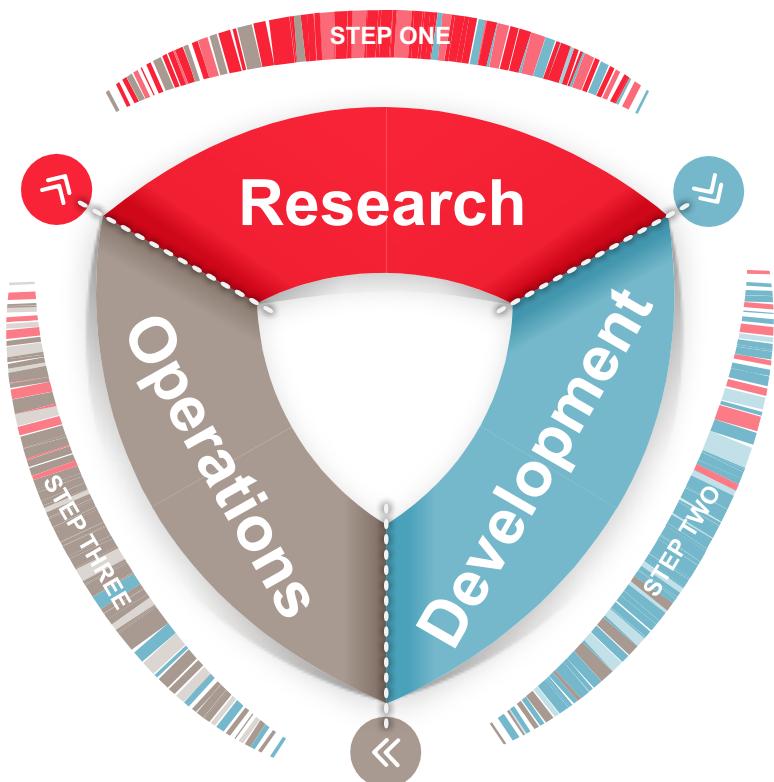
- We expect you to come to class
- Each Friday class will usually be split between lectures by the instructors and student-led presentations, whiteboard exercises, and assignment discussions.
- Student-led discussions will focus on applying the lecture concepts to project assignments

