SciServer

Collaborative data-driven science

SciServer Compute

Mike Rippin

JOHNS HOPKINS UNIVERSITY

NSF

idies

Agenda

- Background and History of SciServer
- Major Objectives
- Current System
- SciServer Compute – Now
- SciServer Compute – Future
- Q&A

SciServer

Collaborative data-driven science

Background

SciServer

"The Project aims to create a sustainable collaborative ecosystem built around several large scientific data sets for the broader science community, based upon the expertise developed for the Sloan Digital Sky Survey (SDSS) SkyServer and associated projects."

Project Management

- NSF Cooperative Agreement
- 5 years duration, just completed first 3
- Development of Cyberinfrastructure
- Science Driven

Motivation and History

- Started with the SDSS SkyServer
- Goal: instant access to rich content
- Idea: bring the analysis to the data
- Interactive access at the core

Where Are We Going?

- Interactive science on petascale data
- Create scalable open numerical laboratories
- Large footprint across many disciplines
- Use commonly shared building blocks
- Major national and international impact

SciServer

Collaborative data-driven science

Current System

JOHNS HOPKINS UNIVERSITY

NSF

idies

SciServer : Core Functions

Cyber Infrastructure

Science Collaboration

SDSS Integration

Outreach & Education

SciServer : Core Functions

Cyber Infrastructure

Science Collaboration

SDSS Integration

Outreach & Education

Main components:

