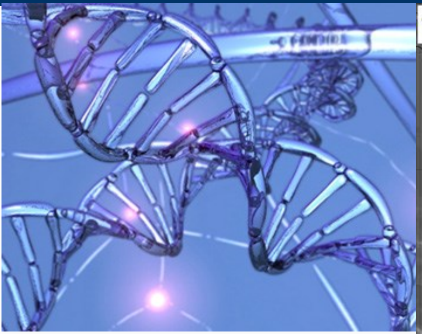
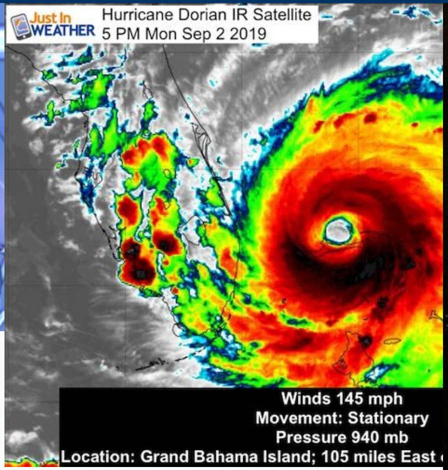
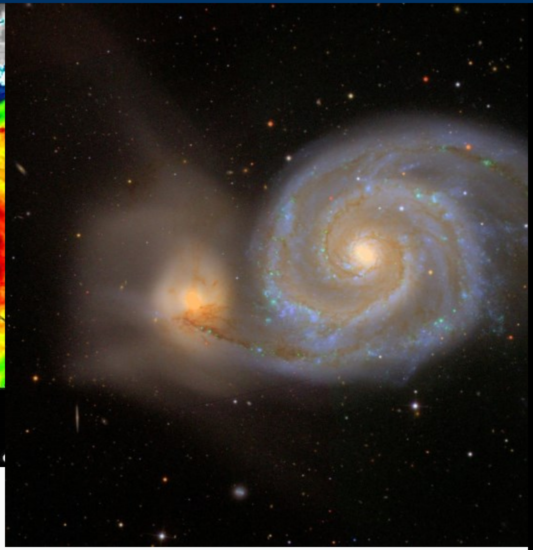
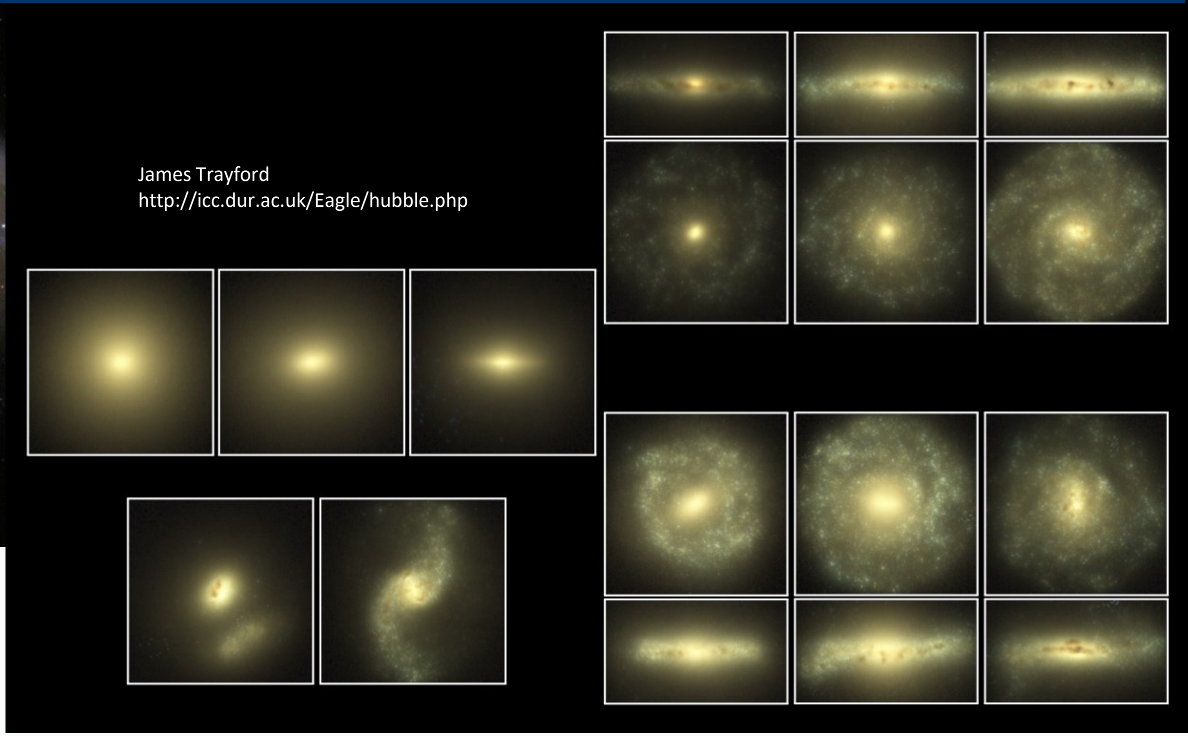
Collaborative data-driven science

SciServer








# Cosmological Simulations on SciServer

Gerard Lemson
Chair Theory Interest Group IVOA
IDIES, The Johns Hopkins University

idies

# Why theory interest group?

- Theory needs special attention
  - whitepaper with Joerg Colberg 2004, https://wiki.ivoa.net/internal/IVOA/IvoaTheory/Theory_in_the_VO_whitepaper.pdf
- Hard to standardize simulations
  - Observational standards do not apply
- Heterogeneity of data products
  - not just photons at time T from direction V with energy E and polarization P
- No common sky
  - what to query?
- No common objects
  - what to compare, cross-match?

# Theory Interest Group Charter

The IVOA Theory Interest Group will:

- Provide a forum for discussing theory specific issues in a VO context.
- Contribute to other IVOA working groups to ensure that theory specific requirements are included.
- Incorporate standard approaches defined in these groups when designing and implementing services on theoretical archives.
- Define standard services relevant for theoretical archives.
- Promote development of services for comparing theoretical results to observations and vice versa.
- Define relevant milestones and assign specific tasks to interested parties.

https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaTheory

# Theory Interest Group

- Represents theorists to IVOA
- Represents IVOA to theorists
- 2 standards:
  - Simulation Data Model (SimDM):
    - Kinda registry for simulation products
  - Simulation Data Access Layer (SimDAL)
    - https://ivoa.net/documents/SimDAL/20170320/index.html
    - Discovery and retrieval of simulation products
- I think fair to say not much take up

