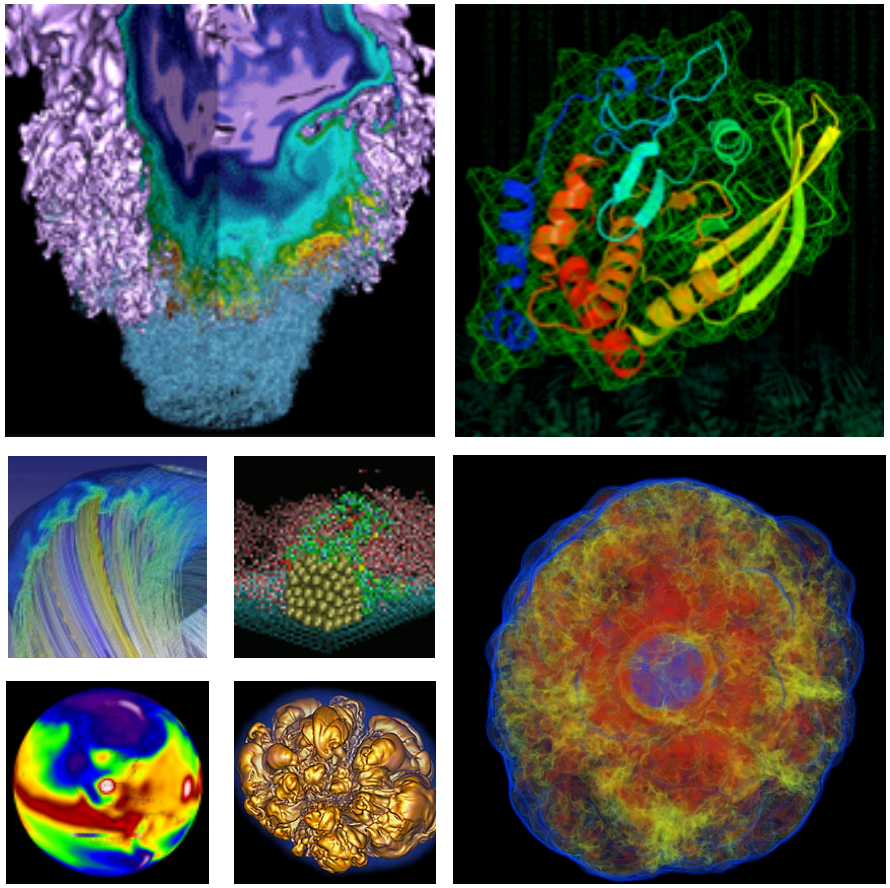
# Science Gateway Patterns and Practices:

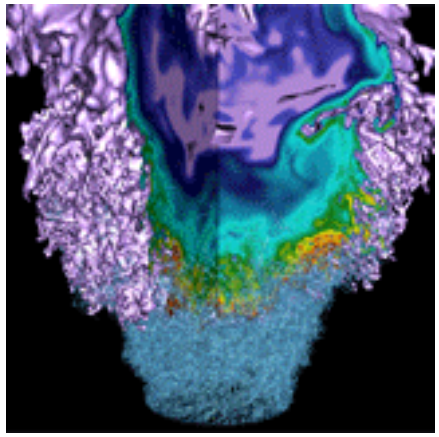
## Experiences Deploying Science Gateways at a Supercomputing Center

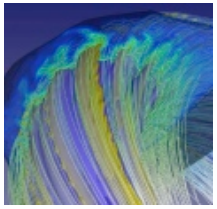


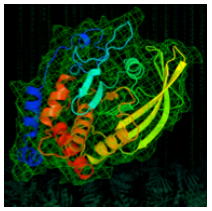

**Shreyas Cholia**
**Annette Greiner**
**Rollin Thomas**
**NERSC/LBL**

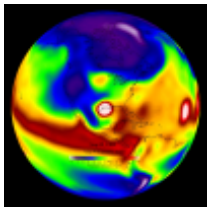


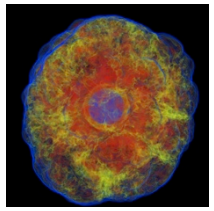
Gateways 2017 – Ann Arbor MI, Oct 24 2017

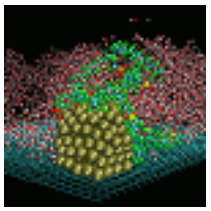




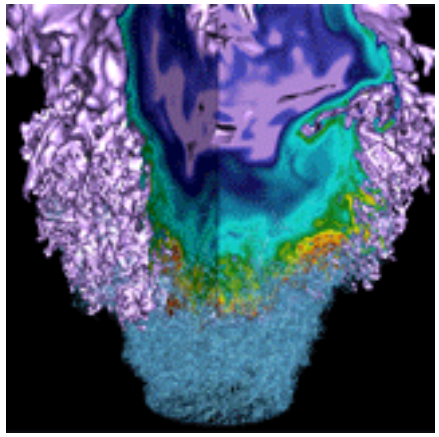


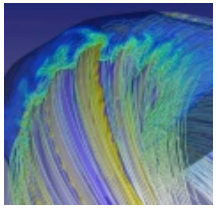


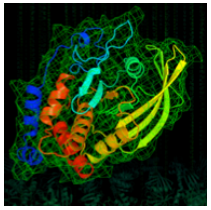


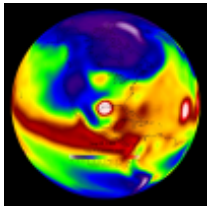


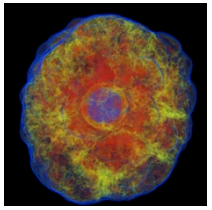
# Introduction

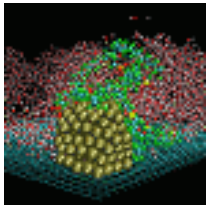










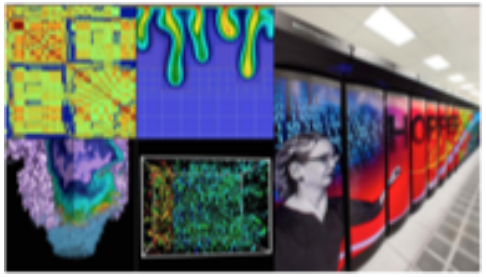


# NERSC

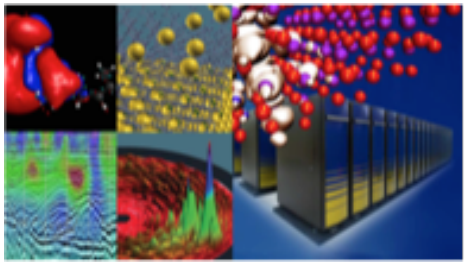
**NERSC at Lawrence Berkeley Lab is the production HPC & Data Facility for the Department of Energy Office of Science**