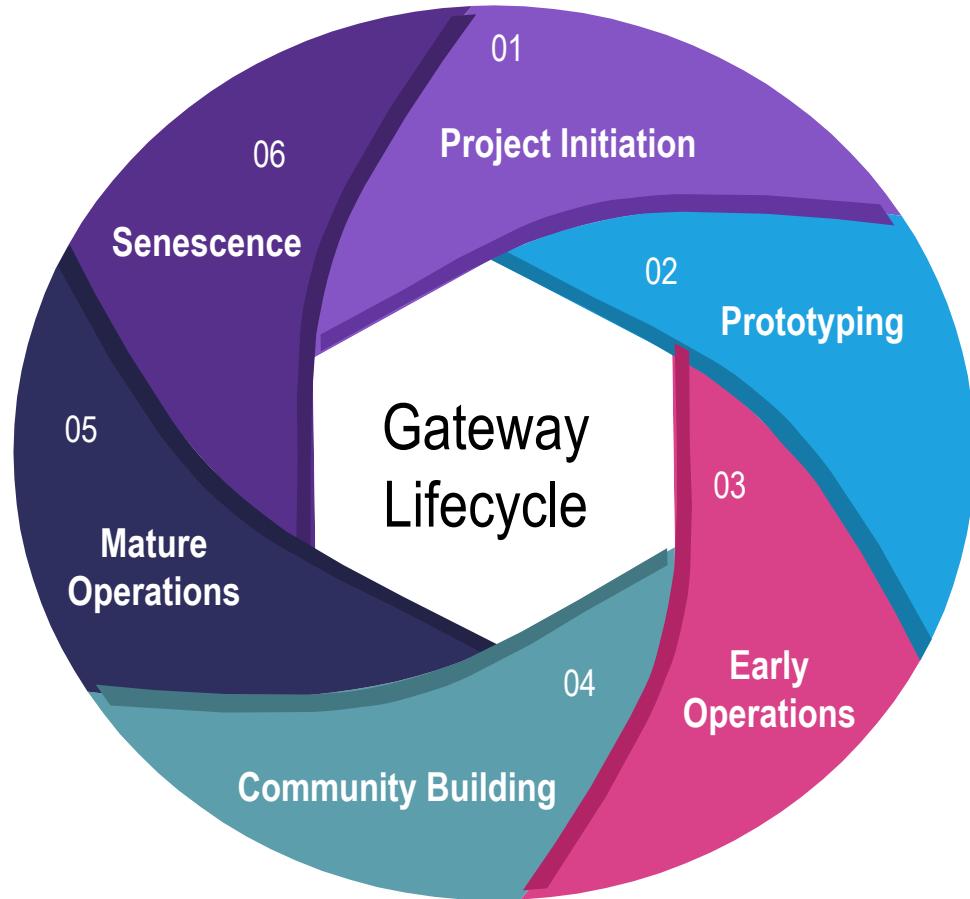


INDIANA UNIVERSITY

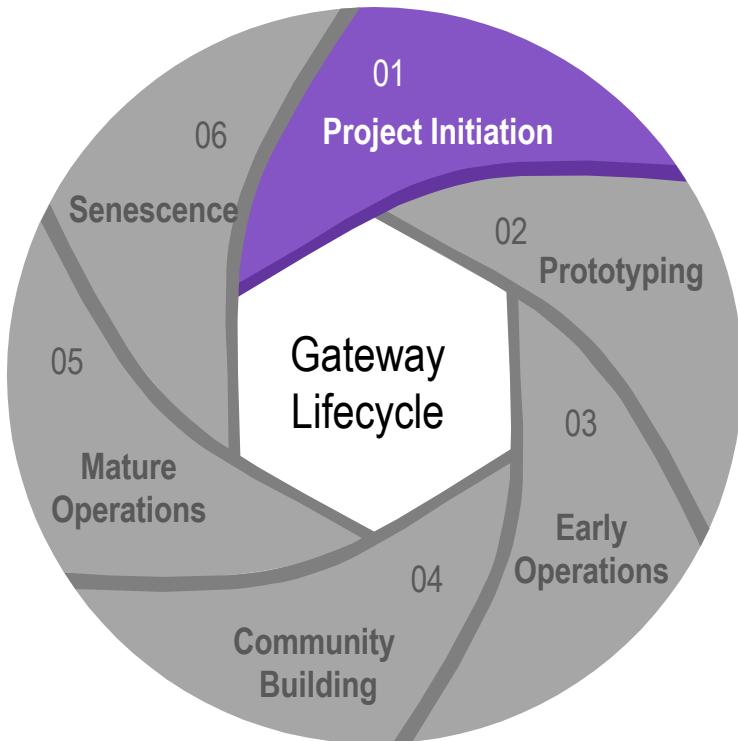
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Lifecycle of typical Science Gateway





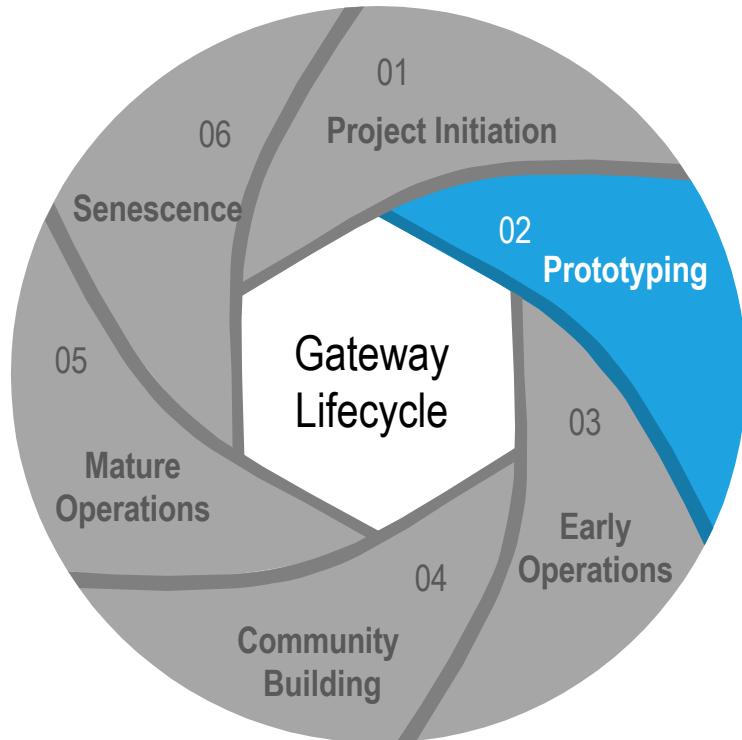
Project Initiation: you want to build a gateway