Database storage & Query: CasJobs

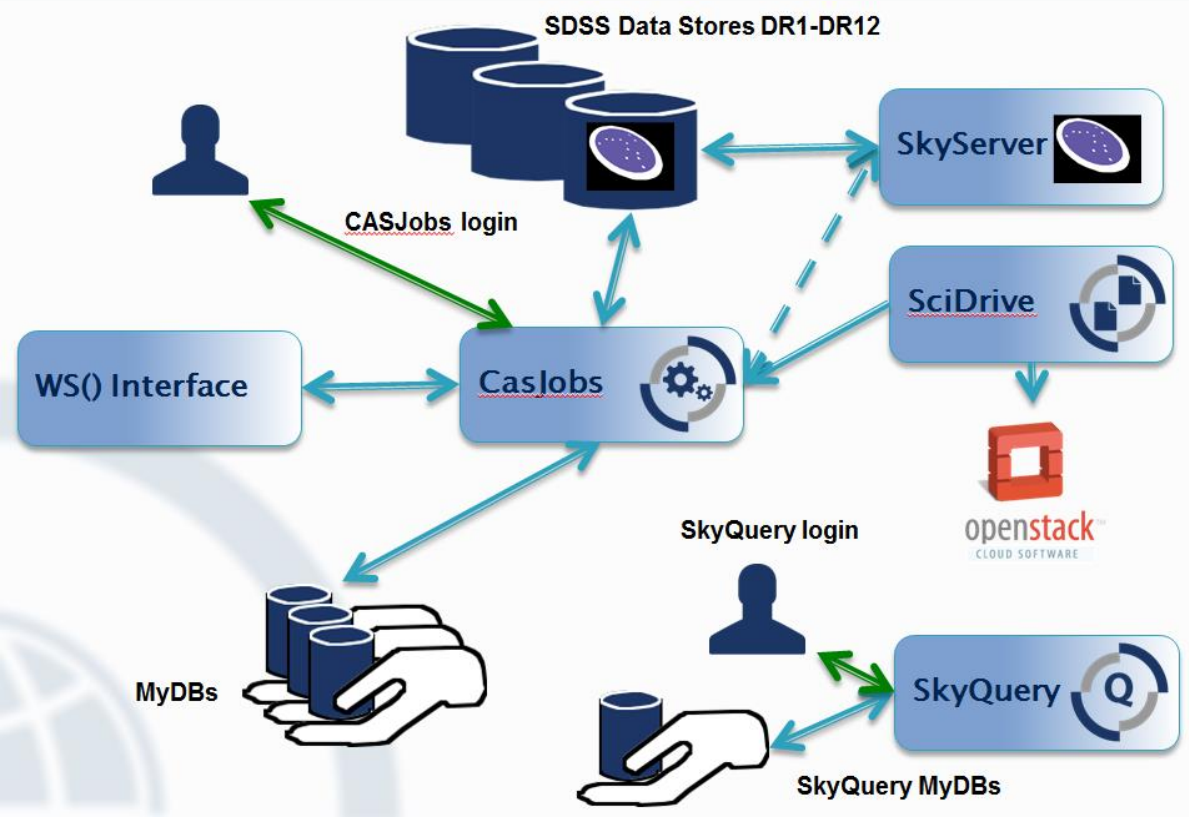
Data analysis: Compute

File storage: SciDrive

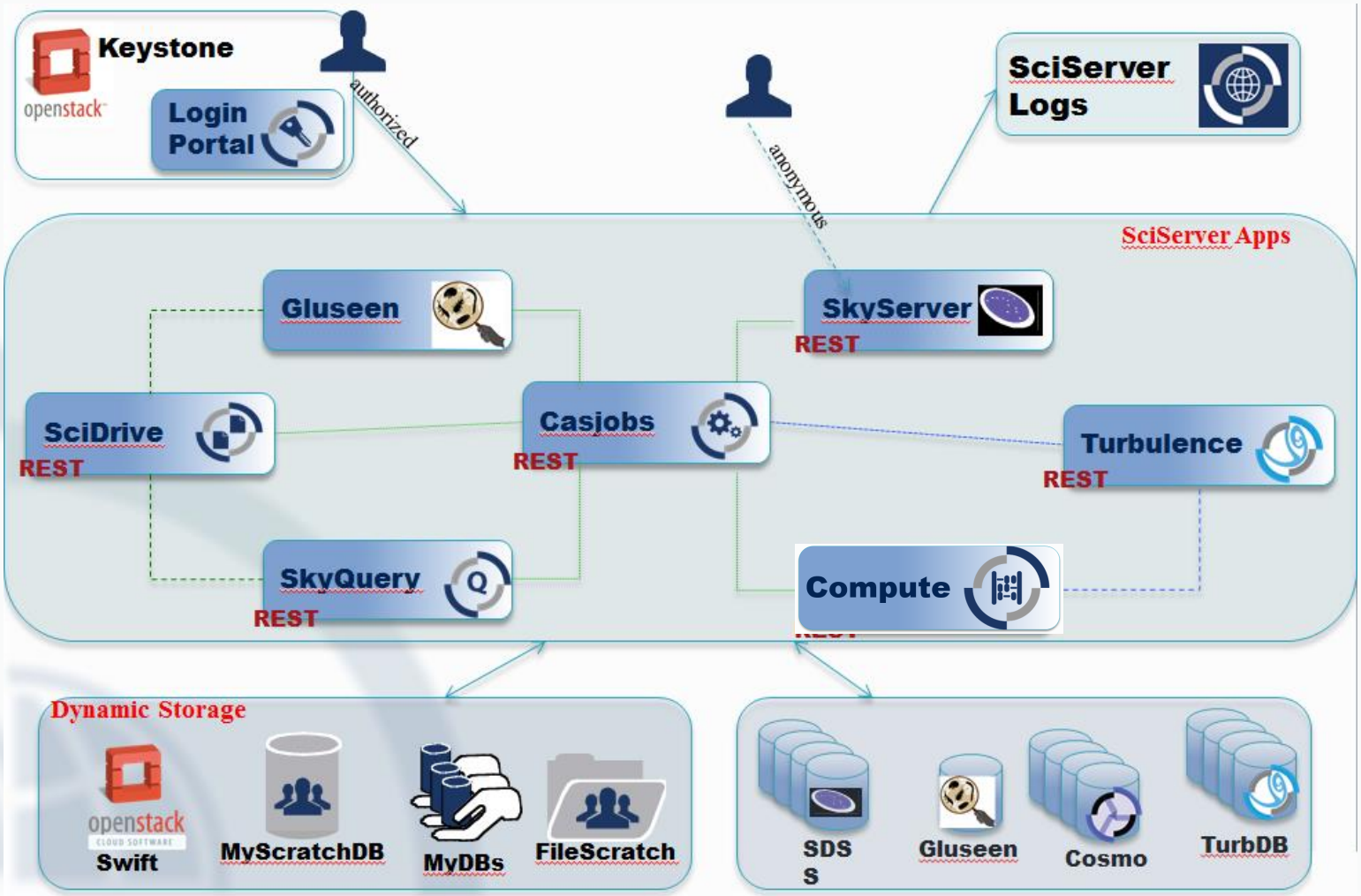
Data exploration: SkyServer

User sign-on: Portal

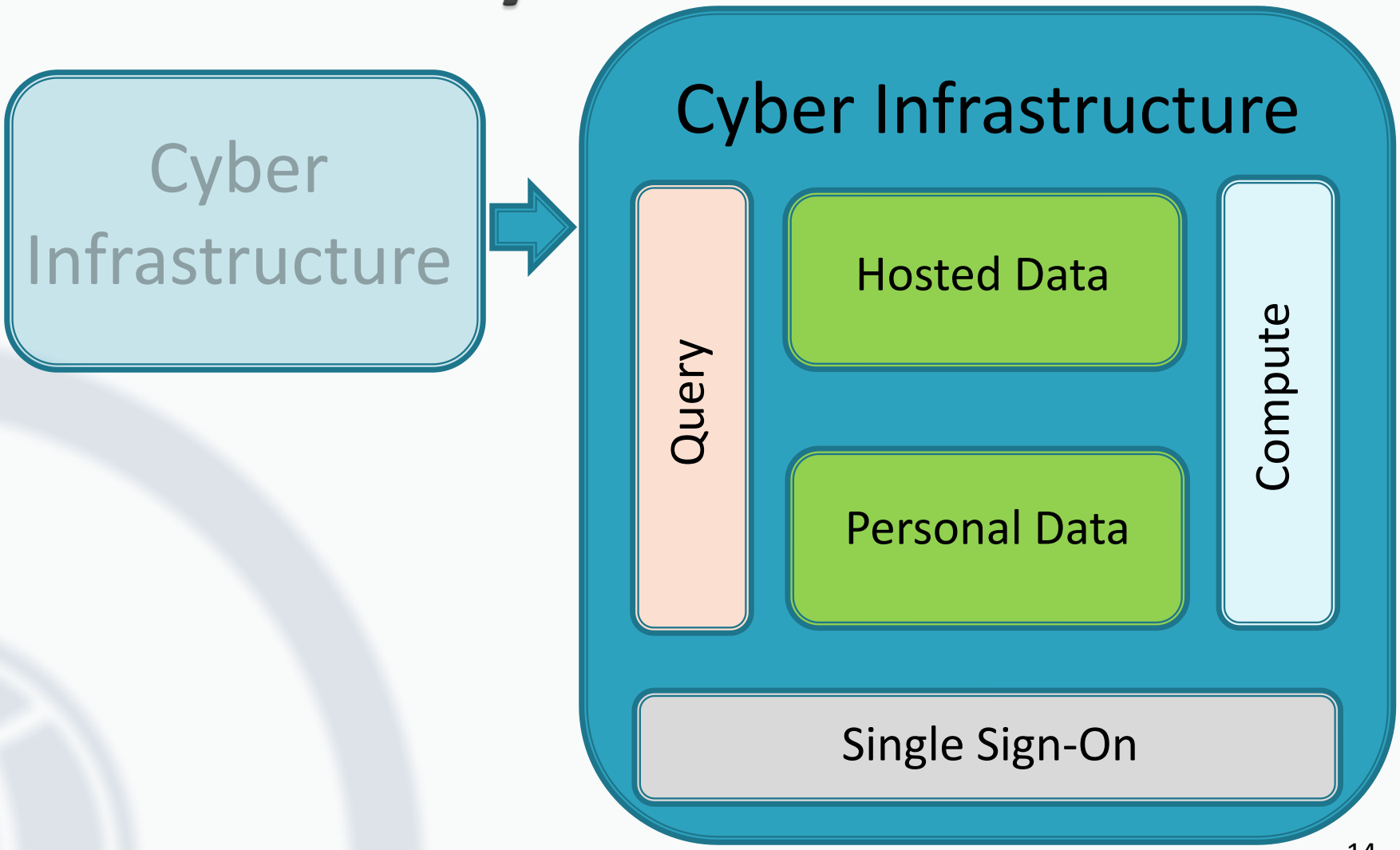
How they fit together : Original



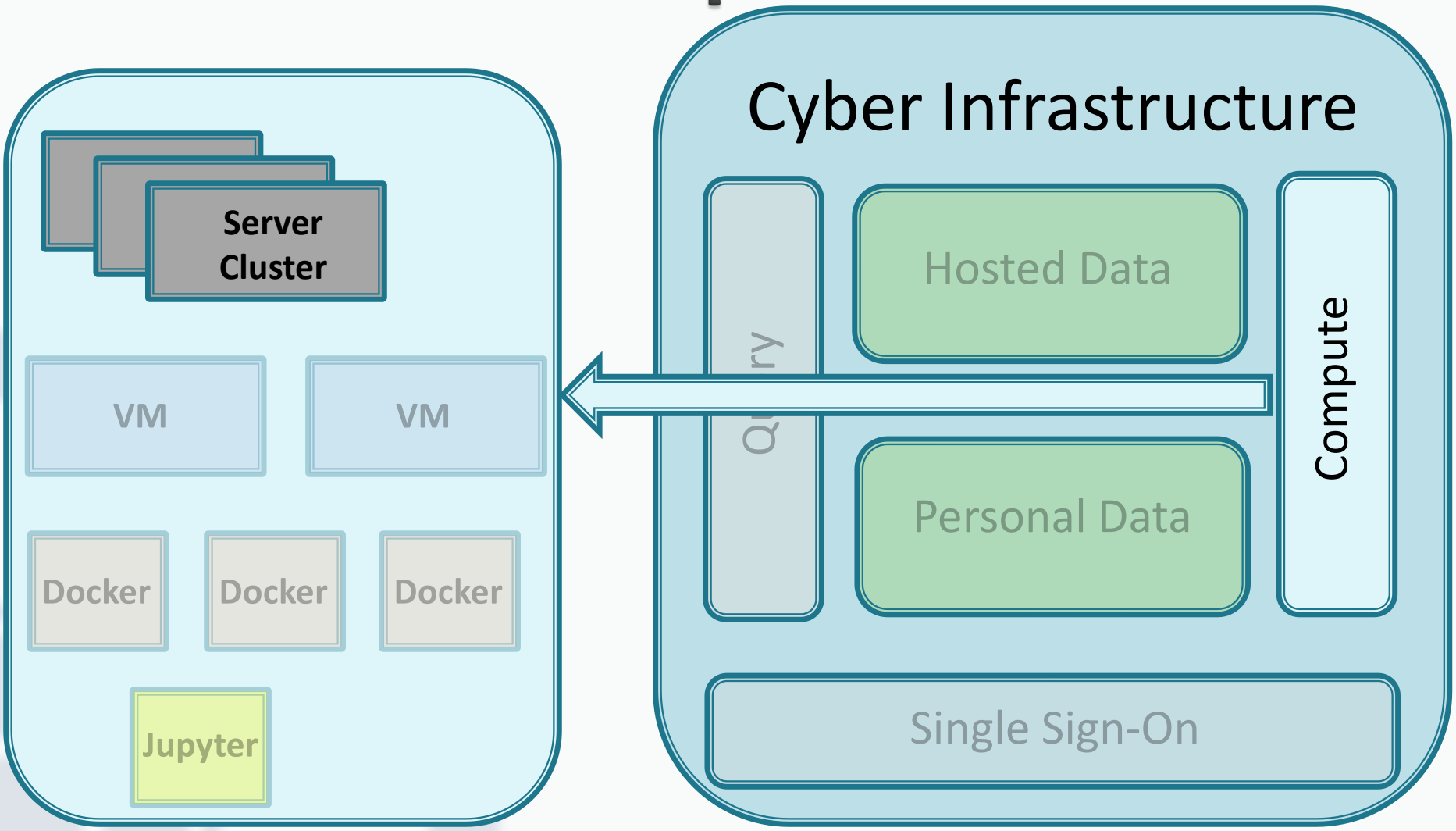
Intermediate Refactoring



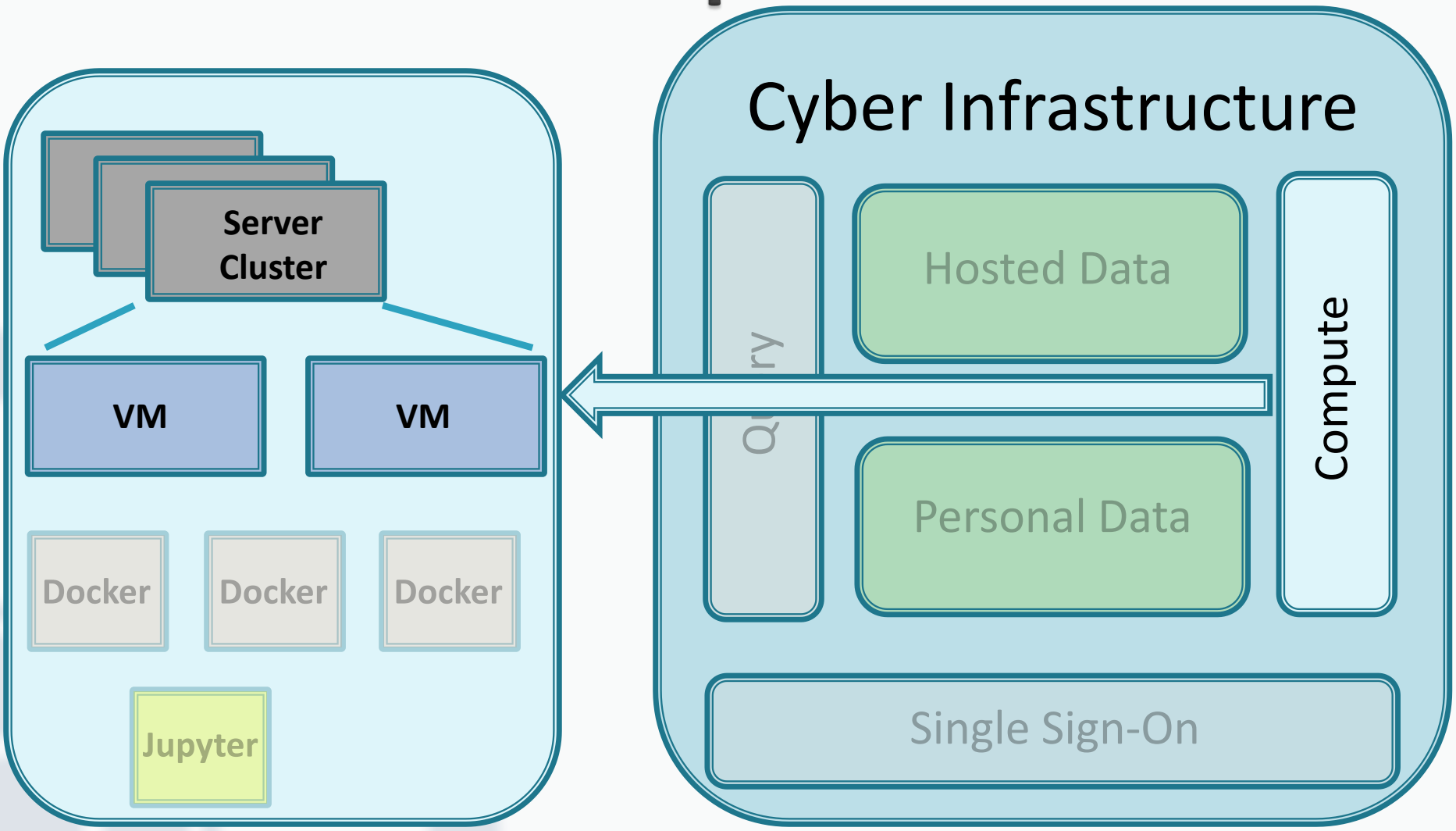