**NERSC has been building and deploying science gateways since 2009**

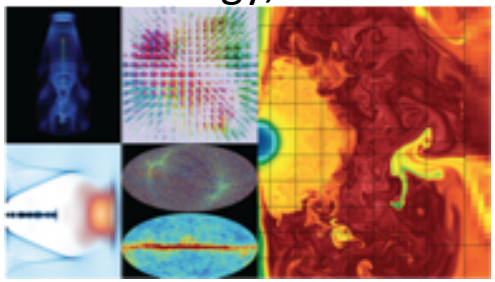
Bio Energy, Environment



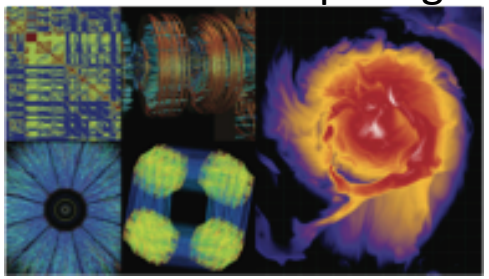
Advanced Computing



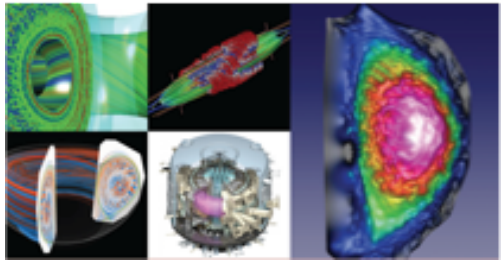
Materials, Chemistry, Geophysics



High Energy Physics



Nuclear Sciences



Fusion, Plasma Physics

U.S. DEPARTMENT OF ENERGY | Office of Science

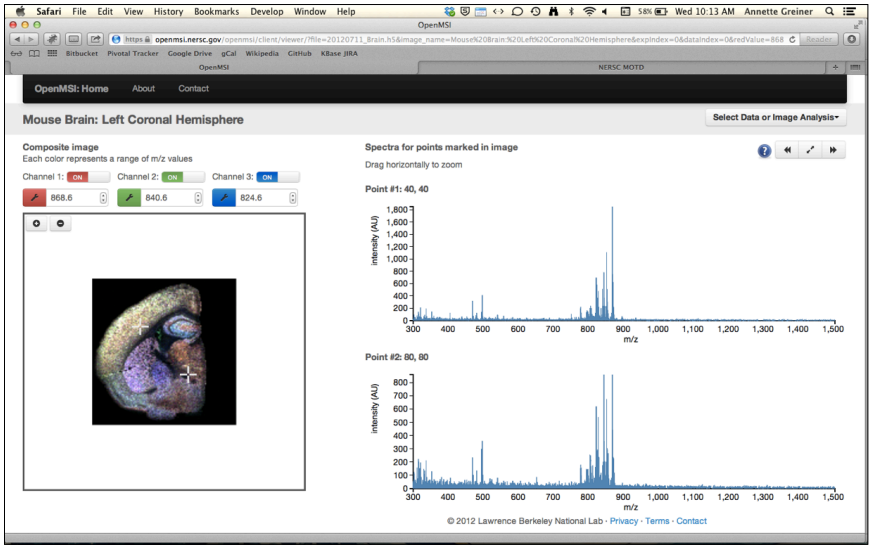
BERKELEY LAB

# Science is Collaborative

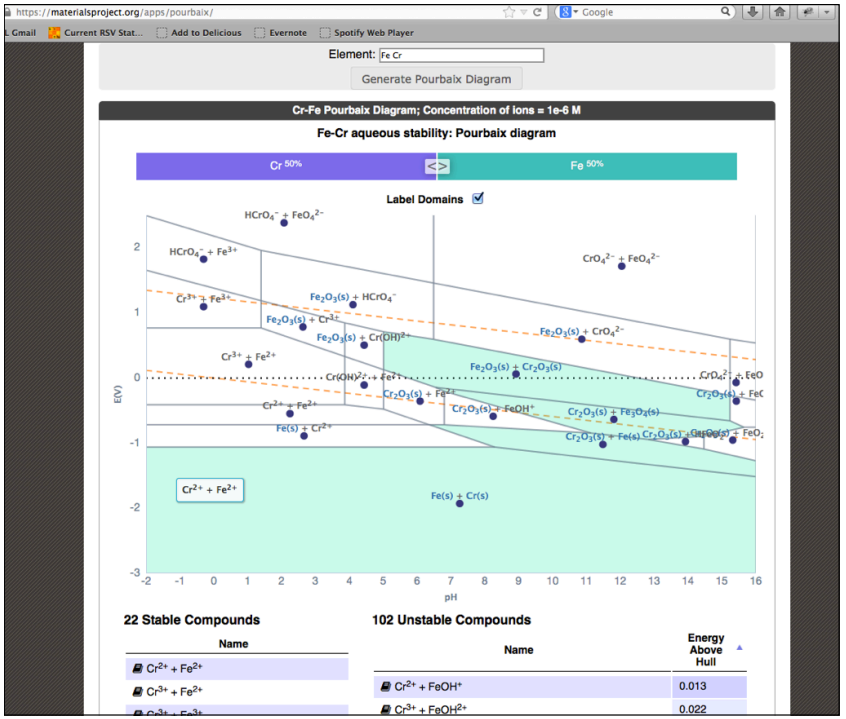
- Science is now a collaborative effort
- Large teams of people
- Requires shared access to compute and data resources

# Harnessing The Power Of The Web

- **Web interfaces enable science-centric views to data**
- **Enable new discoveries and insights through collaborative tools and rich interfaces**

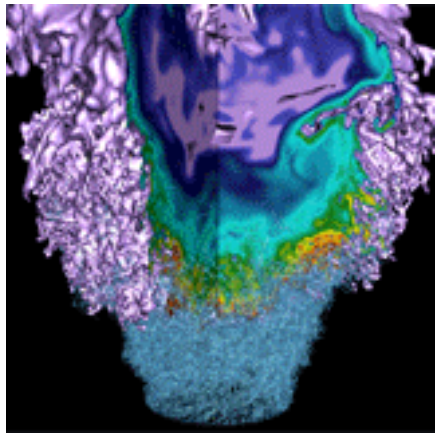


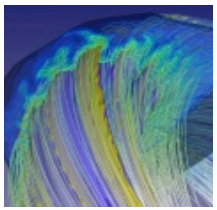

https://openmsi.nersc.gov

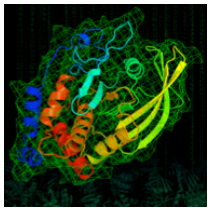


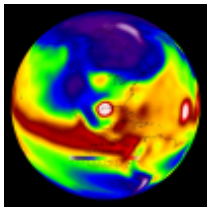

https://materialsproject.org

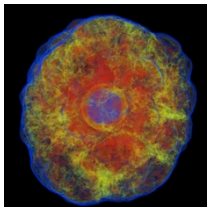
# Patterns and Practices in Science Gateways

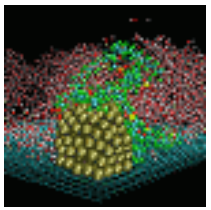




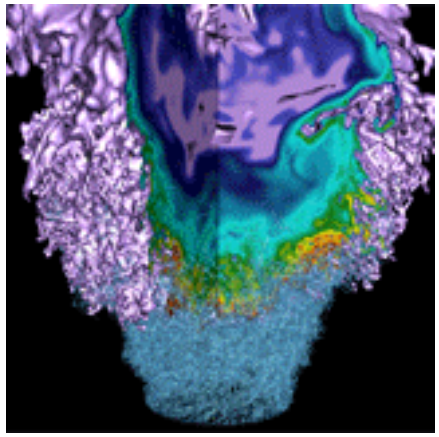


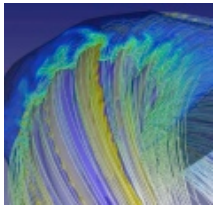


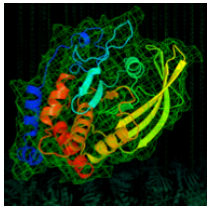


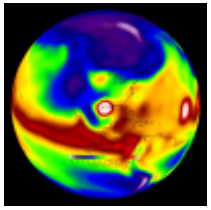


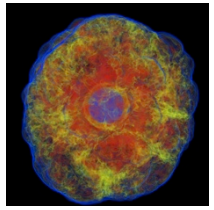
# Sharing Data Over The Web

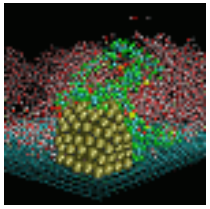












# Sharing (your data) is important!

**J.M. Wicherts, M. Bakker, and D. Molenaar:**

**Willingness to Share Research Data Is Related to the Strength of the Evidence and the Quality of Reporting of Statistical Results**

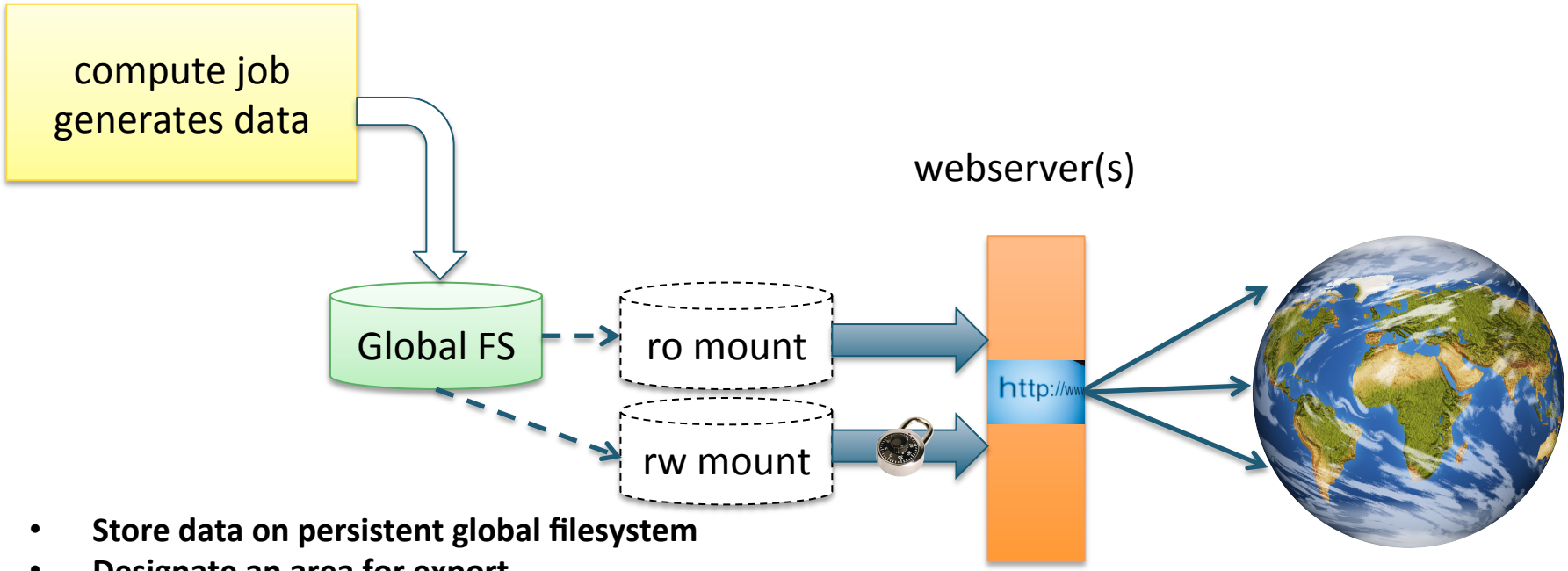
***PLoS ONE*, 6(11): e26828, 2011, doi:10.1371/journal.pone.0026828.**

***Content for this slide courtesy Greg Wilson, Software Carpentry***

# Data Sharing

- Open access to data through common interfaces
- Use your own local tools with centrally managed data
- Everyone sees the same data – better collaboration

# Data Sharing Over The Web

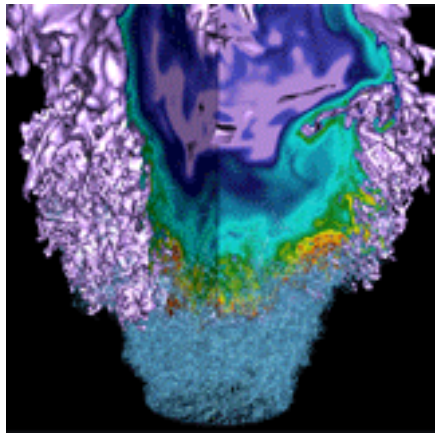



- **Store data on persistent global filesystem**
- **Designate an area for export**
- **Mount export area RO on a web node using a shared FS**
- **Run a webserver on web node to share with world**
- **Add auth to web server for rw**

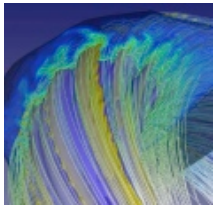
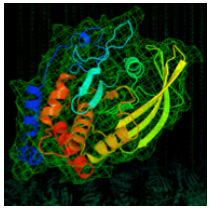
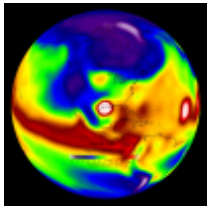
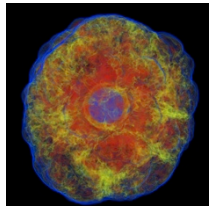
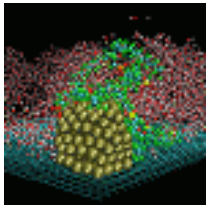
# HTML makes it easy

- **Also easy to build a simple science gateway this way by simply adding HTML + JS to your gateway to provide a nicer UI**

# Web Frameworks For Science



NERSC

# Advantage of Frameworks

- Go beyond simple HTML wrapping your data
- Automatically adds a bunch of scaffolding for common functionality
- Allow for more complex "Full Stack" features to interact with backend resources

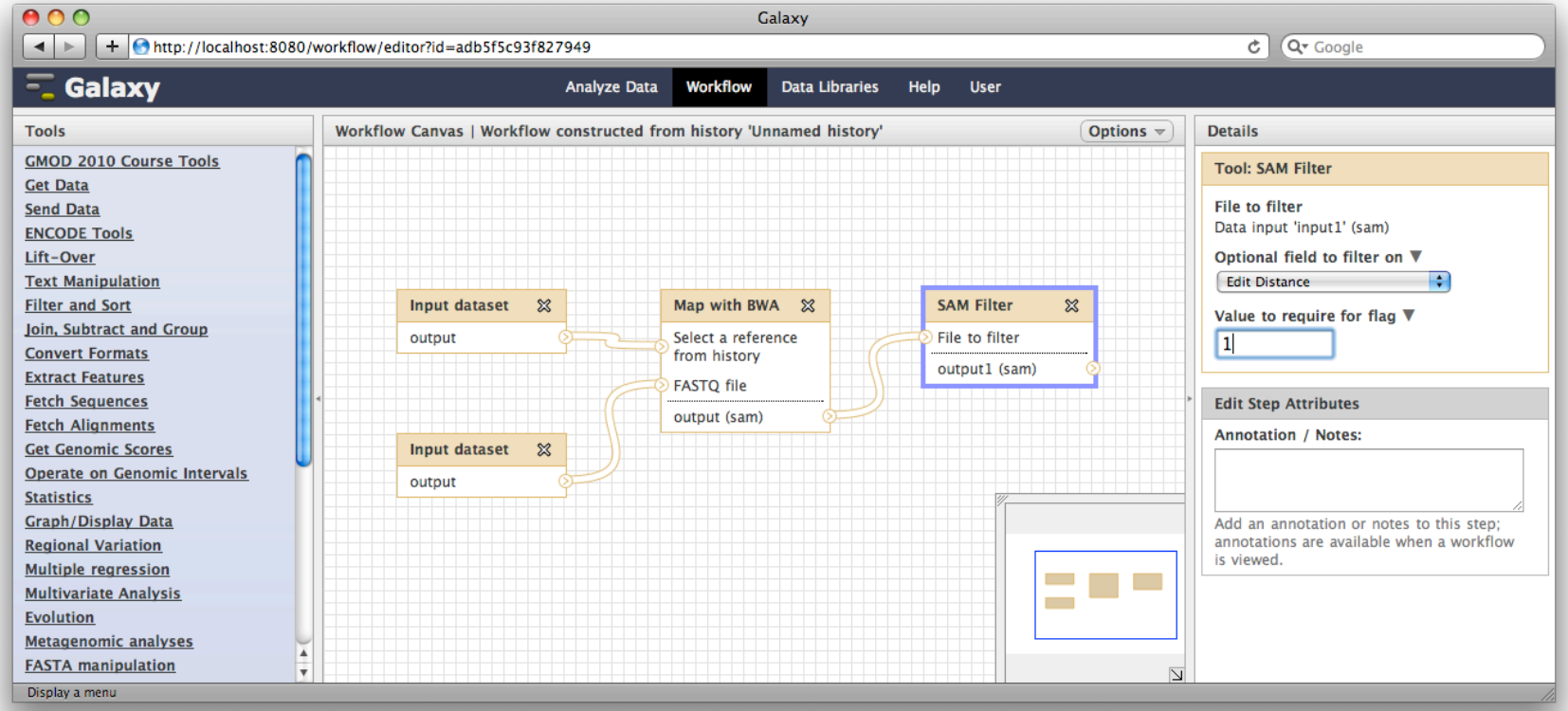
# MVC Frameworks

- **Support Model-View-Controller (MVC) pattern**
  - eg. Django, Flask, Ruby-On-Rails, Angular, Backbone.js
- **Scaffolding to build out application**
  - Authentication, Routes, DB access, Templating etc.

# Science Gateway Frameworks

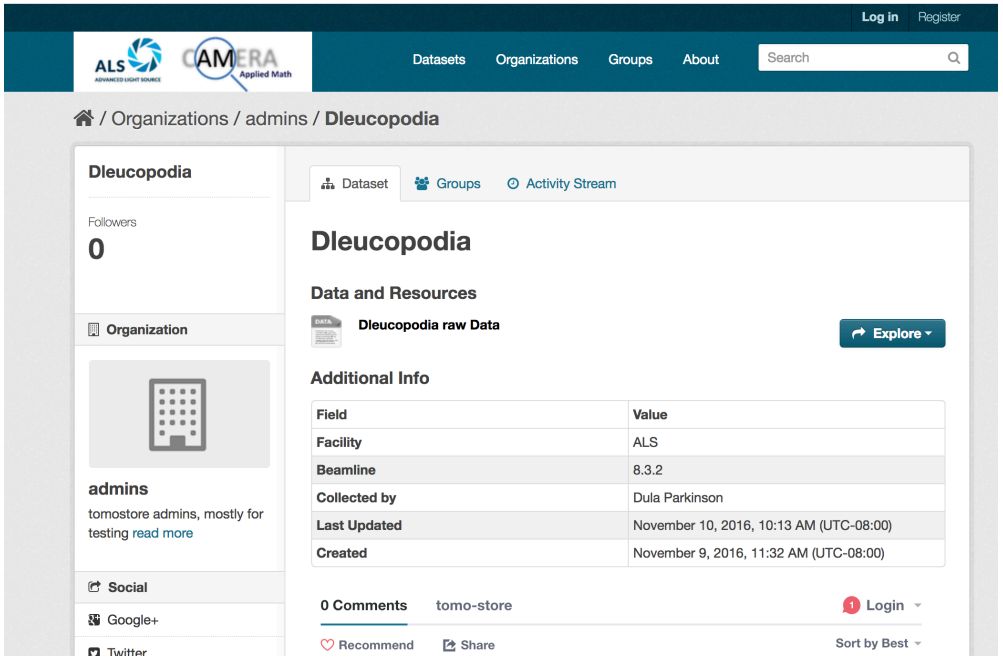
- **Open source frameworks that can be customized for specific science use cases:**
  - HubZero, Galaxy, DataOne Metacat, CKAN etc.
- **Out of the box functionality for science gateways**
- **Plugin + configuration based approach to domain specific customizations.**

# Galaxy

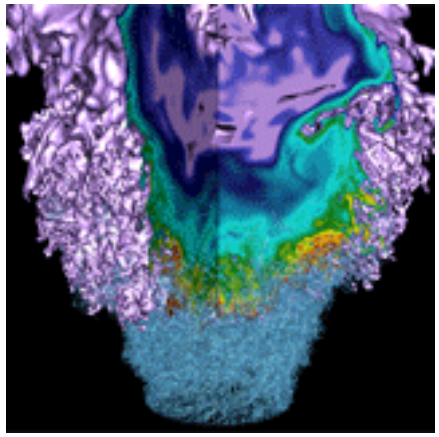

Galaxy
http://localhost:8080/workflow/editor?id=adb5f5c93f827949
Google
Galaxy
Analyze Data
Workflow
Data Libraries
Help
User
Tools
GMOD 2010 Course Tools
Get Data
Send Data
ENCODE Tools
Lift-Over
Text Manipulation
Filter and Sort
Join, Subtract and Group
Convert Formats
Extract Features
Fetch Sequences
Fetch Alignments
Get Genomic Scores
Operate on Genomic Intervals
Statistics
Graph/Display Data
Regional Variation
Multiple regression
Multivariate Analysis
Evolution
Metagenomic analyses
FASTA manipulation
Workflow Canvas | Workflow constructed from history 'Unnamed history'
Options
Input dataset
output
Input dataset
output
Map with BWA
Select a reference from history
FASTQ file
output (sam)
SAM Filter
File to filter
output1 (sam)
Details
Tool: SAM Filter
File to filter
Data input 'input1' (sam)
Optional field to filter on
Edit Distance
Value to require for flag
1
Edit Step Attributes
Annotation / Notes:
Add an annotation or notes to this step; annotations are available when a workflow is viewed.
Display a menu

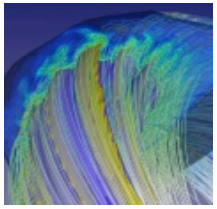
# CKAN

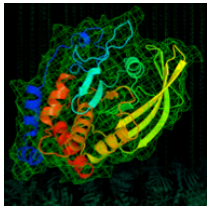
- **CKAN – data management with metadata**

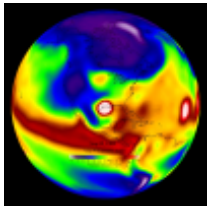


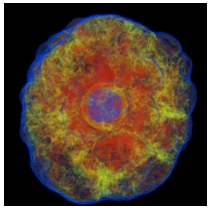
# Web IDEs and Interactive HPC

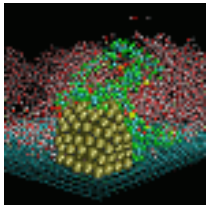








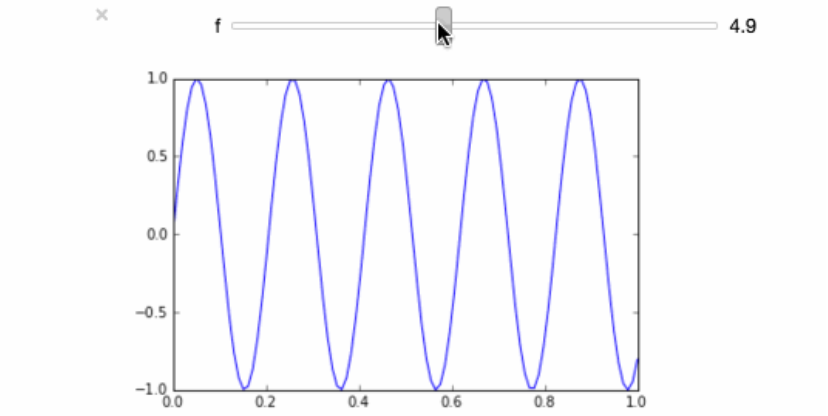




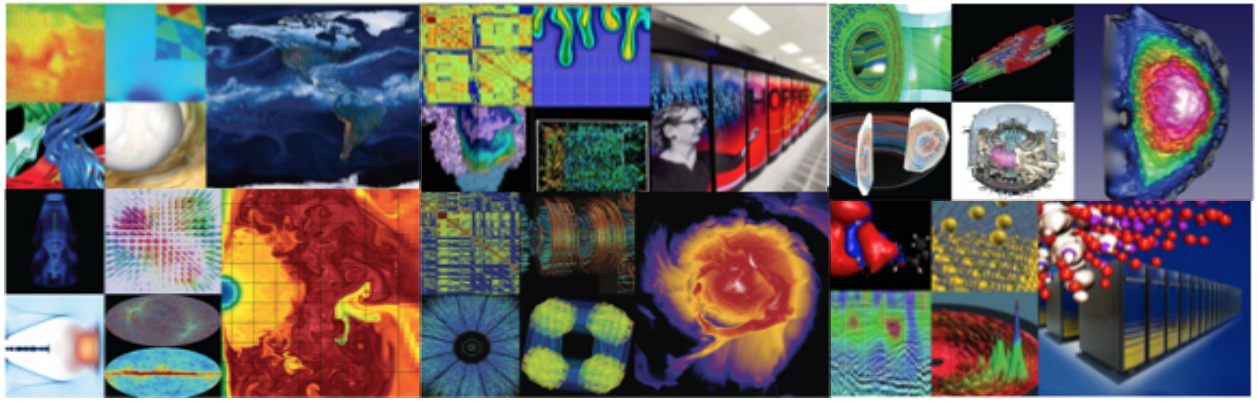
# Interactive Environments

- **Time to science – Reduce the time to insight by tightening the loop between the human and computation**
- **Exploratory Computing!**

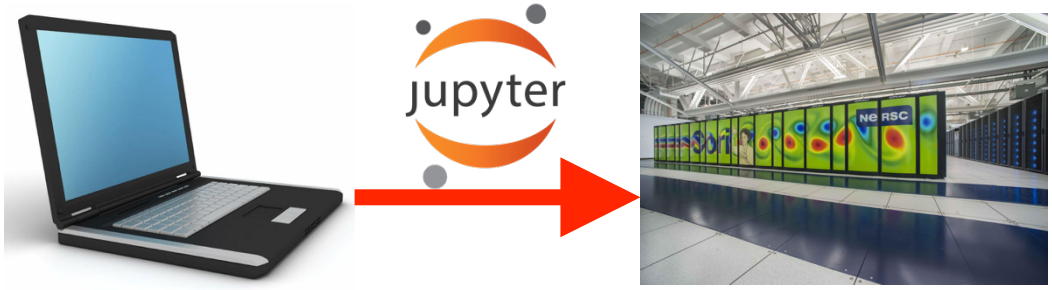
```
In [22]: from IPython.html.widgets import *
         t = arange(0.0, 1.0, 0.01)

         def pltsin(f):
             plt.plot(x,sin(2*pi*t*f))
             plt.show()

         interact(pltsin, f=(1,10,0.1))
```



# Enable Big Science

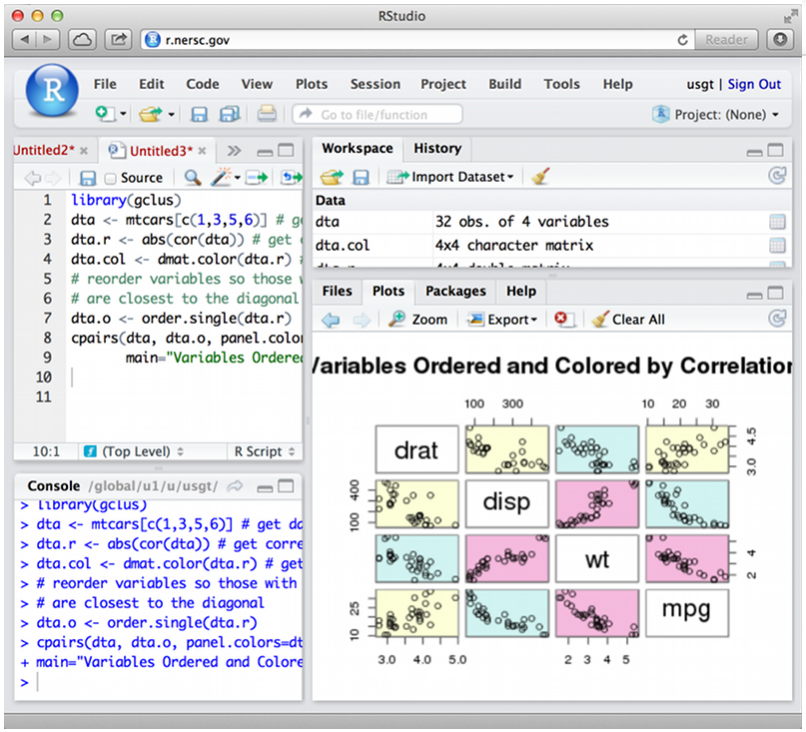


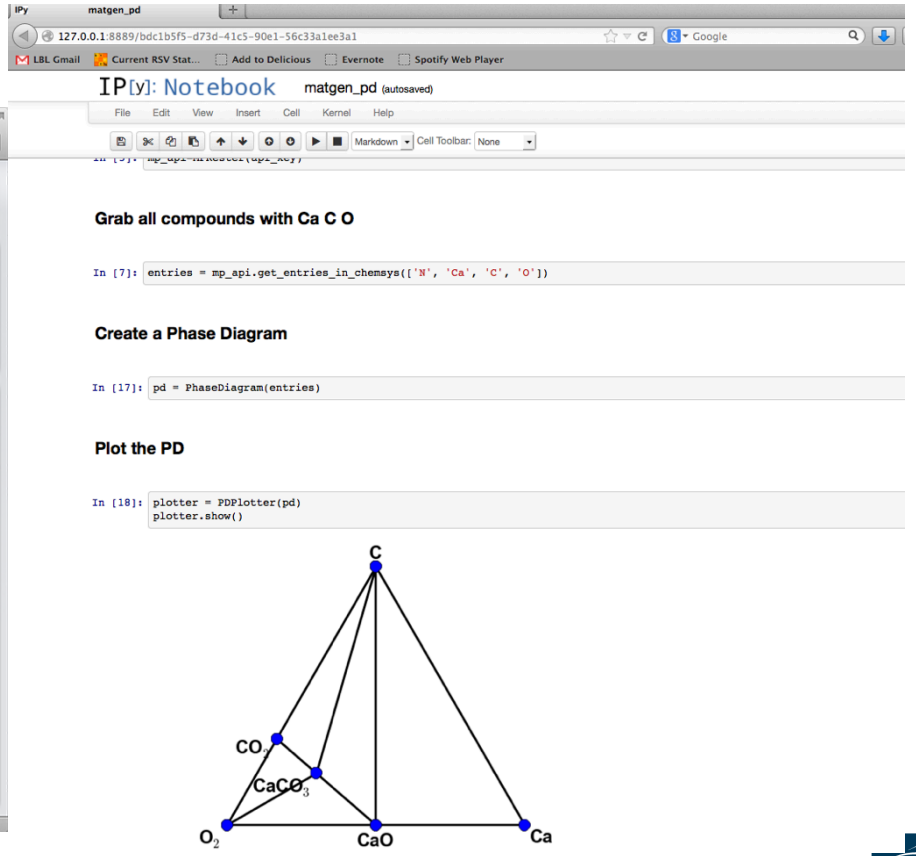
**Deep** Questions → **Expensive** Detector Technologies → **Insightful** Real time predictions?
Instruments/Facilities
High-bandwidth Networks
Simulations
Exploratory analysis?
Decision making?



# Interactive Web Enviroments

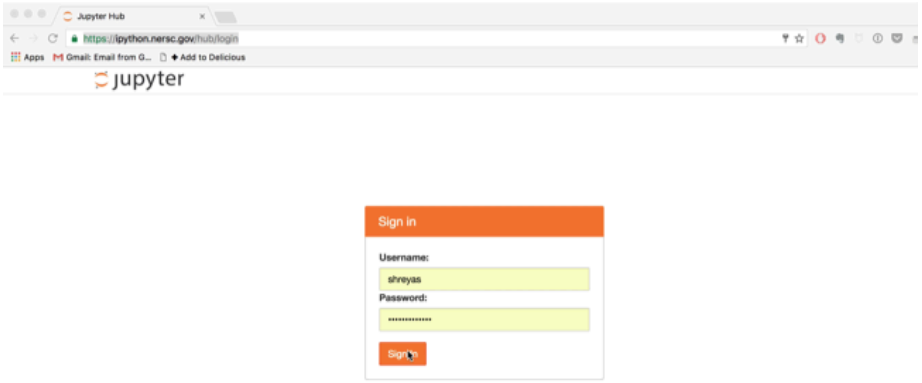
- Combine the expressiveness of programming tools with web GUIs
- RStudio, Mathematica, Jupyter

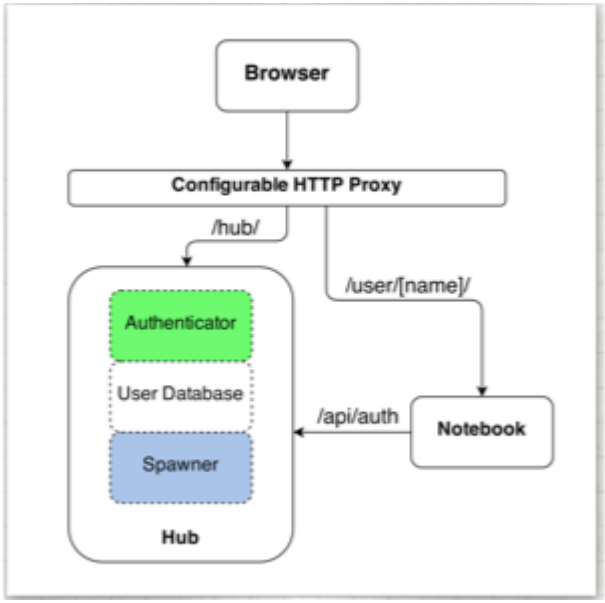


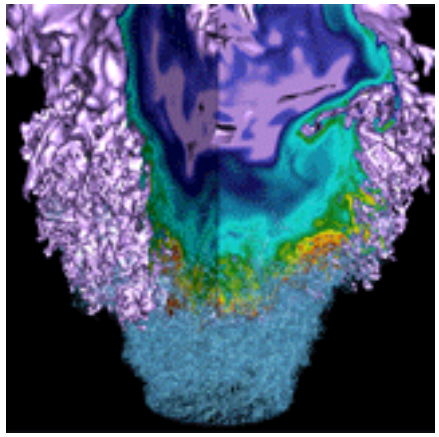
# Jupyterhub: Jupyter As A Service

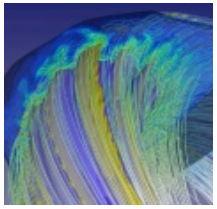
- Service to deploy notebooks in a multi-user environment
- Manages user authentication, notebook deployment and web proxies

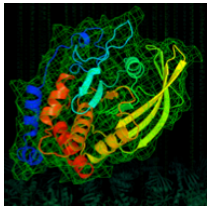


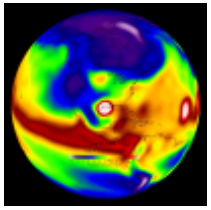



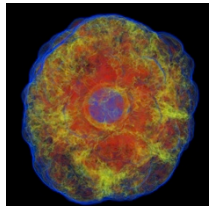
# HTTP APIs

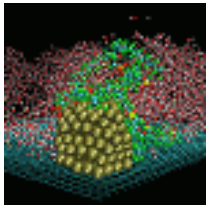












# APIs APIs APIs

- **HTTP APIs are everywhere and form the backbone of the modern web**
- **Separate front-end views from backend through API**

# APIs Drive Programmatic Access to HPC

- **Enable heavy duty server side operations and long running tasks**
- **Perform sub-selection of data – pull down only what is needed in client and render it in the browser**
  - eg. OpenDAP connects HDF5 datasets with web clients

# The 1-minute intro to REST APIs

- **Use HTTP verbs in conjunction with URLs**
  - GET, POST, PUT, DELETE
- **Verb + URL + parameters = function call**
- **Return structured data**
- **For instance (from NEWT API):**


| VERB | RESOURCE | DESCRIPTION |
|---|---|---|
| POST | /queue/R | Submits POST data to queue on R; returns job id |
| GET | /file/R/path/ | Returns directory listing for /path/ on R |
| GET | /account/user/U | Returns user account info for U |
| DELETE | /queue/R/id | Deletes jobs from Queue |

- **Structured Output (eg. JSON):**

```
{"status": "OK",
"output":["status": "up", "system": "hopper"}, {"status": "up", "system": "carver"}]
"error": ""}
```

# Standardize Common Operations

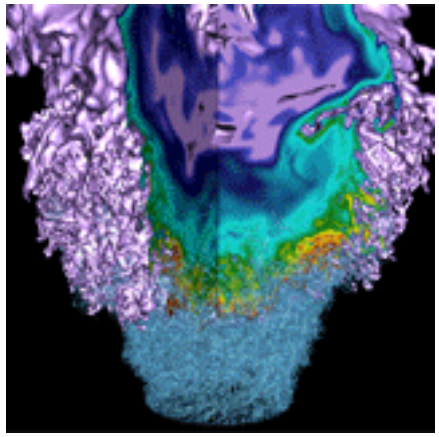
- **Capture common operations in a standard way**
- **Don't reinvent the wheel for each portal**
- **Eg. NEWT, Agave, Airavata**
  - Authentication
  - Jobs
  - Files
  - Workflows
  - Accounting
  - Database
  - Commands
  - Events
  - …

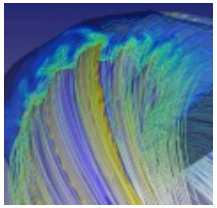
# JAM Stack

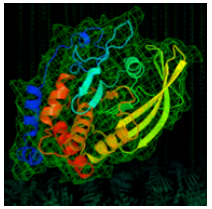
**JAMstack**: noun \'jam-stak'\
Modern web development architecture based on client-side **JavaScript**, reusable **APIs**, and prebuilt **Markup**.

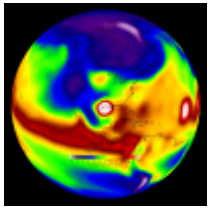
This model also allows for rich hosted content (JS + HTML) that interacts with an API on the backend

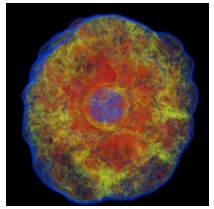
# Federated Identity and Single Sign-On

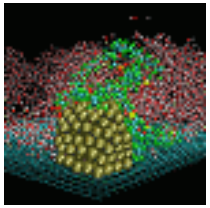












# Federated Identity

- **One identity to rule them all**
- **Allow users to log-in using common credentials**
  - home institution (Shibboleth)
  - External providers like Google, Facebook, Github etc.
- **Send attributes to gateway service provider and map federated ID to local roles or accounts**

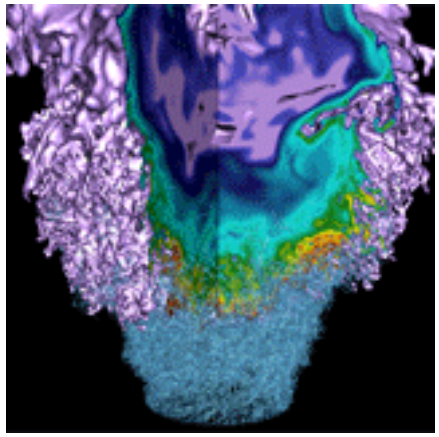
# Authentication Aggregators

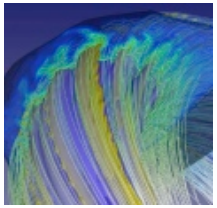
- Janrain
- Auth0
- Globus

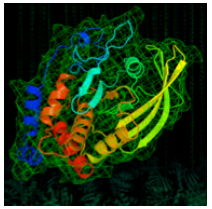
Provide common Authentication services that integrate multiple auth providers

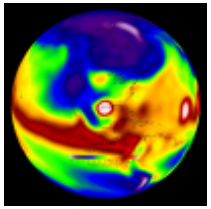
Starting to see increasing use of JWT

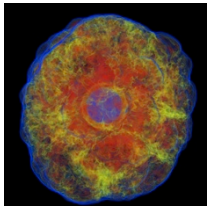
# Edge Services

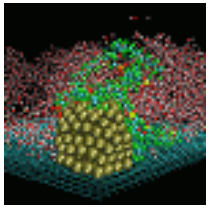












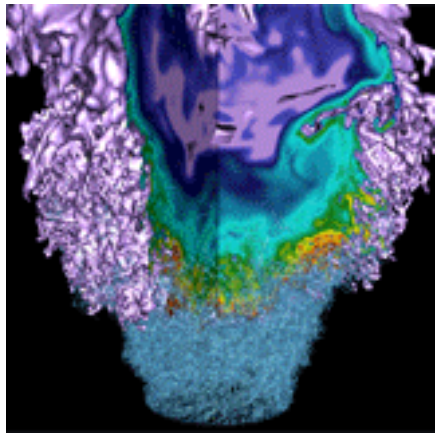
# Bridge to Everywhere

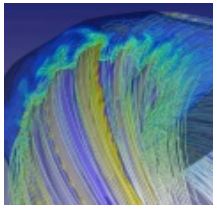
- Edge Service nodes act as connectors between HPC and the wider internet
- Live on the same network as HPC system, but have external access
- Database services, message queues, CI services, data transfer nodes, web portals, API services

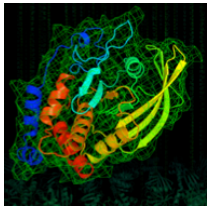
# Science DMZ

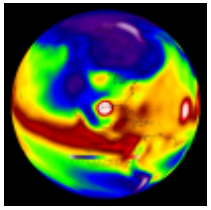
- **The Science DMZ is a portion of the network, built at or near the campus or laboratory's local network perimeter that is designed such that the equipment, configuration, and security policies are optimized for high-performance scientific applications**

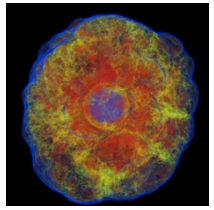
# Data Transfer Services

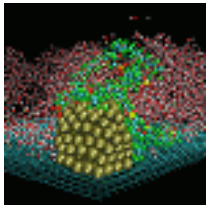










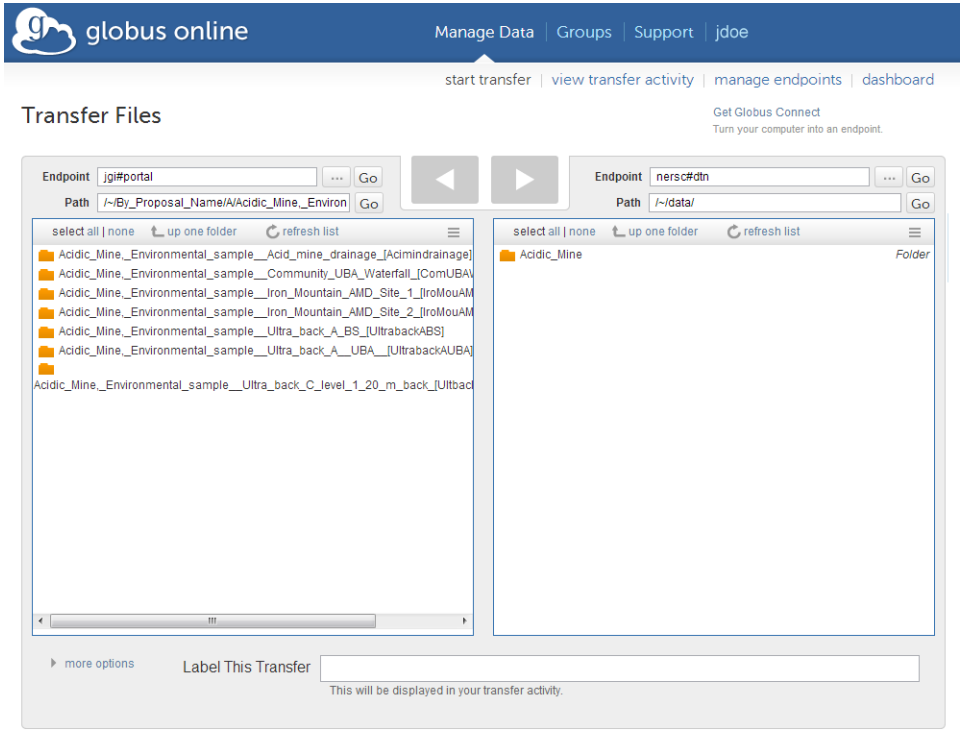


# Data Transfer Nodes

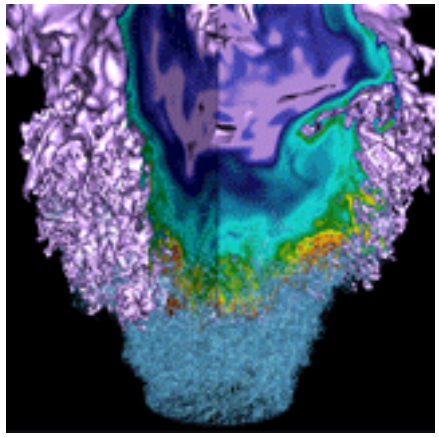
- **Data Transfer Nodes (DTN) are dedicated servers for moving data at NERSC.**
  - Servers include high-bandwidth network interfaces & are network is tuned for efficient data transfers
  - Monitored bandwidth capacity between NERSC & other major facilities such as ORNL, ANL, BNL, SLAC…
  - Provide direct access to global NERSC file systems
- **Data Transfer Services – Globus**
- **Science DMZ model**

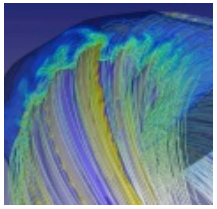
# Globus Integrated with Gateways

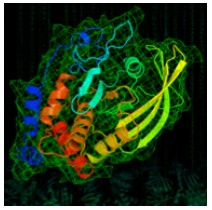
- **Detailed documentation for integrating Gateways with Globus API**
- **Provides a much higher performance interface for users to get data compared to HTTP web sharing**

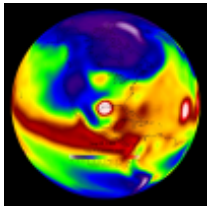


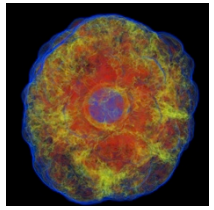
# Cloud Hosted Portals

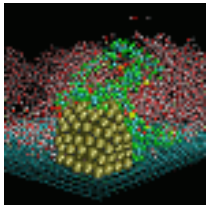












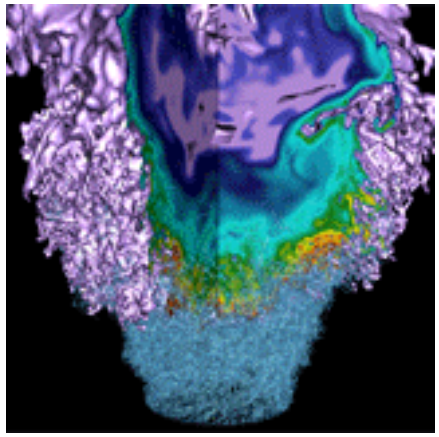
# Cloud Deployments

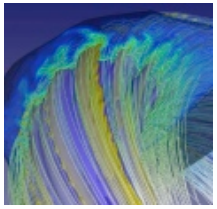
- Gateway Application deployed in cloud service
- Communicate with HPC only through APIs and edge services

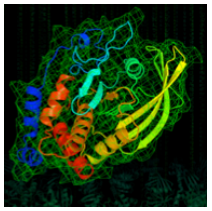
# When in the Clouds

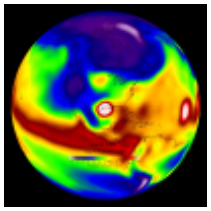
- **HPC hosting environments have policy restrictions – deploying in the cloud completely frees up application constraints**
- **Clean separation of functionality**
- **Cross-Origin Resource Sharing (CORS) allows you to deal with cross site issues in a standard way**

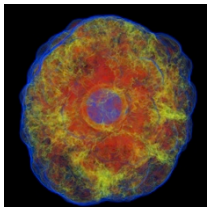
# Containers

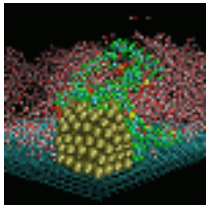










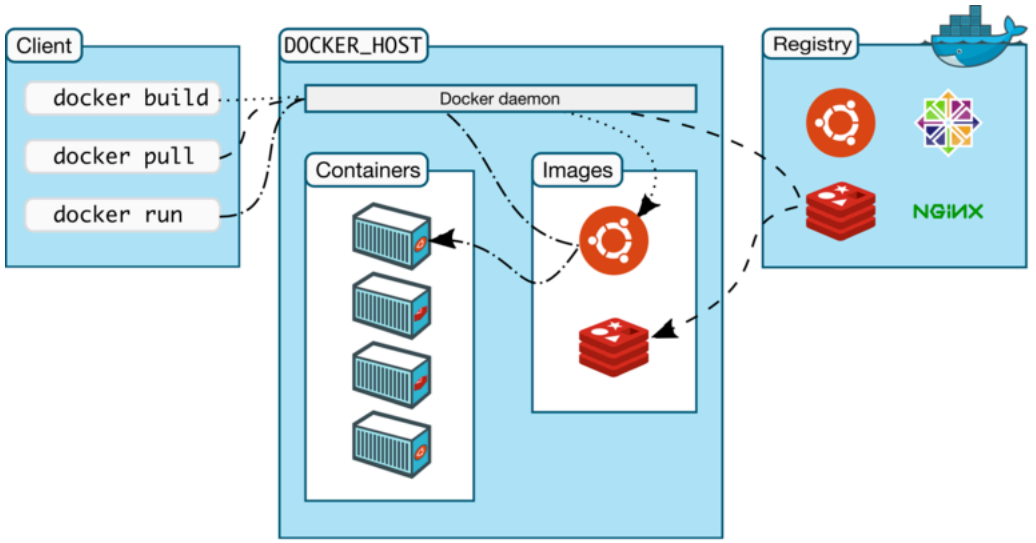


# Containers

- **Lightweight form of virtualization where the user can bundle the entire application stack**
- **Portability**
- **Scalability**

# Docker

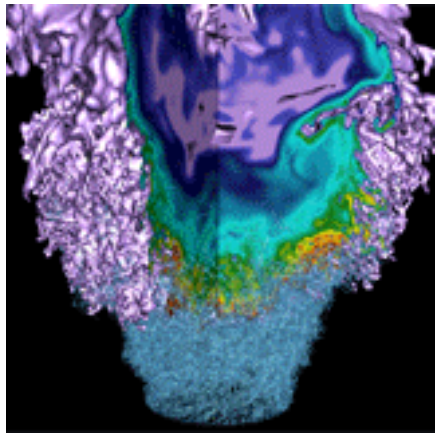
- **Quickly becoming most popular container technology**
- **Build docker images with full gateway app locally**
- **Deploy in multiple environments**
  - Laptop / Cloud / HPC (Shifter) / Gateway Hosting

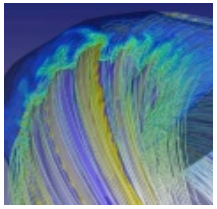


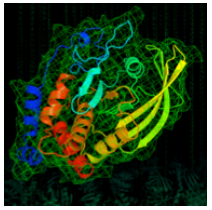
# SPIN

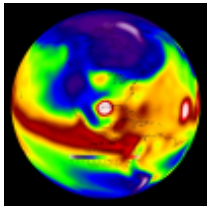
- **NERSC Service to manage orchestration of containers using Rancher**
- **Test locally; Docker push image; click a button to deploy and it magically works**
  - OK – not quite magic yet but that is the goal
- **Automatic load-balancing and scaling**

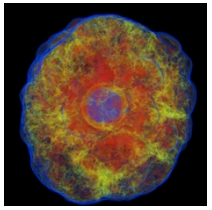
# Fin

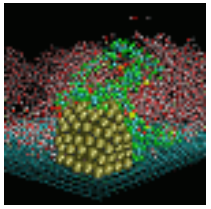












# What Next?

- This is by no means a comprehensive list
- Say hi and tell us what you are doing
- We'd love to do a community oriented Tech Radar style survey of tools and technologies people are using

# Thanks

- **Contact Us:**
  - Shreyas Cholia scholia@lbl.gov
  - Annette Greiner agreiner@lbl.gov
  - Rollin Thomas rcthomas@lbl.gov