Babylonian confusion

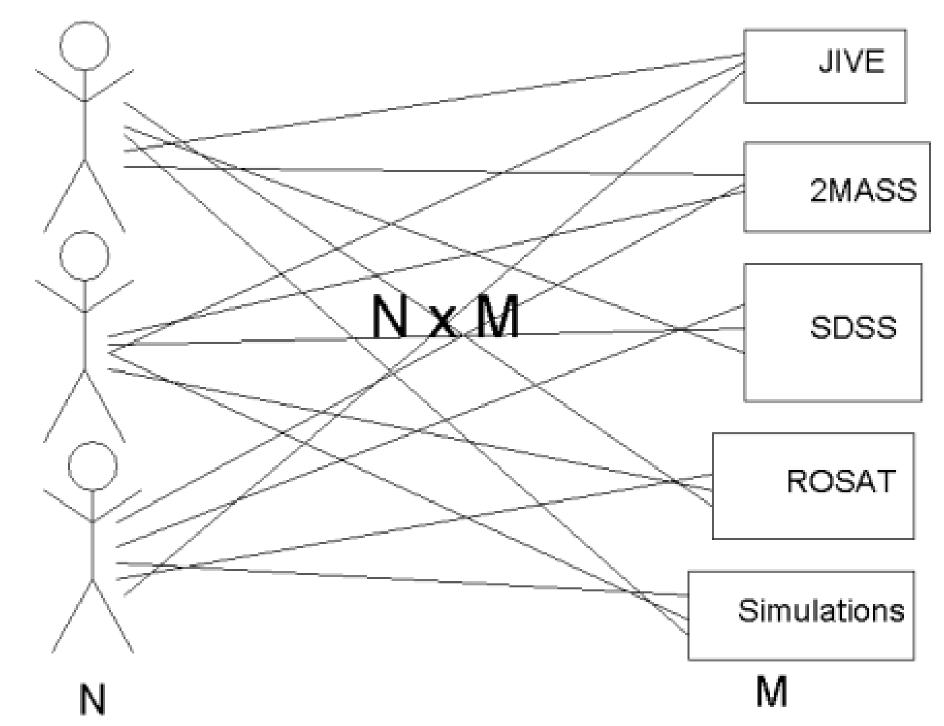



Figure 1: The problem

"Esperanto"

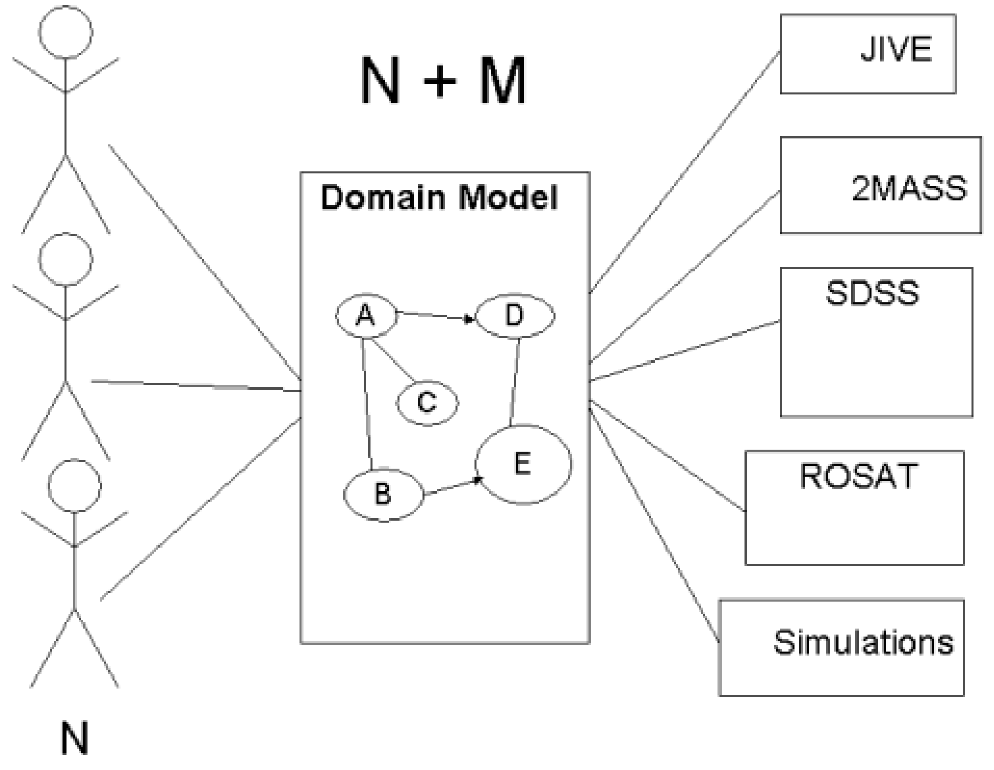


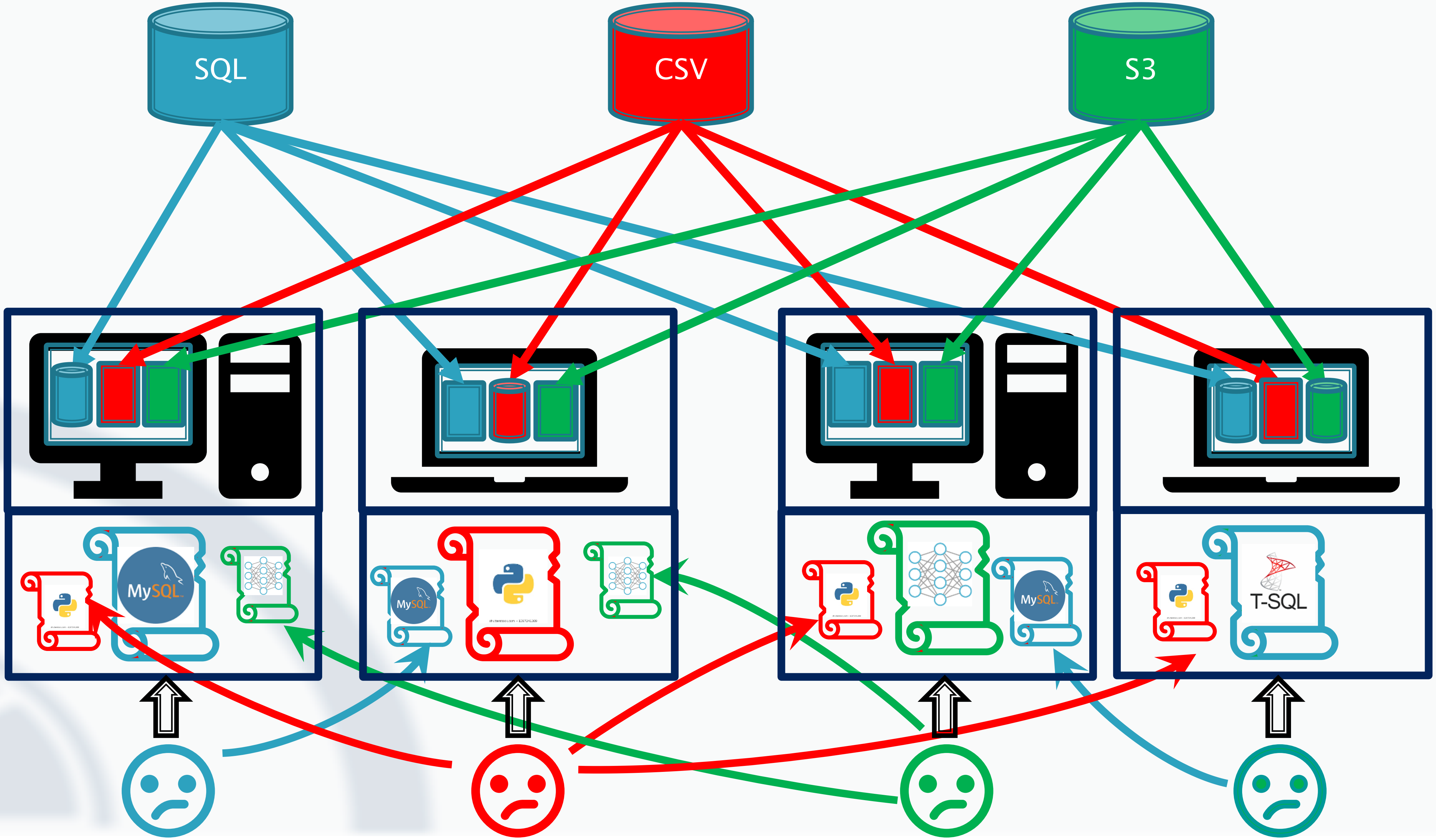

Figure 2: The solution

*Theory in the VO* (Lemson & Colberg, 2004)