SciServer : CyberInfrastructure



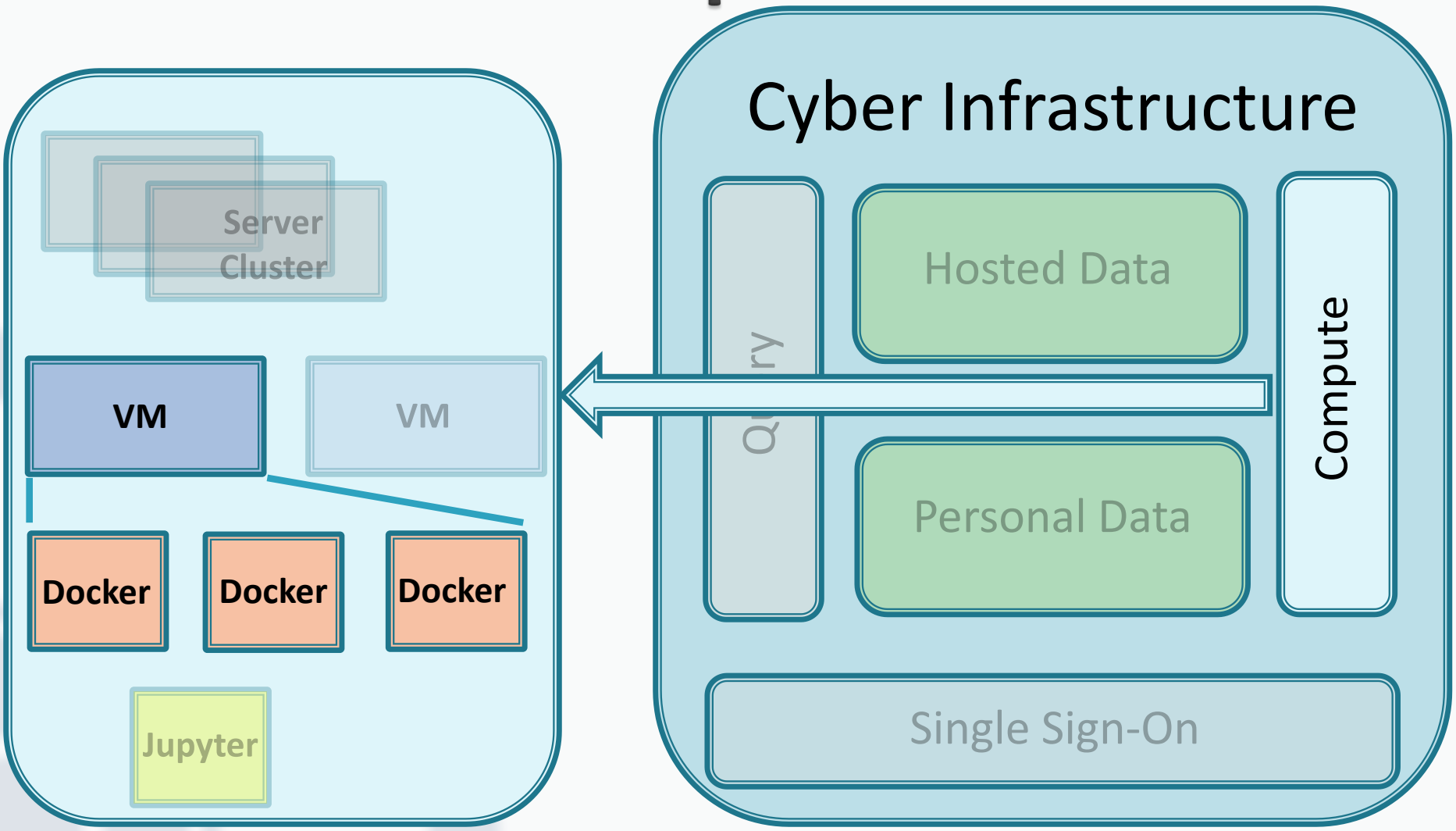
SciServer : Compute



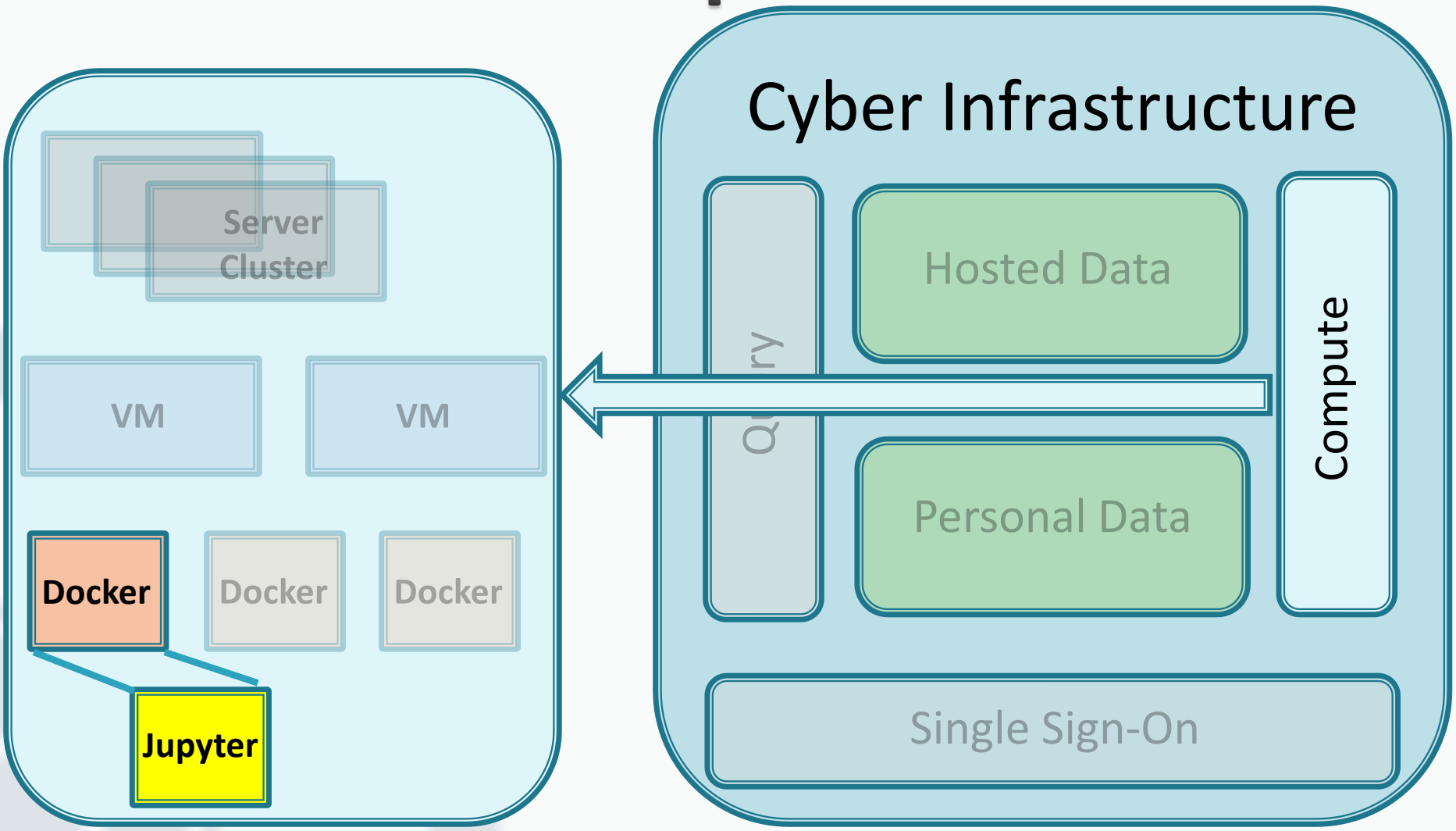
SciServer : Compute



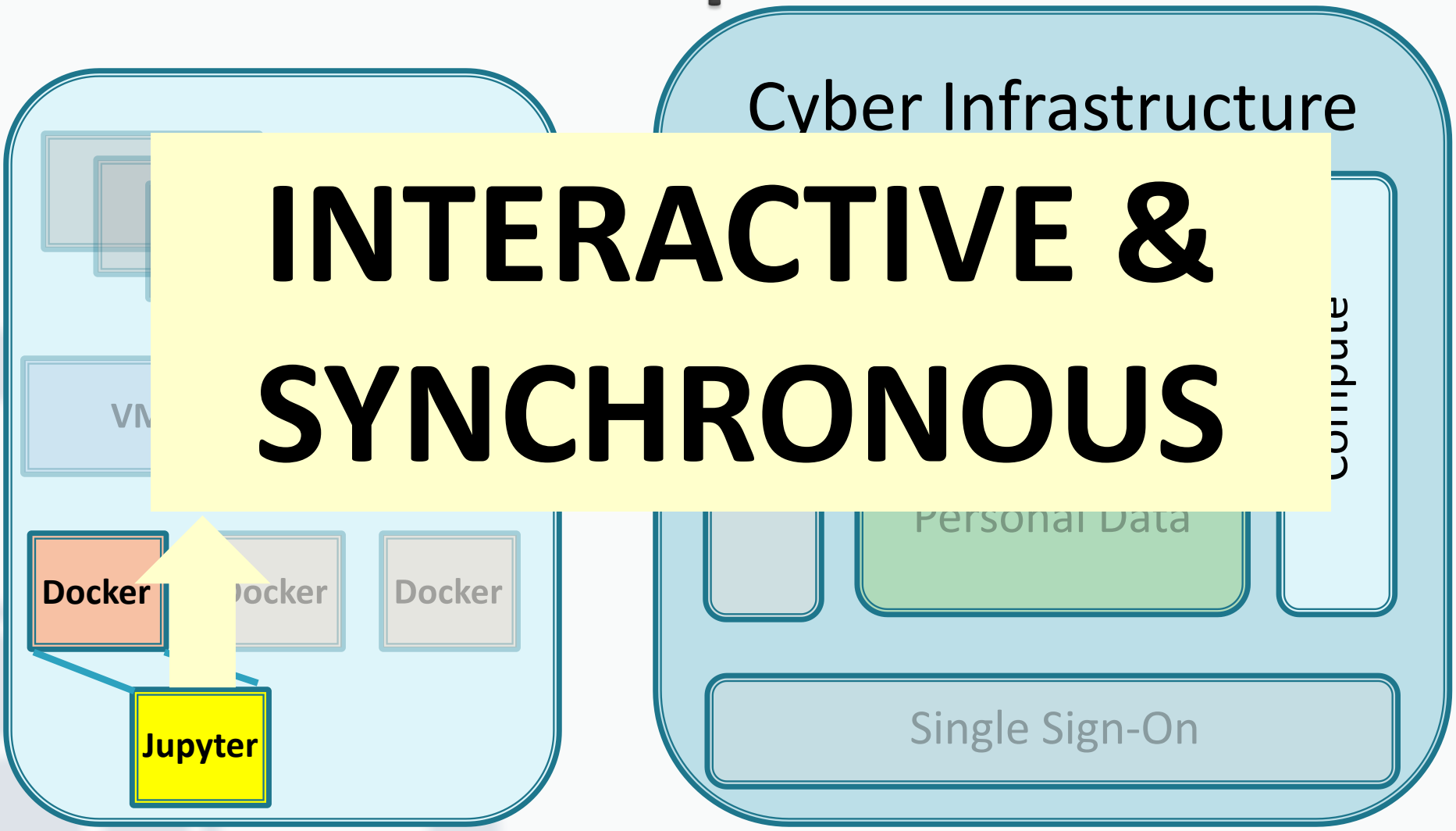
SciServer : Compute



SciServer : Compute



SciServer : Compute

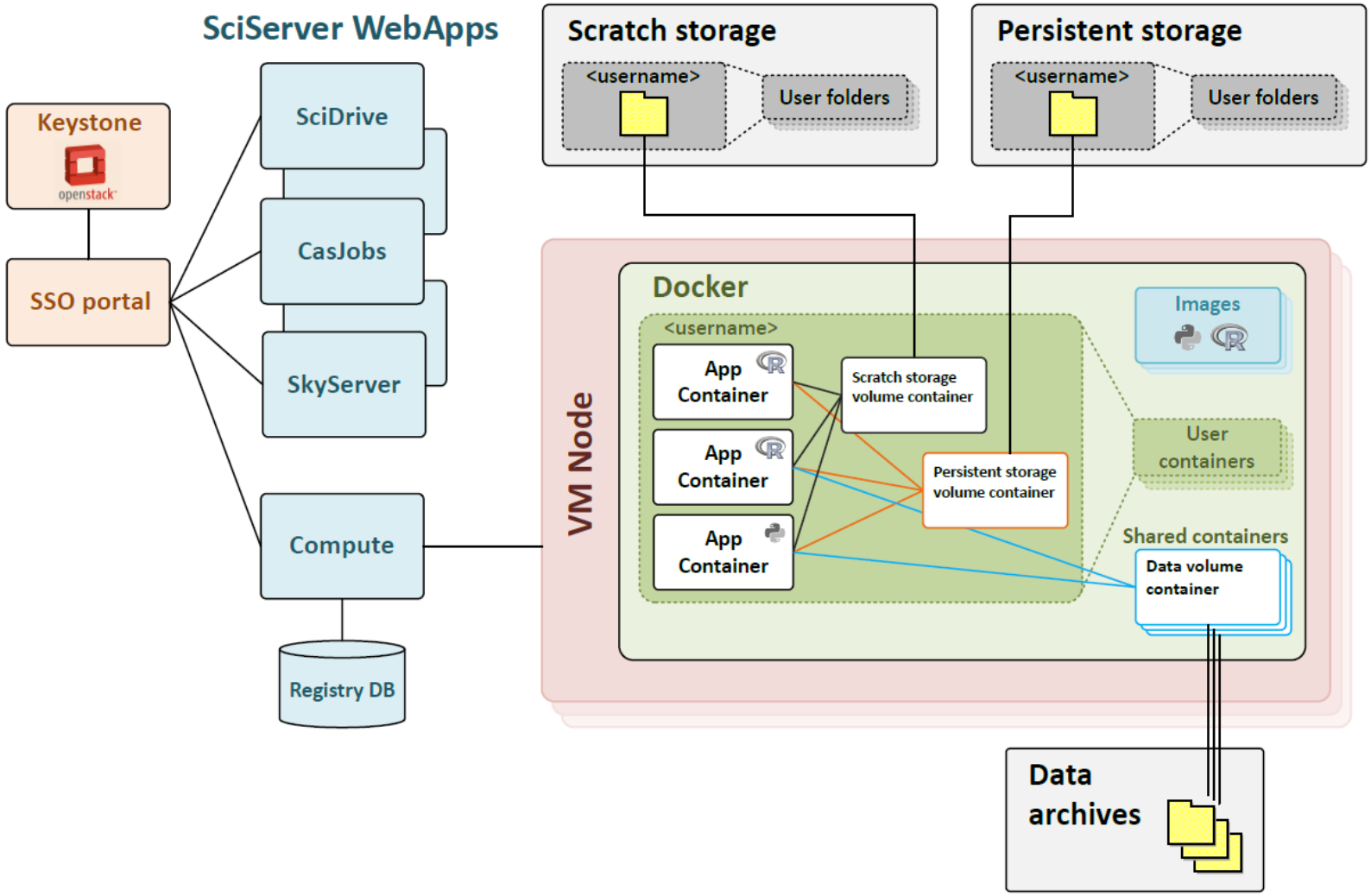


INTERACTIVE & SYNCHRONOUS

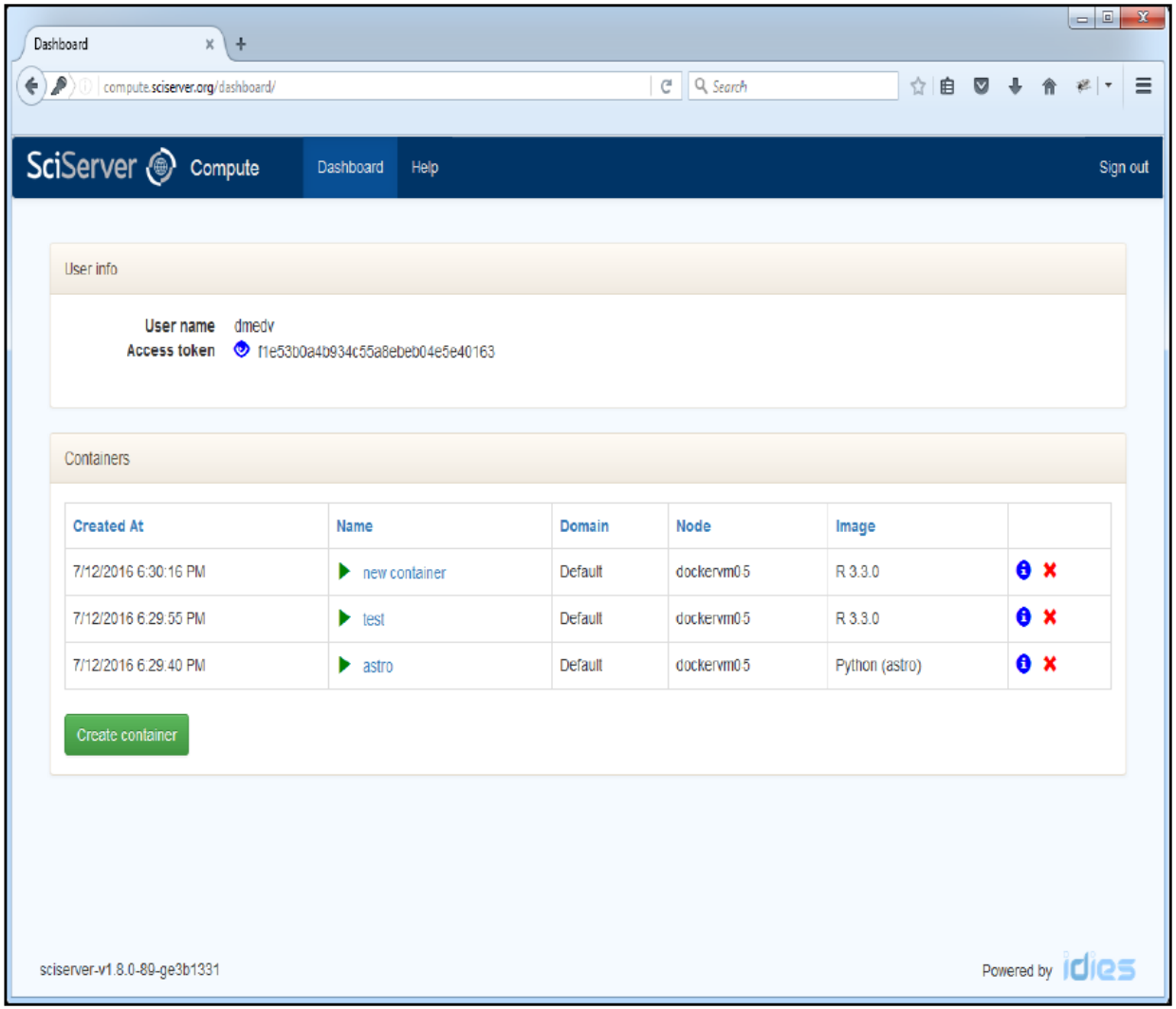
SciServer : Compute

- "Engine" for executing analysis on data sets
- Environment for executing Python Notebooks **Interactively**
- Utility API Libraries in Python and R
- Interacts with ALL other SciServer components that have a WS API:
  - Login Portal for authentication
  - CASJobs for Queries
  - SkyServer and SkyQuery for Astronomy data
  - SciDrive for Storage

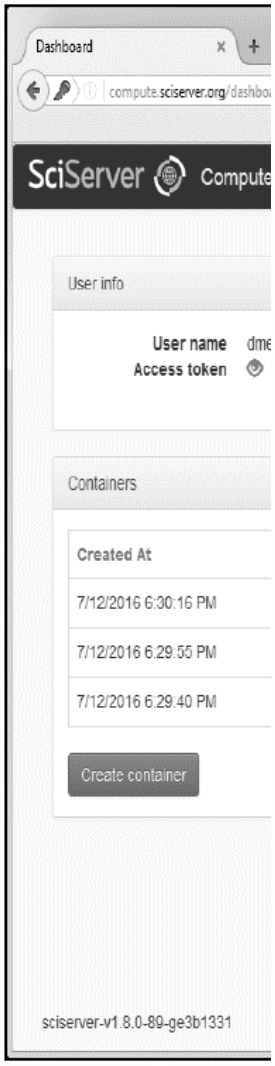
Compute Architecture



Compute Dashboard



Compute : Jupyter Notebook



jupyter 1. SDSS-Thumbnails Last Checkpoint: a minute ago (unsaved changes)

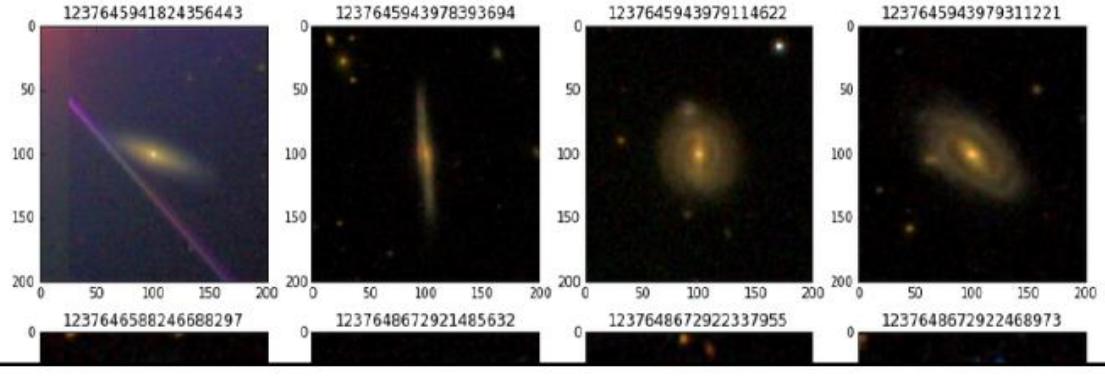
File Edit View Insert Cell Kernel Help Python 3

Markdown CellToolbar

In [71]:
```
width=200
height=200
pixelsize=0.396
plt.figure(figsize=(15, 15))
subPlotNum = 1

for index,gal in gals.iterrows():
    scale=2*gal['petror90_r']/pixelsize/width
    url="http://skyservice.pha.jhu.edu/DR12/ImgCutout/getjpeg.aspx?ra="+str(gal['ra'])
    url+="&dec="+str(gal['dec'])+"&scale="""+str(scale)+"&width="+str(width)
    url+="&height="+str(height)
    img=skimage.io.imread(url)
    plt.subplot(4,4,subPlotNum)
    subPlotNum += 1
    plt.imshow(img)
    # show the object identifier (objId) above the image.
    plt.title(index)
```

Last executed 2016-03-14 20:31:32 in 7.61s



Example : Astronomy (SDSS)

jupyter 2. Stripe82-coadd-Copy1 Last Checkpoint: 3 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Help | Python 3

Code | CellToolbar

```
import SciServer.LoginPortal as Login
token=Login.getToken()
import pandas
import tables
import numpy as np
import astropy
from astropy.io import fits
from astropy import wcs
import skimage.io
import urllib
import os
import matplotlib
import matplotlib.pyplot as plt
```

In [18]:

```
sql="""
SELECT a.objid as head, c.objid2 as match, b.matchcount,
       p.fieldid as head_field, d.fieldid as match_field,
       dbo.fGetUrlFitsCFrame(d.fieldid, 'g') as fits_g,
       dbo.fGetUrlFitsCFrame(d.fieldid, 'r') as fits_r,
       dbo.fGetUrlFitsCFrame(d.fieldid, 'z') as fits_z,
       p.ra, d.ra as match_ra, p.dec, d.dec as match_dec
       , p.petror90_r
  from (select top 1 * from galaxy where objId=8658194378960928809) a
   join matchhead b on a.objid=b.objid        -- join with matchhead
   join photoobj p on a.objid=p.objid         -- get matchhead photoobj
   join match c on c.objid1=b.objid           -- join with all the matches
   join photoobjall d on c.objid2=d.objid     -- get match photoobj
order by d.fieldid                              -- order by matchhead objid
"""
queryResponse = SciServer.CasJobs.executeQuery(sql, "Stripe82",token=token)
obss = pandas.read_csv(queryResponse,index_col=None)
obss[:10]
```

Out[18]:

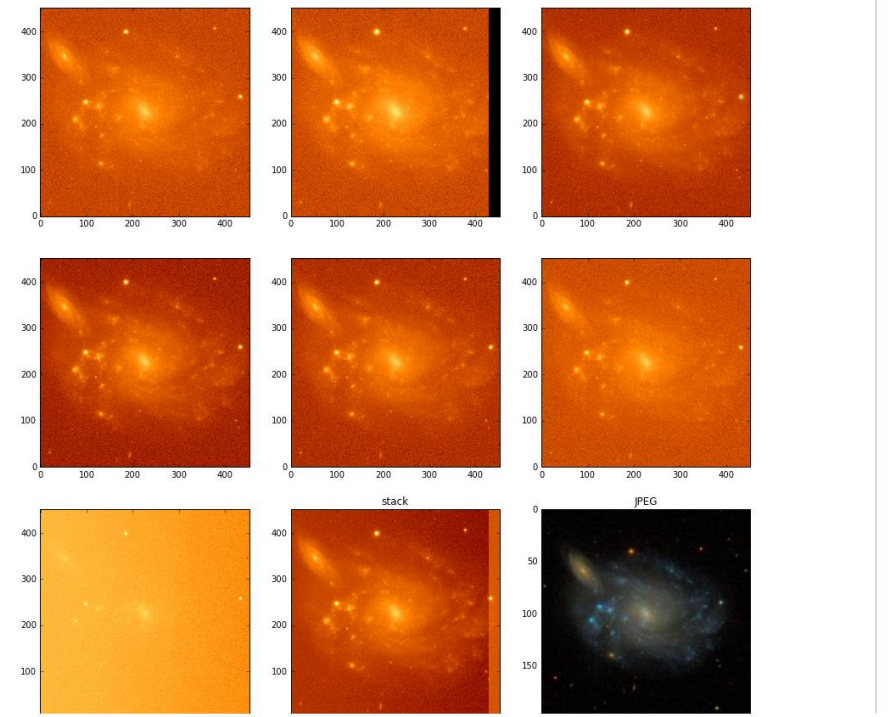
	head	match	matchcount	head_field	match_field	fits_g
0	8658194378960928809	8658194430499553320	57	8658194378960928768	8658194430499553280	http://das.sdss.org/imaging/5622/40/corr/
1	8658194378960928809	8658194477742948377	57	8658194378960928768	8658194477742948352	http://das.sdss.org/imaging/5633/40/corr/
2	8658194378960928809	8658194516375371821	57	8658194378960928768	8658194516375371776	http://das.sdss.org/imaging/5642/40/corr/
3	8658194378960928809	8658194585083510793	57	8658194378960928768	8658194585083510784	http://das.sdss.org/imaging/5658/40/corr/
4	8658194378960928809	8658194804163018771	57	8658194378960928768	8658194804163018752	http://das.sdss.org/imaging/5709/40/corr/
5	8658194378960928809	8658194954470752297	57	8658194378960928768	8658194954470752256	http://das.sdss.org/imaging/5744/40/corr/
6	8658194378960928809	8658195018907910161	57	8658194378960928768	8658195018907910144	http://das.sdss.org/imaging/5759/40/corr/
7	8658194378960928809	8658195044651040803	57	8658194378960928768	8658195044651040768	http://das.sdss.org/imaging/5765/40/corr/
8	8658194378960928809	8658195066151239724	57	8658194378960928768	8658195066151239680	http://das.sdss.org/imaging/5770/40/corr/
9	8658194378960928809	8658195113395748890	57	8658194378960928768	8658195113395748864	http://das.sdss.org/imaging/5781/40/corr/

jupyter 2. Stripe82-coadd-Copy1 Last Checkpoint: a minute ago (autosaved)

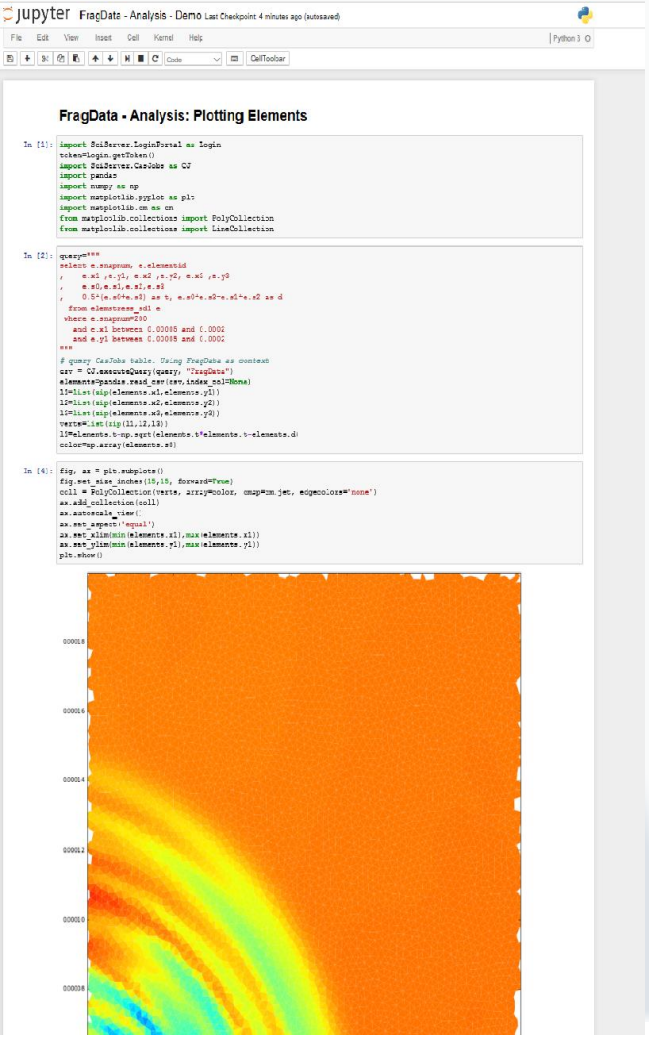
File Edit View Insert Cell Kernel Help | Python 3

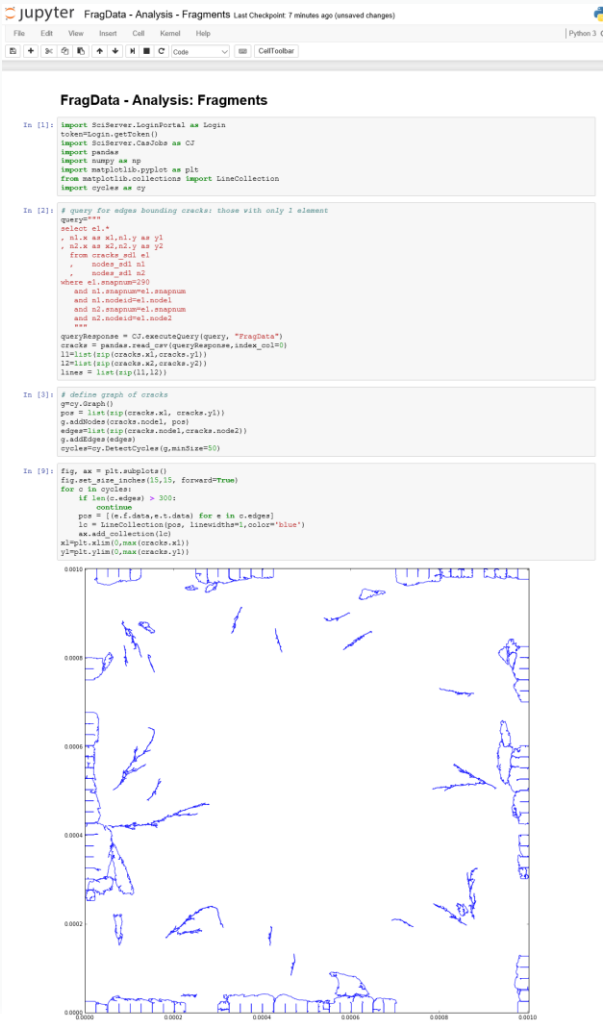
Code | CellToolbar

Out[17]: <matplotlib.text.Text at 0x7f2aa174a6d8>



Example : Material Science





Example : Cosmology

jupyter CosmoUC_5_PlotHalo Last Checkpoint: 11 minutes ago (autosaved)

File Edit View Insert Cell Kernel Help Notebook saved Python 3

Code CellToolbar

Dark-matter Halos in a Cosmological Simulation

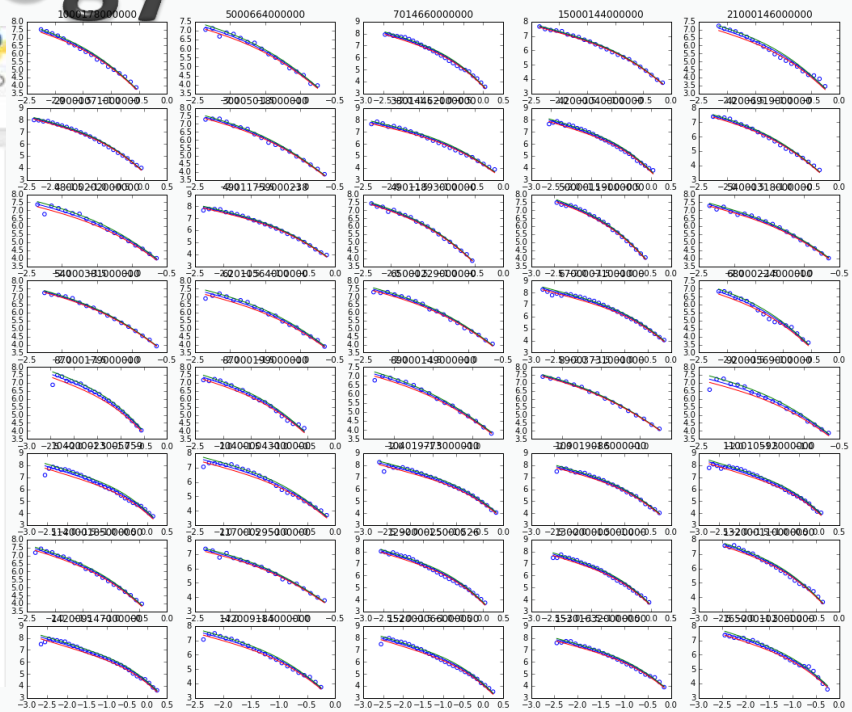
In [3]:
```
import SciServer.LoginPortal as Login
token = Login.getToken()
import SciServer.CasJobs
import pandas
import tables
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

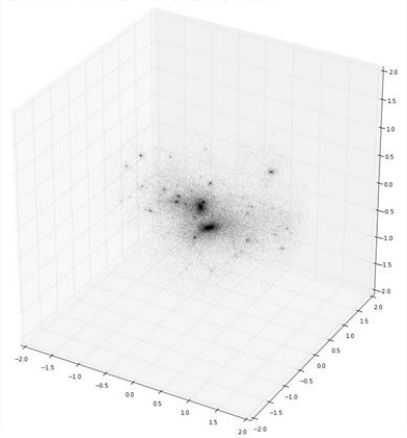
In [10]:
```
%%time
queryString = """
select top 100000 p.x-hh.x as x,p.y-hh.y as y,p.z-hh.z as z
from mpahalotrees.mr hh
   cross apply dbo.MillenniumParticles(hh.snapnum,
      dbo.Sphere::New(hh.x,hh.y,hh.z,3*hh.halfmassradius).ToString()) p
  where hh.haloid=84000007000000 order by newid()
    """
responseStream = SciServer.CasJobs.executeQuery(queryString, token=token,context="SimulationDB")
df = pandas.read_csv(responseStream, index_col=None)
```

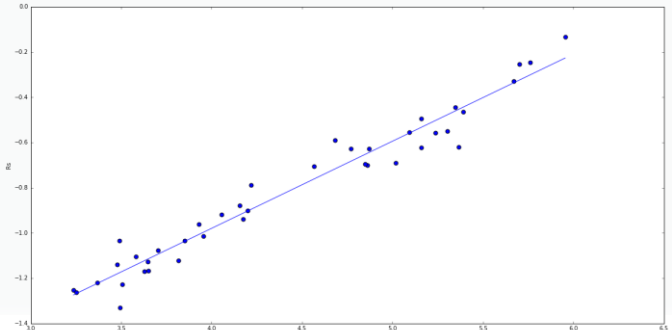
```
CPU times: user 351 ms, sys: 184 ms, total: 535 ms
Wall time: 5.27 s
```

In [13]:
```
fig = plt.figure(figsize=(15, 15))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(df.x,df.y, df.z,s=0.001)
```

Out[13]:
```
<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x7fe86428b9e8>
```







SciServer

Collaborative data-driven science

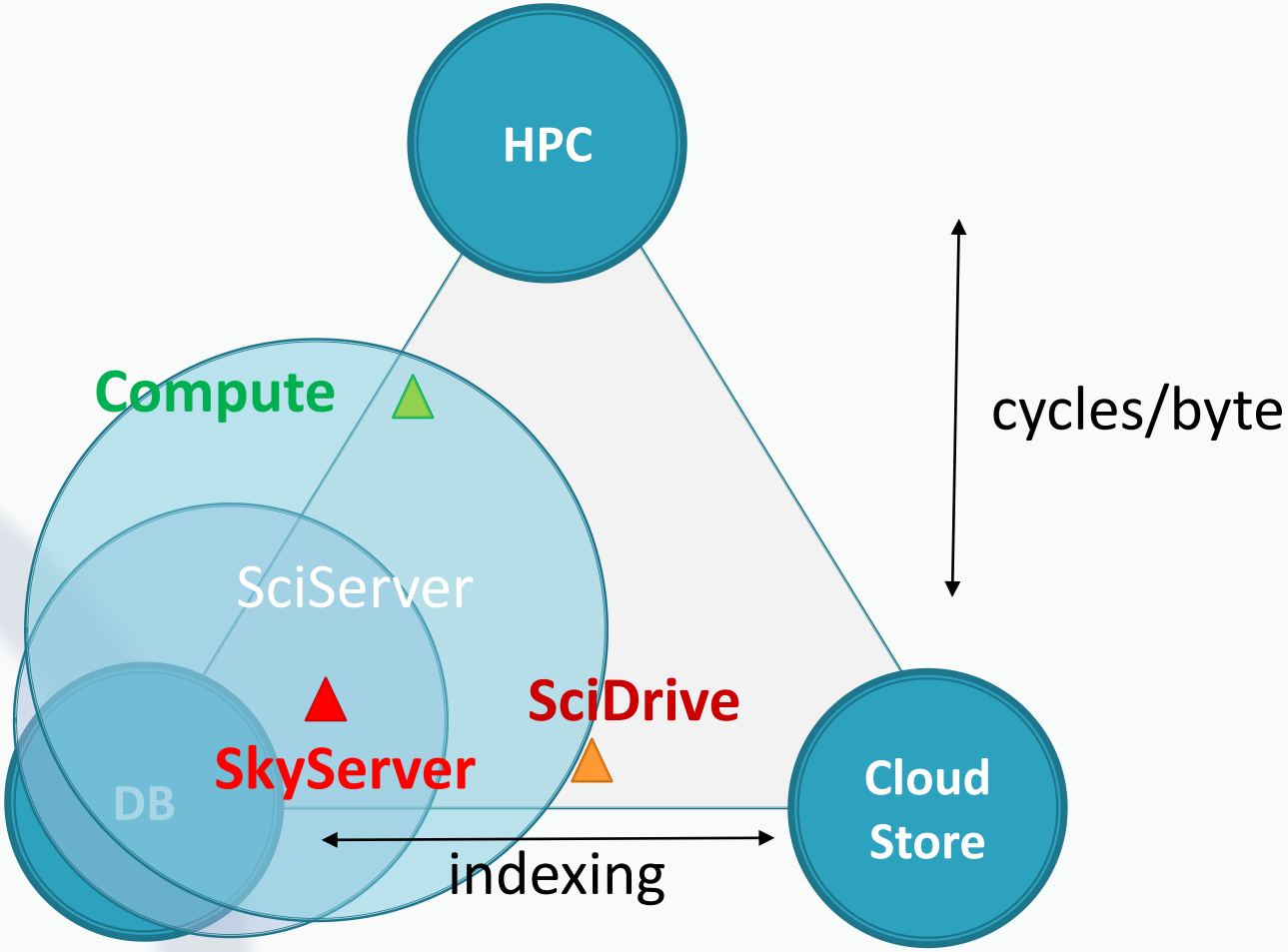
Compute Next Stage

JOHNS HOPKINS UNIVERSITY

NSF

idies

System Balance



Compute Development

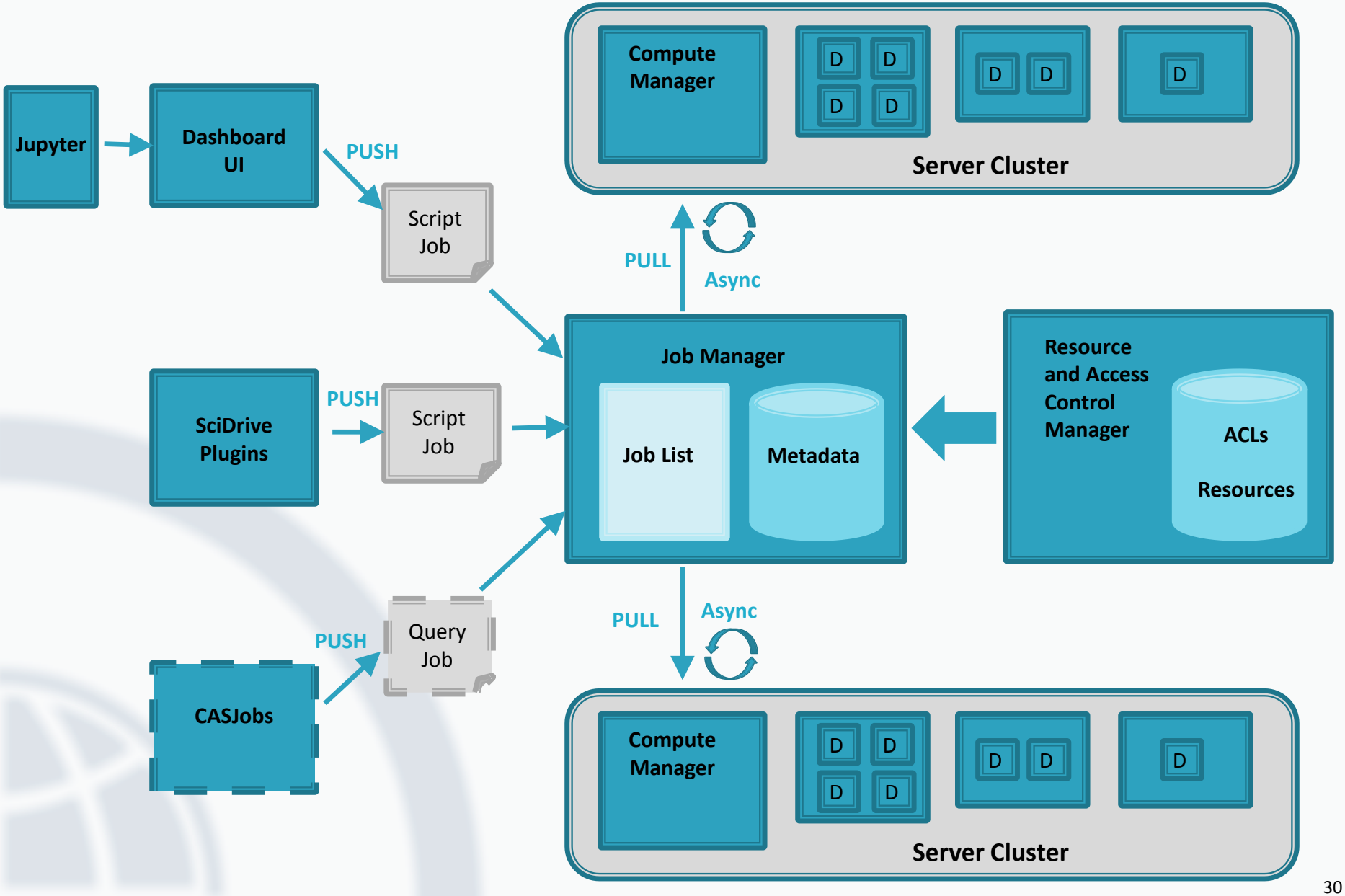
- **Build on VM/Docker Architecture**

- **Scalable non-interactive, asynchronous Job management (JOBM)**
- **Rich Access Controls (RACM)**
- **Distributed compute execution (COMPM)**

- **Support Python, R, Matlab**

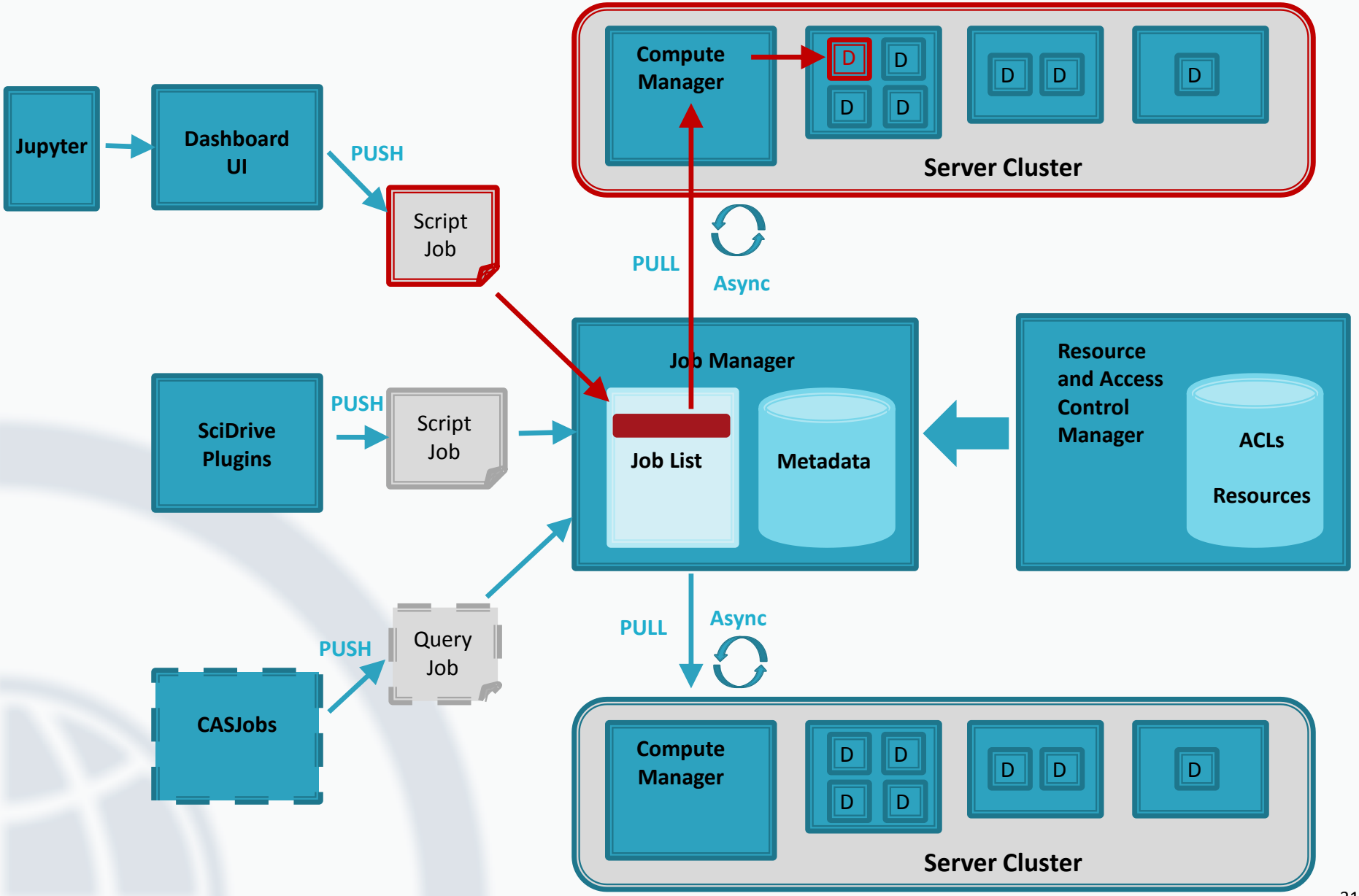
Collaborative data-driven science

SciServer



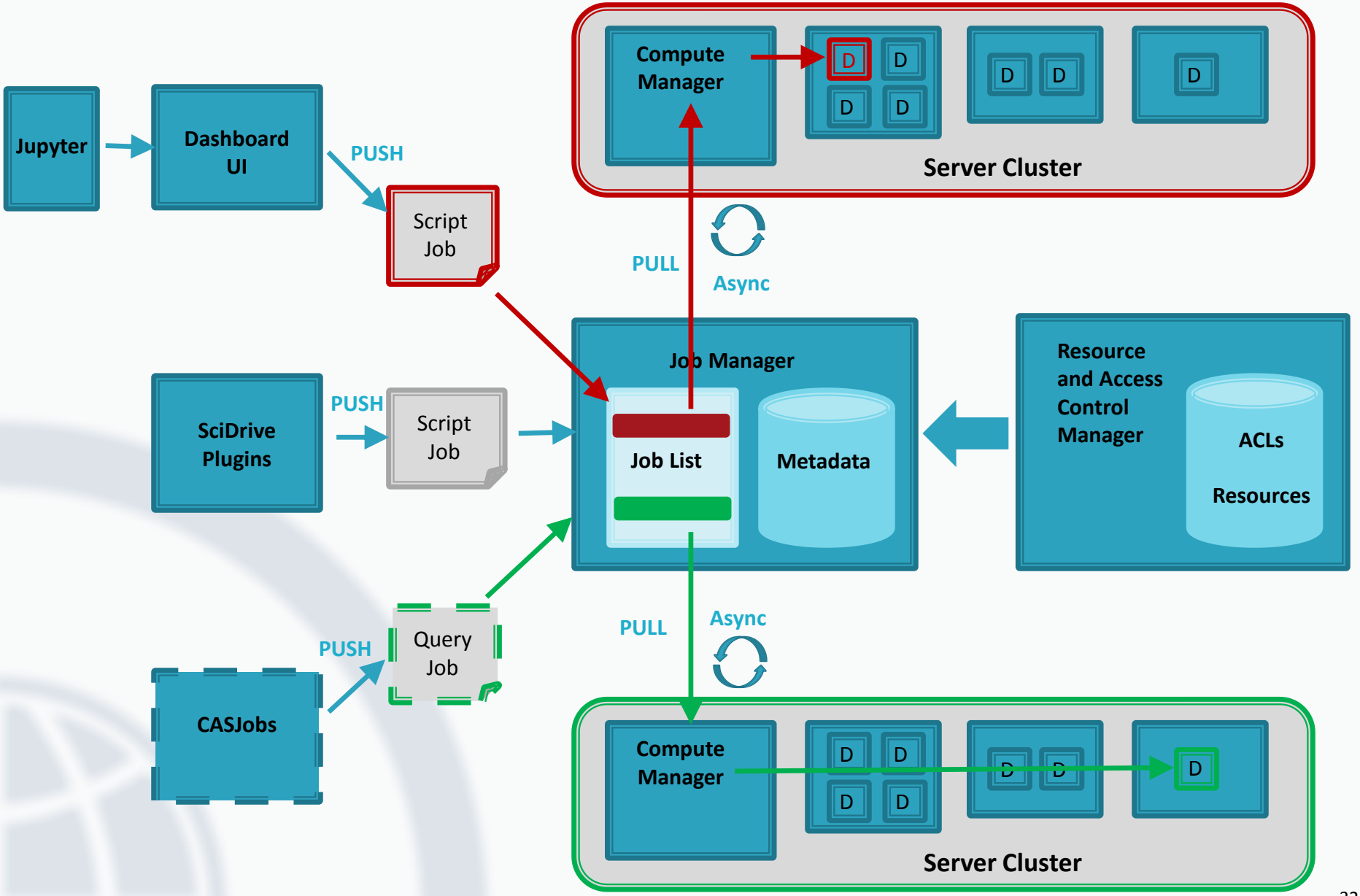
Collaborative data-driven science

SciServer



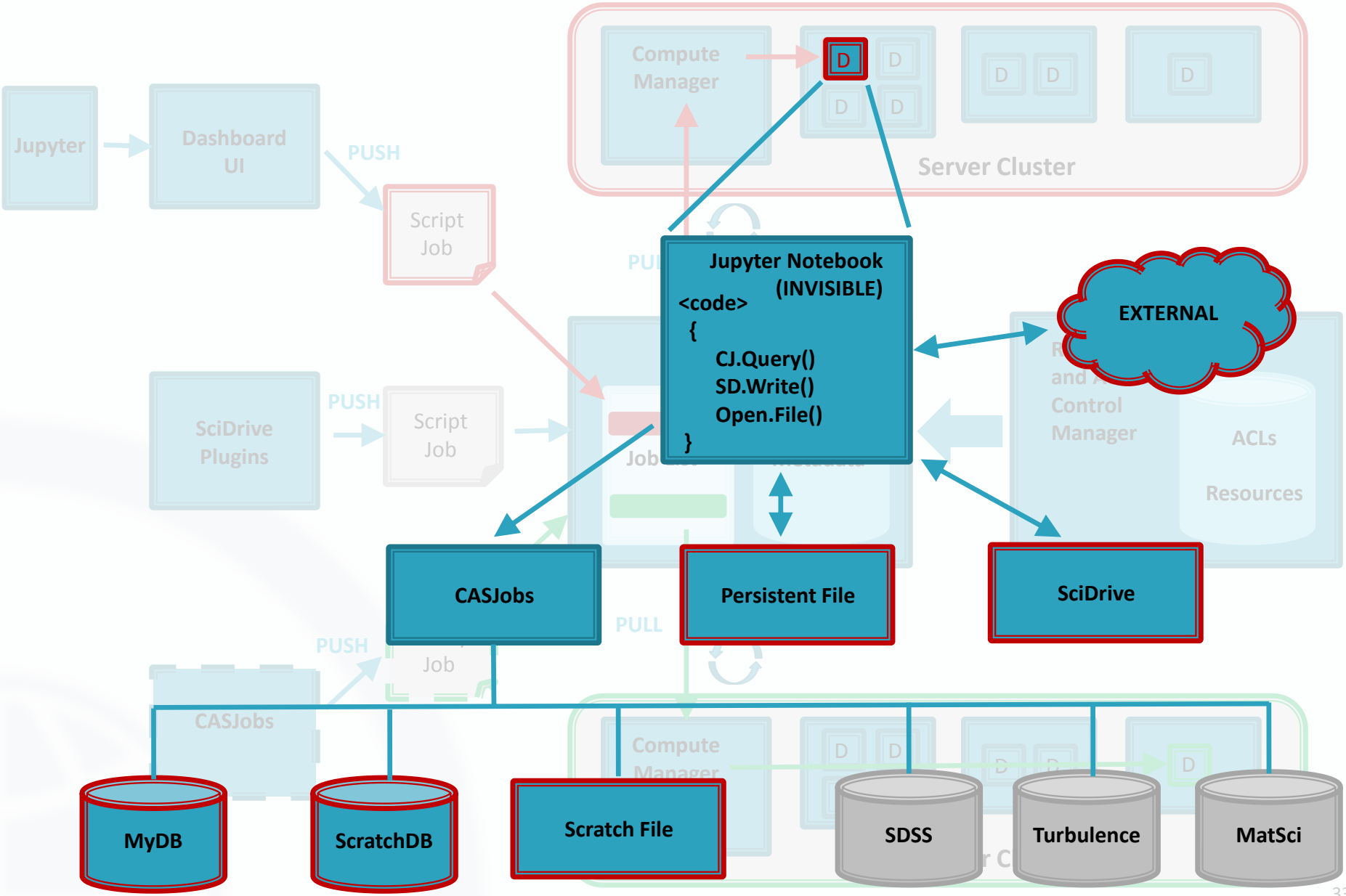
Collaborative data-driven science

SciServer



Collaborative data-driven science

SciServer



Summary

- SciServer Compute Interactive is live now

- Supports Python, R, Jupyter
- Runs on a 4 node cluster
- Access to several domain databases

- Asynchronous Job Execution early 2017

- Please register with SciServer and try it out

SciServer

Collaborative data-driven science

Questions?

JOHNS HOPKINS UNIVERSITY

NSF

idies