Technical Challenges

- Establishing your technical base
- Choosing technologies that maps your value proposition.
- Assemble team: Who do you need

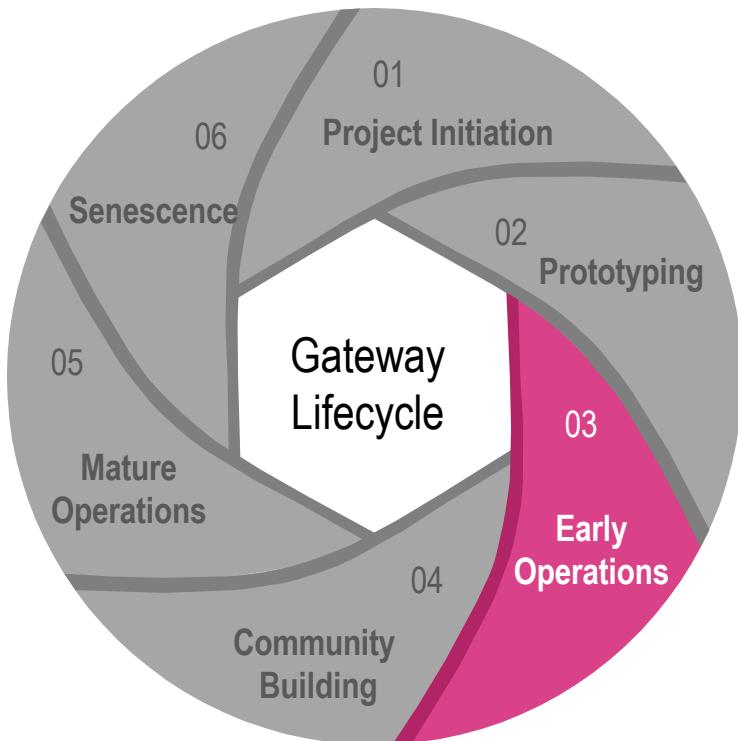
Prototyping: Choosing a framework vs. building things yourself



Technical Challenges

- Evaluating Frameworks: How well can they be adapted to what you want?
- DIY: assembling the right team, establishing the right engineering and operations practices
- DIY: choosing third party systems

Early Operations: Transitioning prototype into “production”

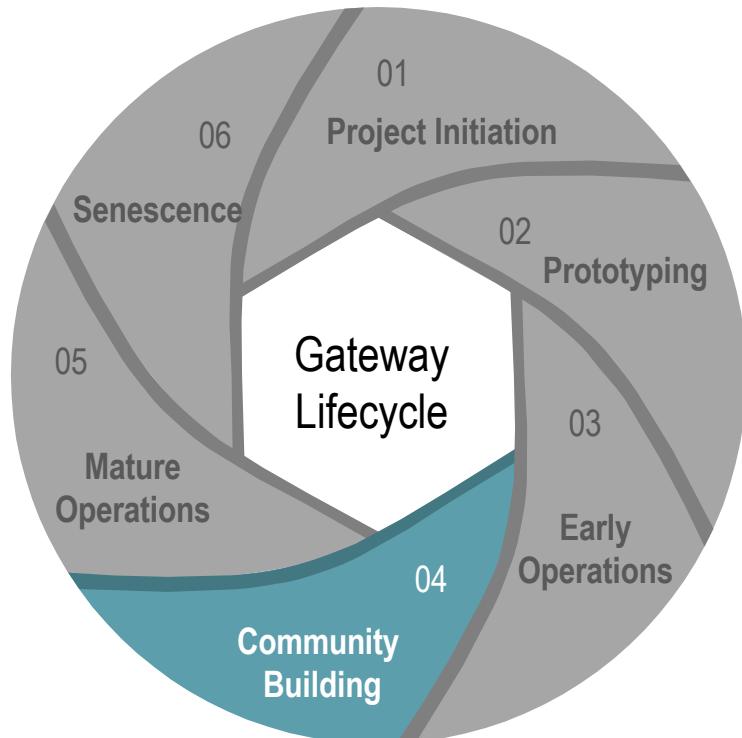


Technical Challenges

- High Availability
- Scaling up your technology for more users
- Monitoring
- Loss of key tech people;
Onboarding new developers