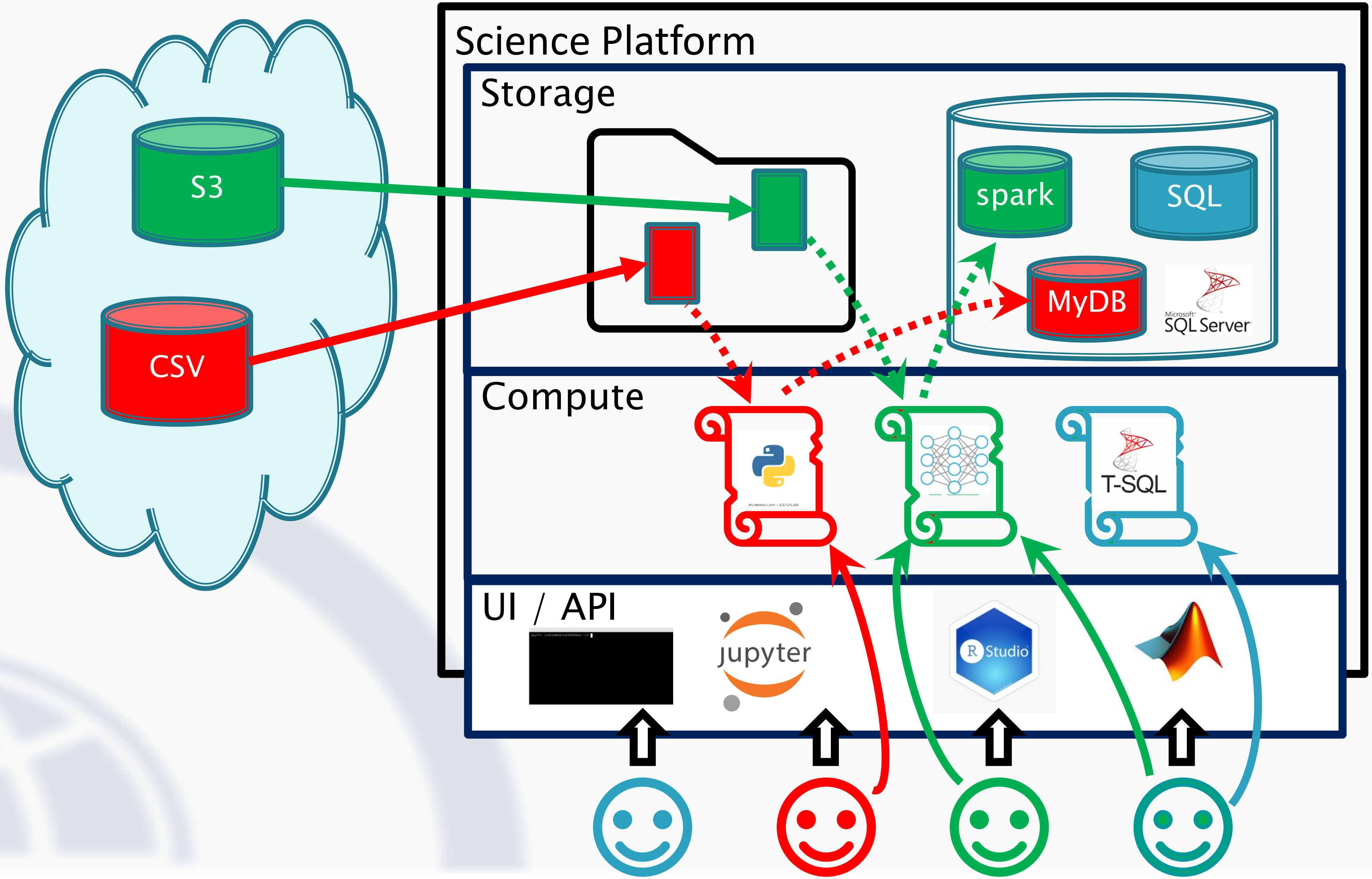
https://wiki.ivoa.net/internal/IVOA/IvoaTheory/Theory_in_the_VO_whitepaper.pdf

# This is not trivial, especially for simulations

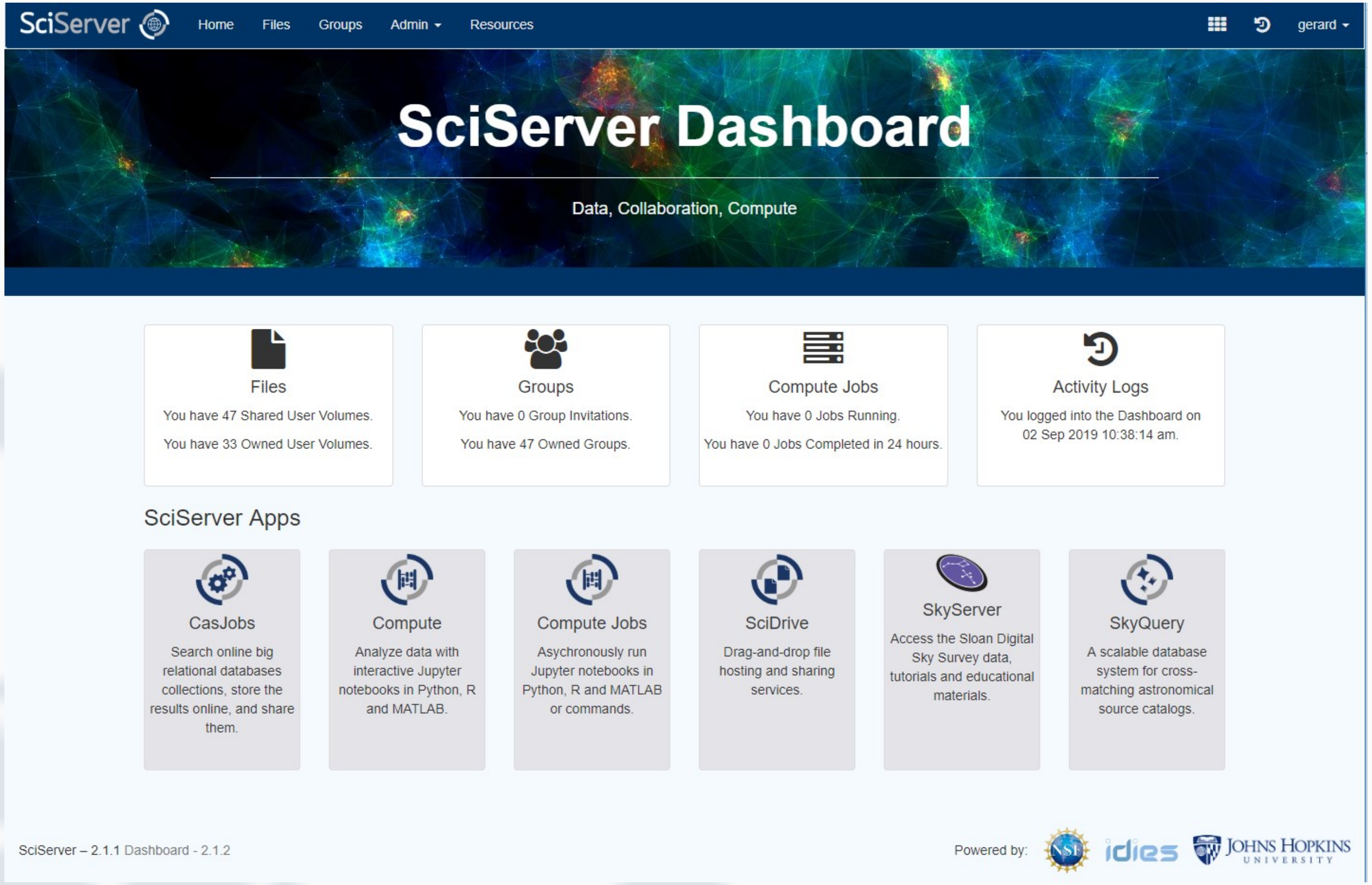
| | |
|---|---|
| Data growing exponentially | Expertise required to optimize big data analysis; downloads unfeasible |
| Science increasingly collaborative | Data and analysis sharing often ad hoc |
| Simulation data sets much more heterogeneous | Diverse set of skills and knowledge required to interpret each of these |
| Analysis methods ever more sophisticated | Increasing desire to apply latest ML techniques, requiring more sophisticated computational resources |

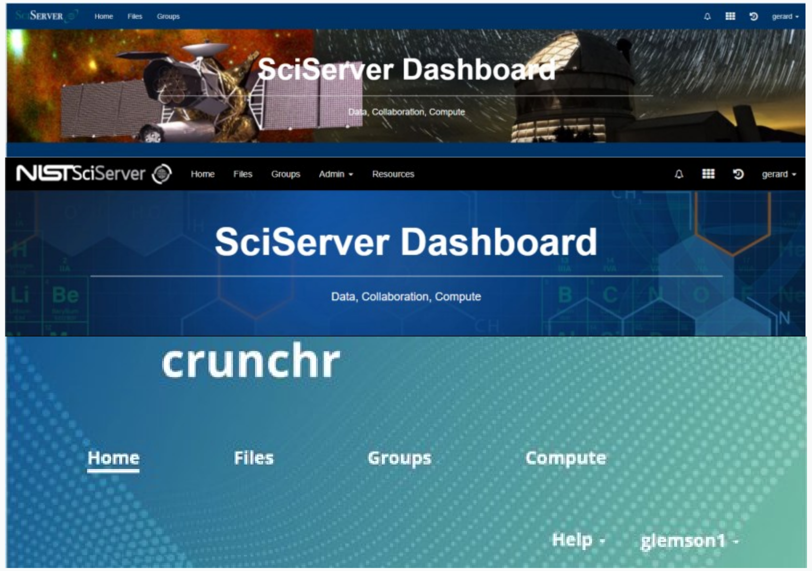

SQL
CSV
S3
MySQL
MySQL
MySQL
T-SQL

SciServer


Science Platform
Storage
S3
CSV
spark
SQL
MyDB
Microsoft SQL Server
Compute
T-SQL
UI / API
jupyter
R Studio

# Our solution: the SciServer Science Platform

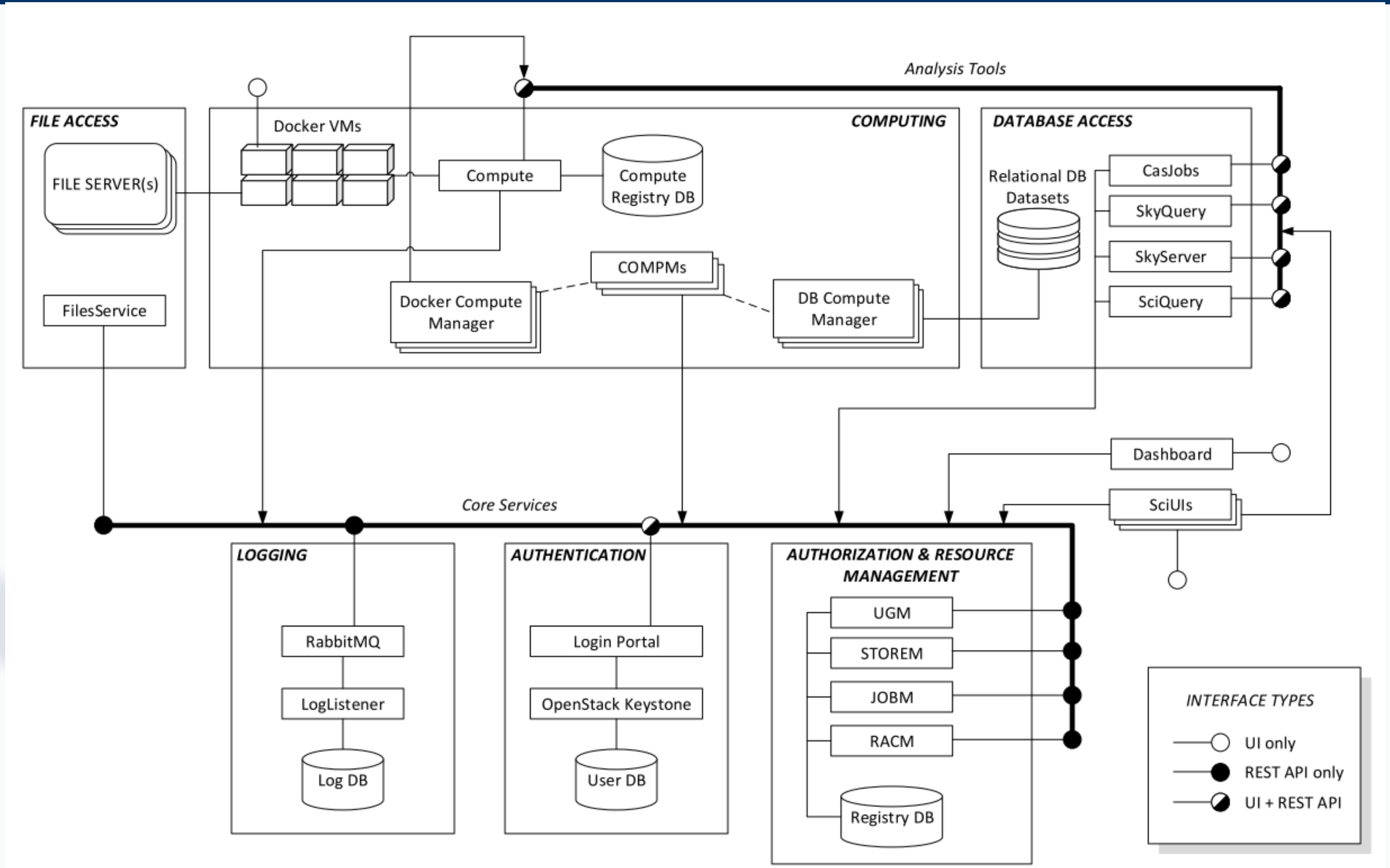


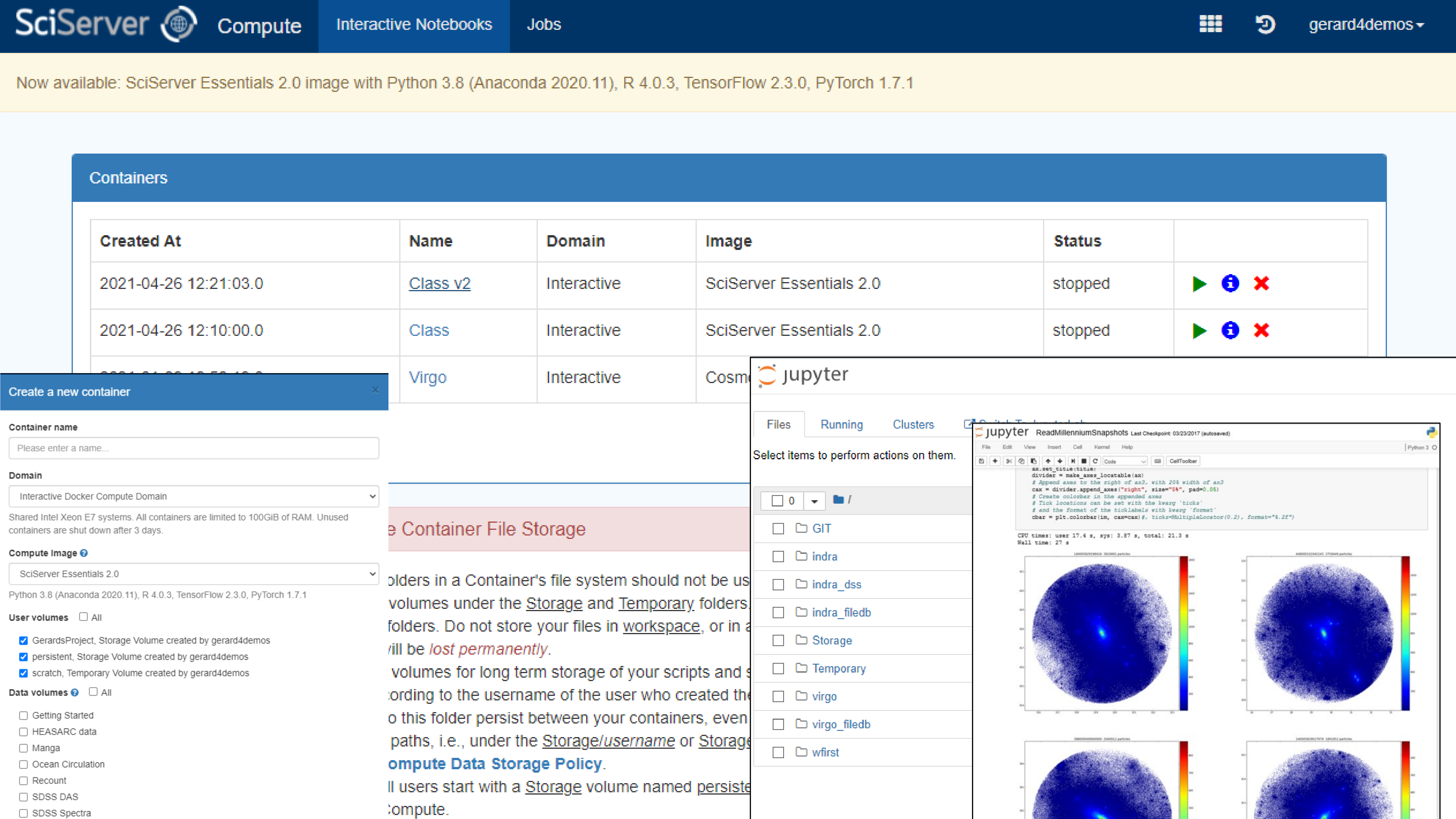

Also, using K8S, at
MPE (eROISTA)
NAOJ (PFS)
NIST
JHMI PMAP (crunchr)

http://www.sciserver.org, https://apps.sciserver.org

SciServer




Taghizadeh-Popp etal, *SciServer: A science platform for astronomy and beyond*
Astronomy and Computing, Volume 33, article id. 100412
See also:
https://www.stsci.edu/contents/newsletters/2018-volume-35-issue-01/science-platformsserver-side-analytics
https://baas.aas.org/pub/2020n7i146/release/1

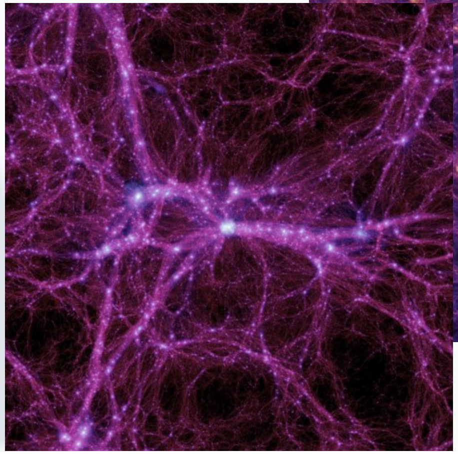

SciServer
Compute
Interactive Notebooks
Jobs
gerard4demos
Now available: SciServer Essentials 2.0 image with Python 3.8 (Anaconda 2020.11), R 4.0.3, TensorFlow 2.3.0, PyTorch 1.7.1
Containers
Created At
Name
Domain
Image
Status
2021-04-26 12:21:03.0
Class v2
Interactive
SciServer Essentials 2.0
stopped
2021-04-26 12:10:00.0
Class
Interactive
SciServer Essentials 2.0
stopped
Virgo
Interactive
Create a new container
Container name
Please enter a name...
Domain
Interactive Docker Compute Domain
Shared Intel Xeon E7 systems. All containers are limited to 100GiB of RAM. Unused containers are shut down after 3 days.
Compute Image
SciServer Essentials 2.0
Python 3.8 (Anaconda 2020.11), R 4.0.3, TensorFlow 2.3.0, PyTorch 1.7.1
User volumes
All
GerardsProject, Storage Volume created by gerard4demos
persistent, Storage Volume created by gerard4demos
scratch, Temporary Volume created by gerard4demos
Data volumes
All
Getting Started
HEASARC data
Manga
Ocean Circulation
Recount
SDSS DAS
SDSS Spectra
e Container File Storage
olders in a Container's file system should not be us
volumes under the Storage and Temporary folders
folders. Do not store your files in workspace, or in a
ill be lost permanently.
volumes for long term storage of your scripts and
ording to the username of the user who created the
o this folder persist between your containers, even
paths, i.e., under the Storage/username or Storage
ompute Data Storage Policy.
ll users start with a Storage volume named persiste
ompute.
jupyter
Files
Running
Clusters
Select items to perform actions on them.
0
/
GIT
indra
indra_dss
indra_filedb
Storage
Temporary
virgo
virgo_filedb
wfirst
jupyter ReadMillenniumSnapshots Last Checkpoint: 03/23/2017 (autosaved)
File
Edit
View
Insert
Cell
Kernel
Help
Python 3
Code
CellToolbar

# Cosmological Simulations on SciServer

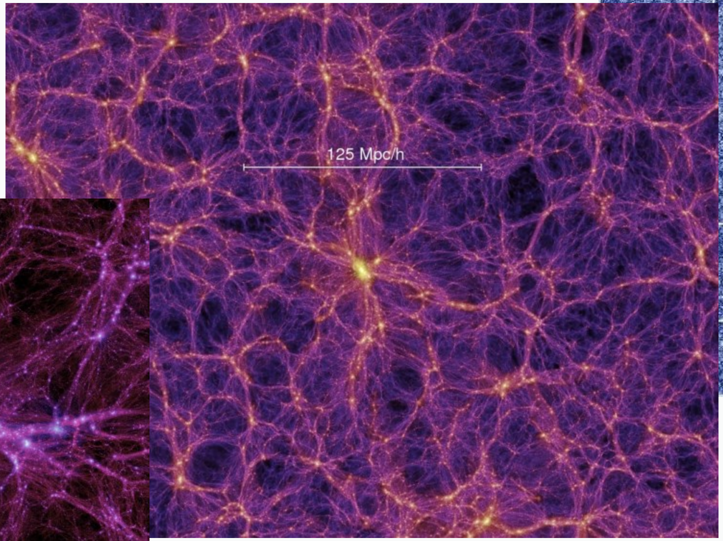
Data sets

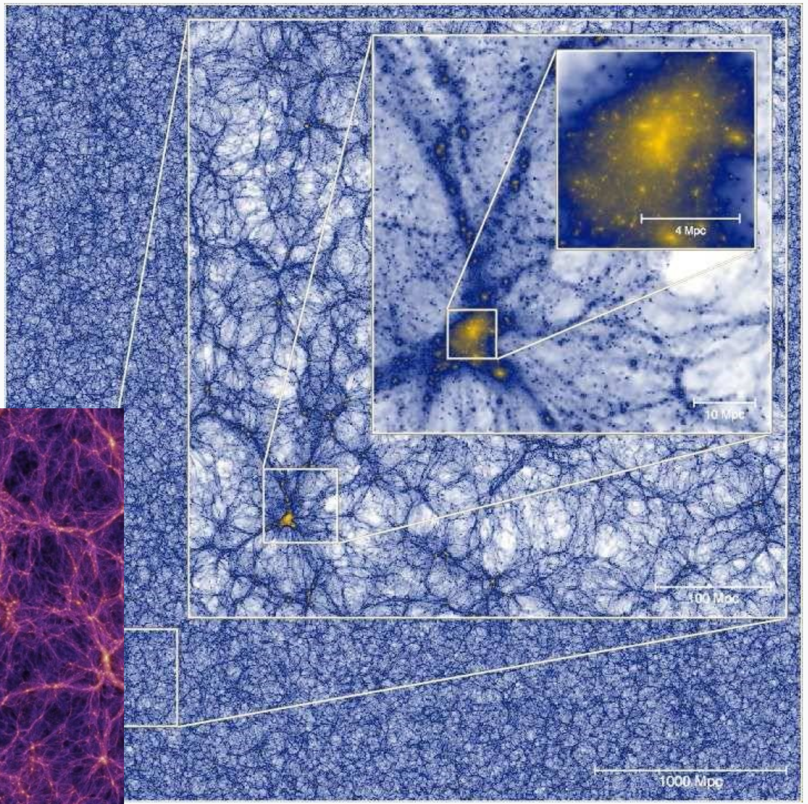
- Millennium-s ~70TB
- Indra ~700TB
- Eagle subset
- Illustris subset
- TBD VirgoDC

(II)



()




(XXL)

# Special features

- Relational database: Millennium, Indra, Eagle
  - Halos, merger trees, semi-analytic galaxies, light cones
- Access to raw data files
  - Direct mounted access in compute containers
  - From database through "FileDB"
- *Virtual Telescope* software
- Dask  cluster for distributed calculations
  - Bridget Falck: P(k) for all Indra simulations
- Custom codes, e.g.
  - LSST Simulation pipeline docker image
  - Nemo simulation code installed in user container

# Millennium halo density profiles (pure SQL)
# Fit to Hernquist profile (python)

jupyter CosmoUC_5_PlotHalo Last Checkpoint: 11 minutes ago (autosaved)

File Edit View Insert Cell Kernel Help

CellToolbar

## Dark-matter Halos in a Cosmological Simulation

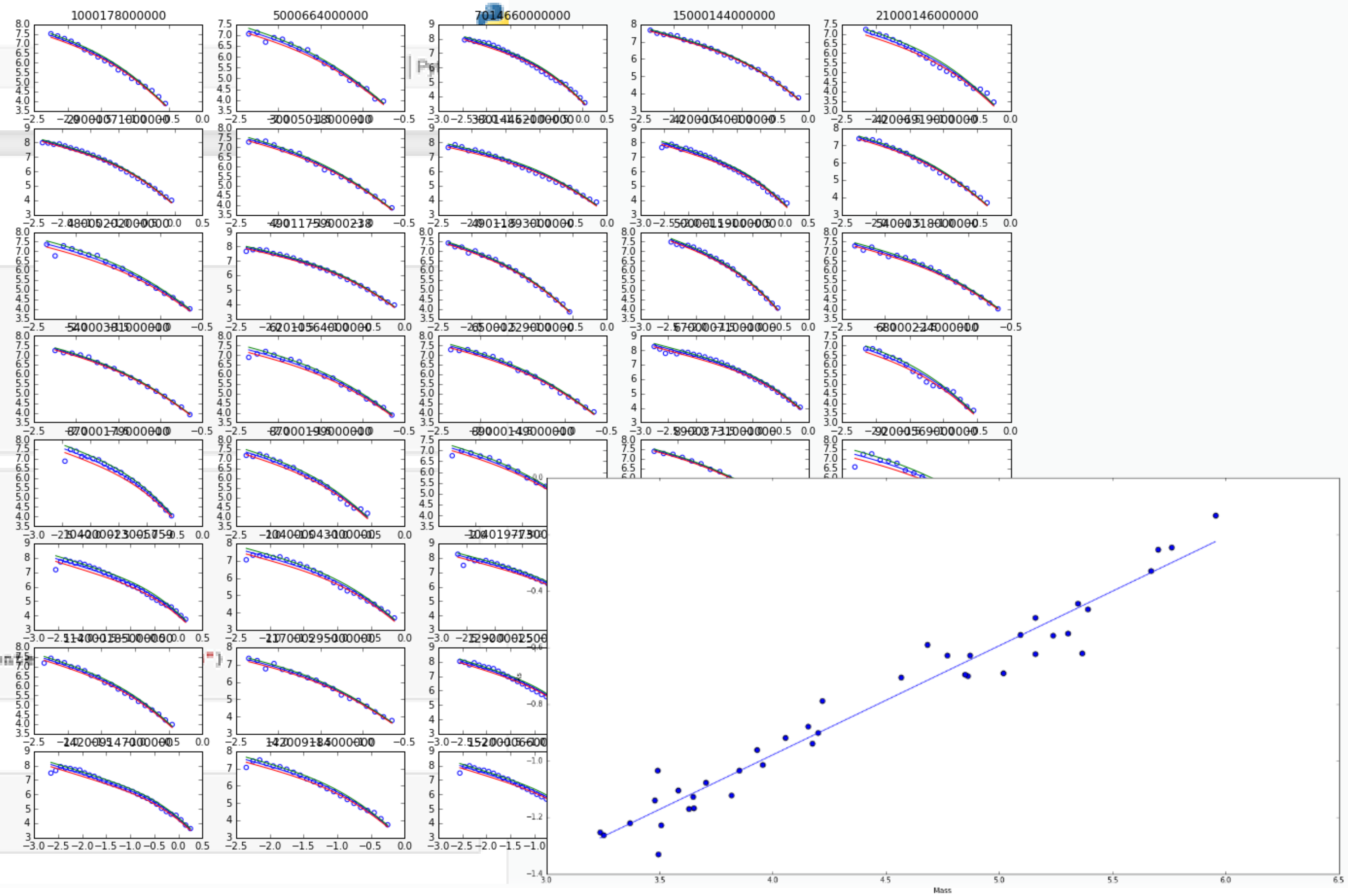
In [3]:
```
import SciServer.LoginPortal as Login
token = Login.getToken()
import SciServer.CasJobs
import pandas
import tables
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

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,con
df = pandas.read_csv(responseStream, index_col=None)
```

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

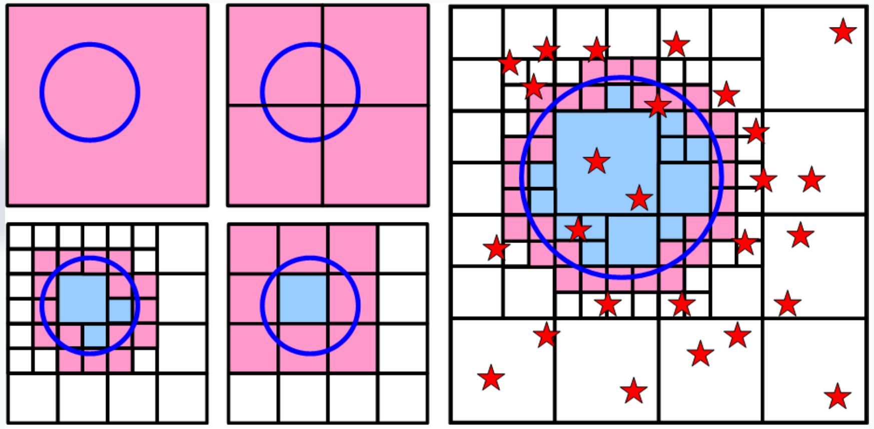
In [13]:
```
fig = plt.figure(figsize=(15, 15))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(df.x,df.y, df.z,s=0.001)
```

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

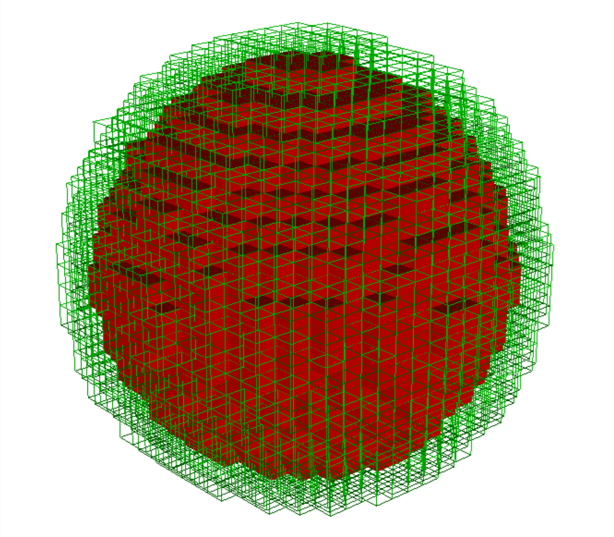


# Query particles in sphere/box

- Divide simulation volume in regular grid
- Index using space filling curve
- Calculate overlap space filling curve with query volume



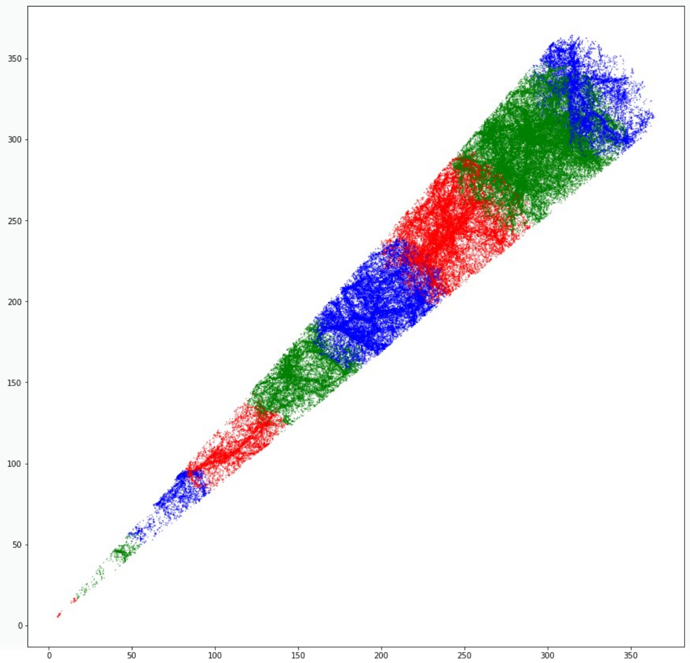
(a) (b)

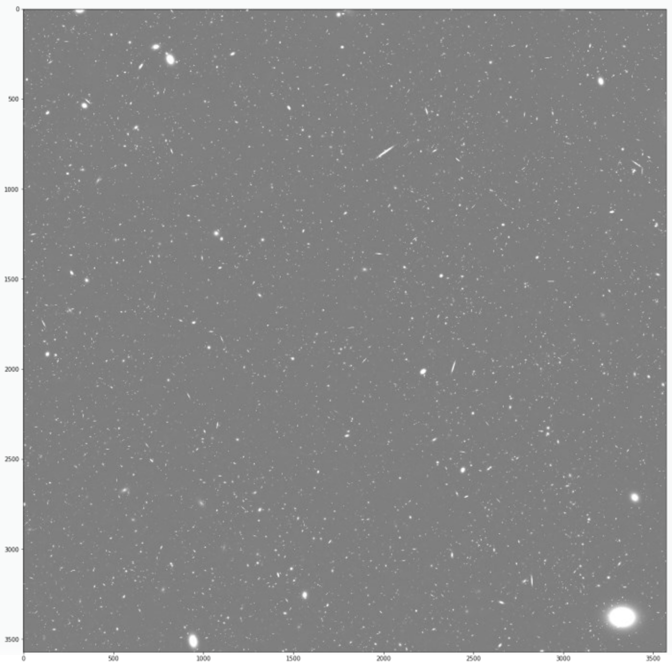


- Execute as table-valued function from database
- Technology: C# for SQLCLR table-valued-function

# Virtual telescopes

- Light cone using python for snapshot boundaries and SQL for data retrieval
- Synthetic image using Millennium Run Observatory code (Overzier etal 2013)



# Eagle: relational source catalog plus raw data

## read Eagle snapshot data

- Use database to find most massive cluster,
- Cutout region around it and find its gas and star particles
- plot these

In [ ]:
```
from pyread_eagle import EagleSnapshot
import SciServer.CasJobs as cj
import matplotlib.pyplot as plt
```

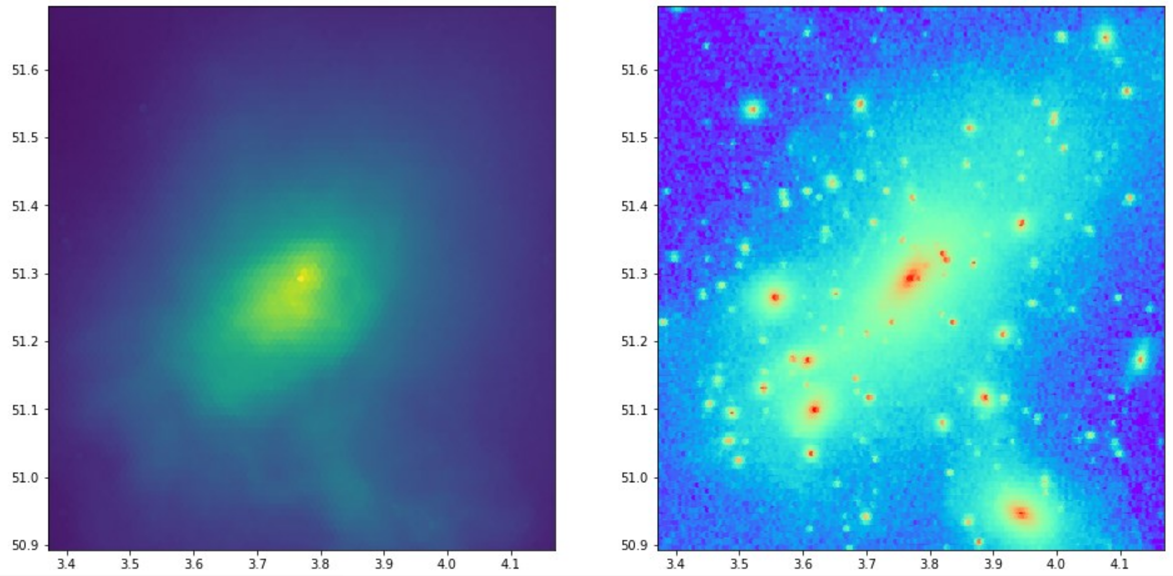
In [ ]:
```
# find 10 most massive subhalos
sql="""
select top 10 *
  from RefL0100N1504_Subhalo
  where snapnum=28
  order by mass desc
"""
df=cj.executeQuery(sql,"eagle_release")
```

In [ ]:
```
snap = EagleSnapshot("/home/idies/workspace/virgo/Eagle/L0100N1504/snapshot_028_z000p000/snap_028_z000p000.0.hdf5")
```

In [ ]:
```
# multiply by .6777 for proper scaling between database and particle positions
centers=df[['CentreOfPotential_x','CentreOfPotential_y','CentreOfPotential_z']].values*.6777
c=centers[0];lims=[c[0]-.4,c[0]+.4,c[1]-.4,c[1]+.4,c[2]-.4,c[2]+.4]
snap.select_region(*lims)
```

In [ ]:
```
gas = snap.read_dataset(0, "Coordinates")
stars = snap.read_dataset(4, "Coordinates")
```

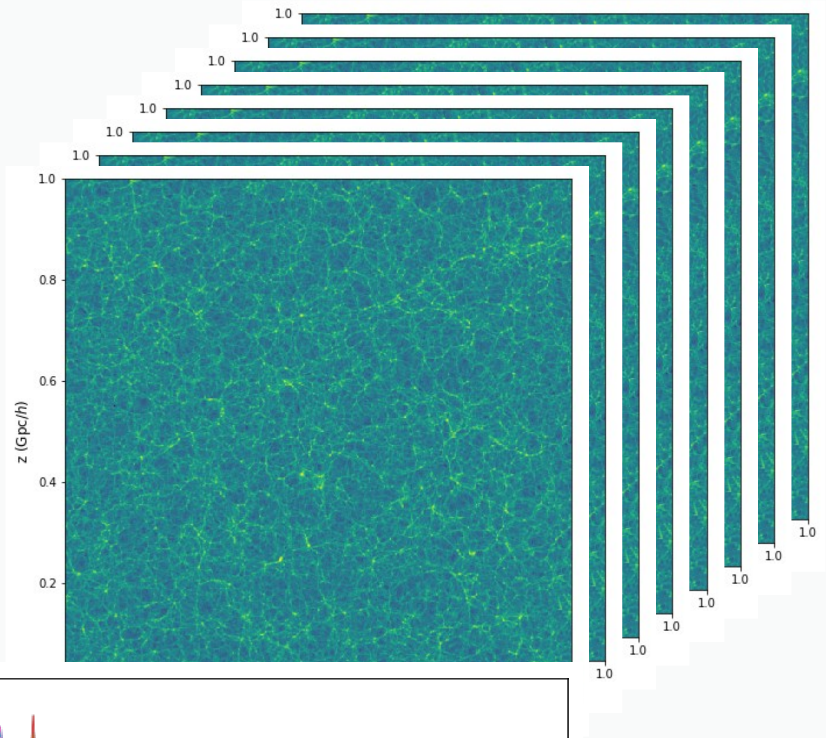
In [6]:
```
# plot gas and stars
fig, (ax1, ax2) = plt.subplots(1, 2,figsize=(16,8))
ax1.hexbin(gas[:,0],gas[:,1])
ax2.hexbin(stars[:,0],stars[:,1],bins='log',gridsize=200,cmap='rainbow');
ax1.set_xlim(lims[0:2]);ax1.set_ylim(lims[2:4]);ax2.set_xlim(lims[0:2]);ax2.set_ylim(lims[2:4]);
```

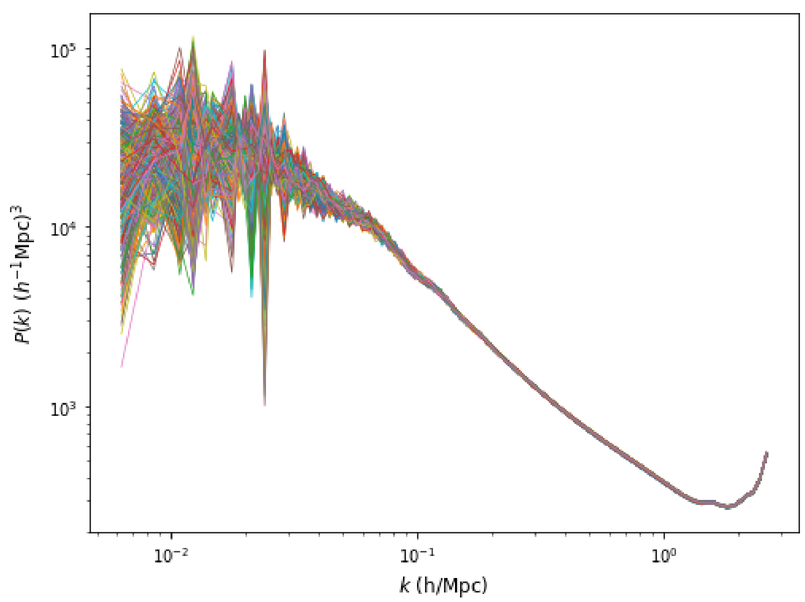


# DASK on Indra

**Bridget Falck, interop Nov 2020, arXiv:2101.036310**

- Distributed file system
- DASK parallel python cluster
- 448 Cloud-In-Cell density grids
- Power spectrum calculation on eac
- 481 billion particles in total
- 2 hours
- (Not generally available yet)



# Ready for standardization?

- Databases
  - TAP ok for simulations?
  - TAP on no-sql backends?
- Direct access I/O libraries from data providers iso DAL protocols
  - astroquery-like
  - *yt*
- Containerization for compute
  - How represent data stores?
  - Upload (docker) images?
  - Need MPI?
  - Role for Kubernetes
- Jupyter
  - Interactive?
  - UWS for batch?
- Authentication and access privileges
  - GWS Groups

# Questions

- Sustainability: Who pays for it all?
  - SciServer involved in grant proposals, but sufficient for Astro?
- Can the cloud help?
  - https://www.nature.com/articles/d41586-020-02284-7
  - Astronomy Data Commons?
    (Juric etal, Cambridge, MA, 2020)
  - Expensive for storage and egress: Are simulators rich enough?
  - But compute scales elastically and maybe user can be asked to pay?
  - Hybrid solutions, cloud overflow
- Which simulation products most valuable?
  - Raw or derived?
  - What use cases? “virtual observatories”?
  - Piggy backing on platforms for the large observational programs?

# Thank you