Community Building:

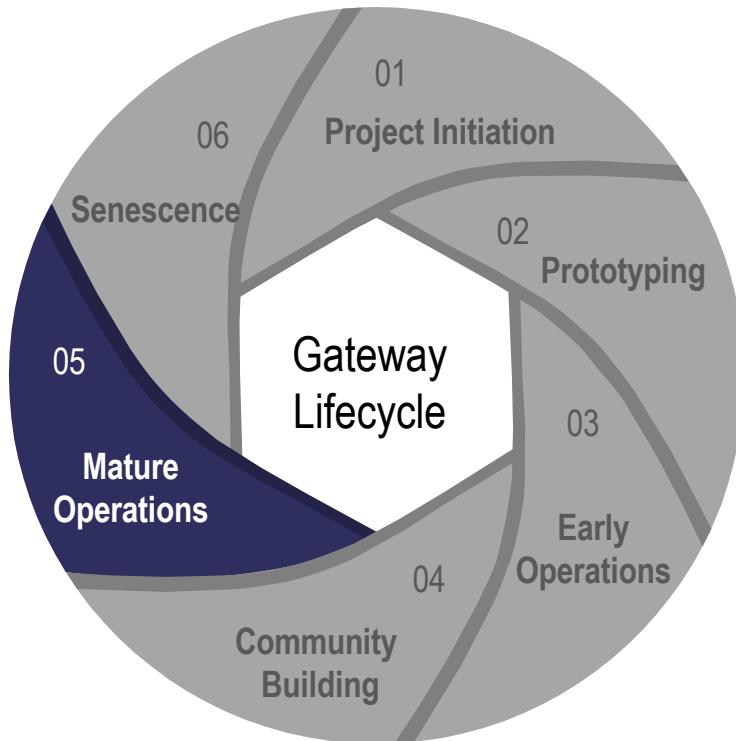
Have a technology but need to grow your user base



Technical Challenges

- Adding features expeditiously
- Managing feature creep

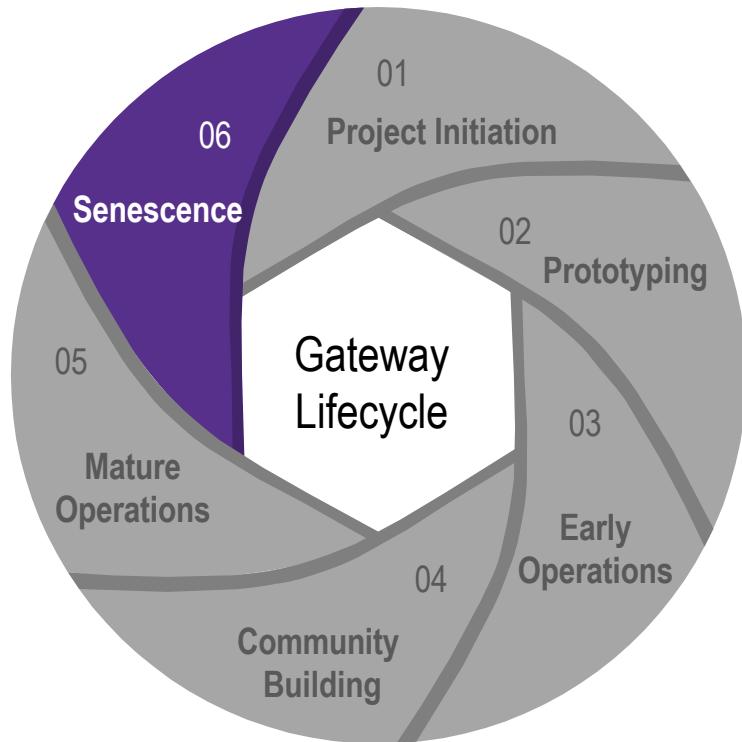
Mature Operations: Mature in-house systems + Community



Technical Challenges

- Technical Debt: dealing with earlier short cuts and shortcomings
- Expanding your team's roles
- Difficulties changing system to take advantage of new technology
- Loss of key people, onboarding
- Plateaued usage

Senescence: Aging technology, value proposition may need revision



Technical Challenges

- Wholesale tech stack change
- Back to square one

Your First Assignment: Brainstorm a Gateway

Pick a “Software as a Service” example which you cannot live with being offline



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING