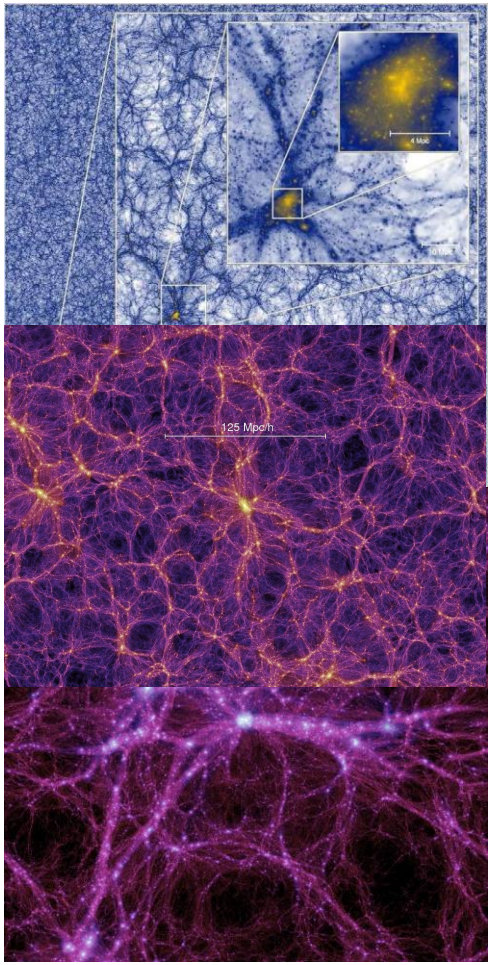
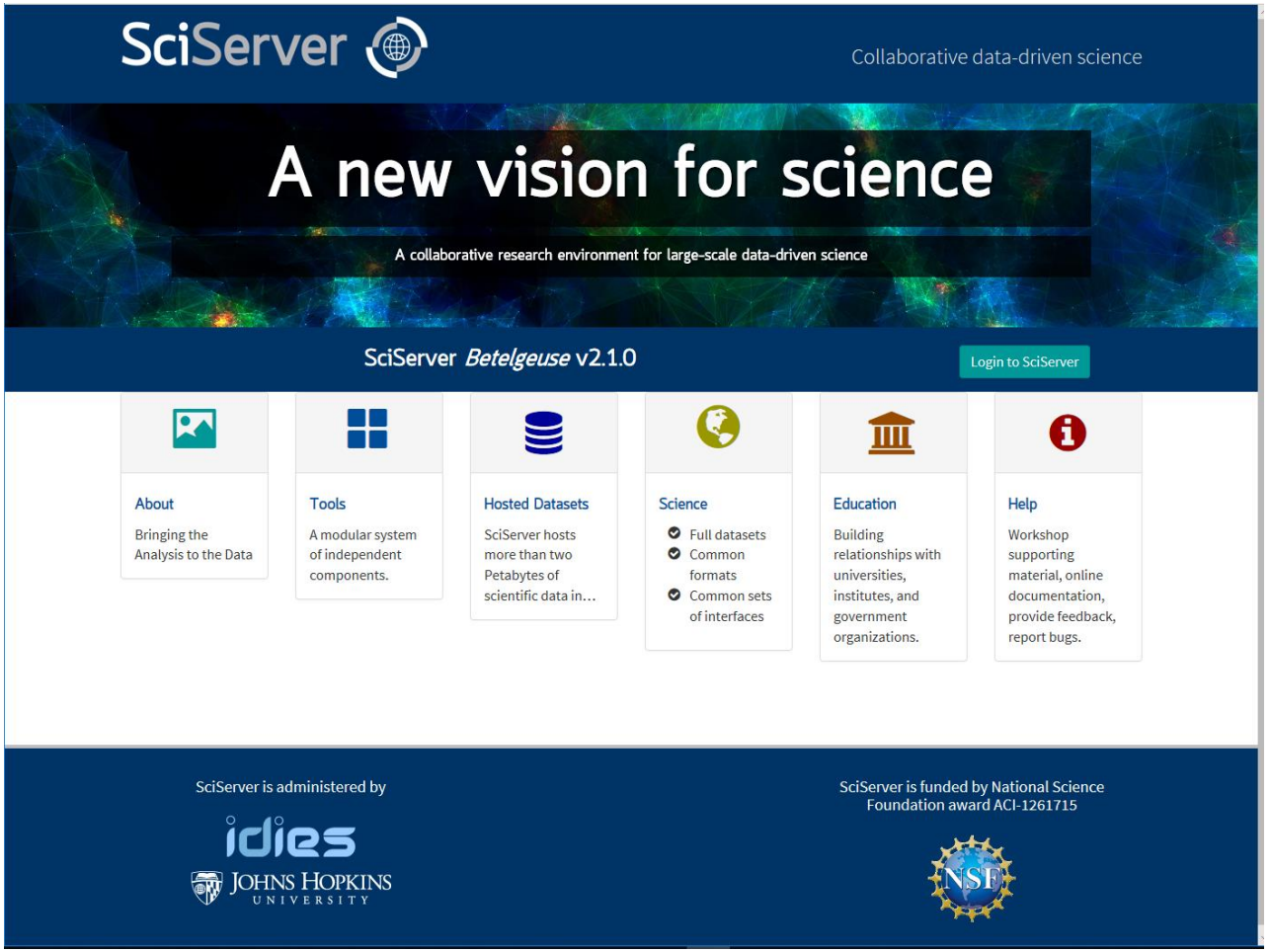
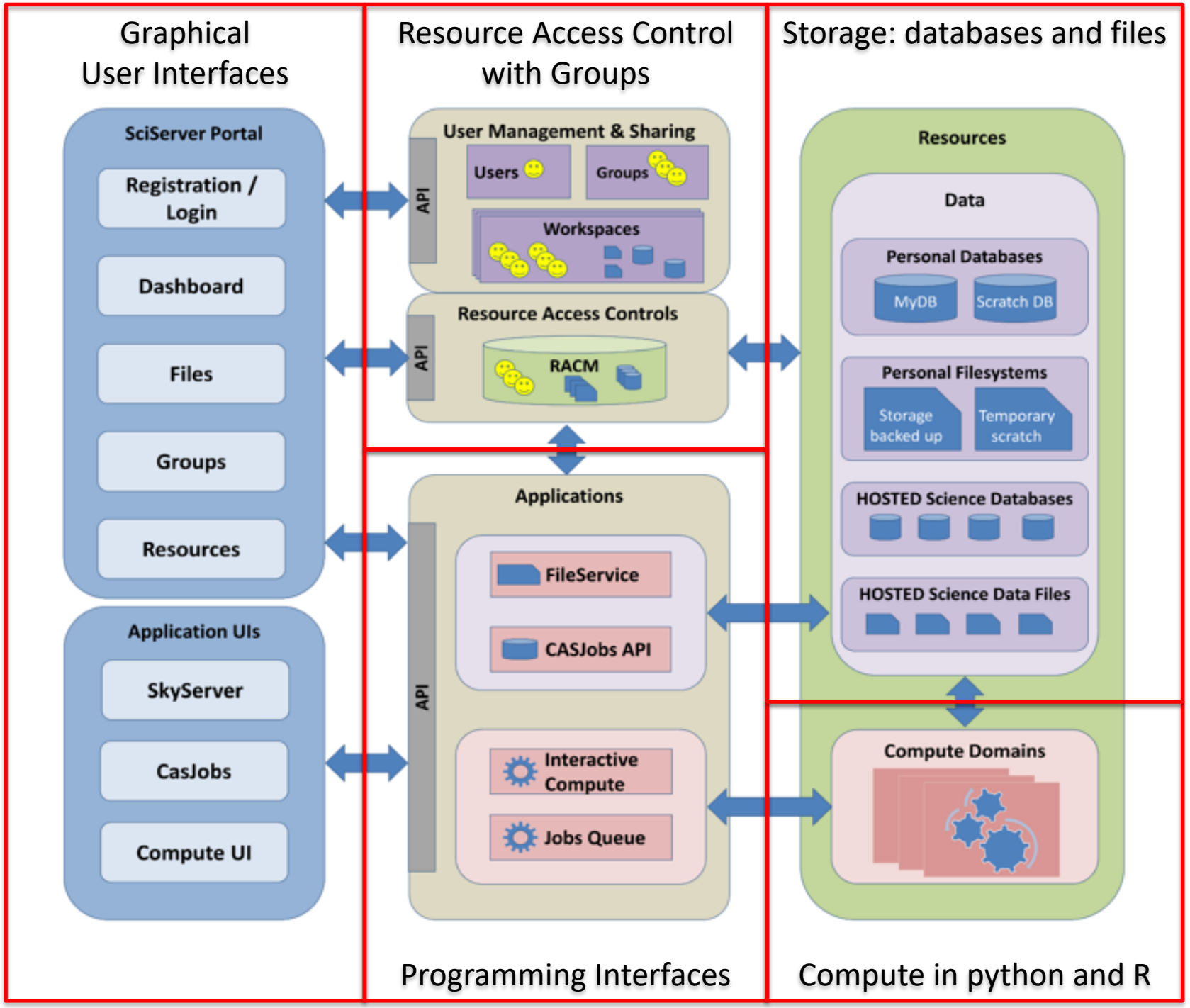
Simulations in a Science Platform

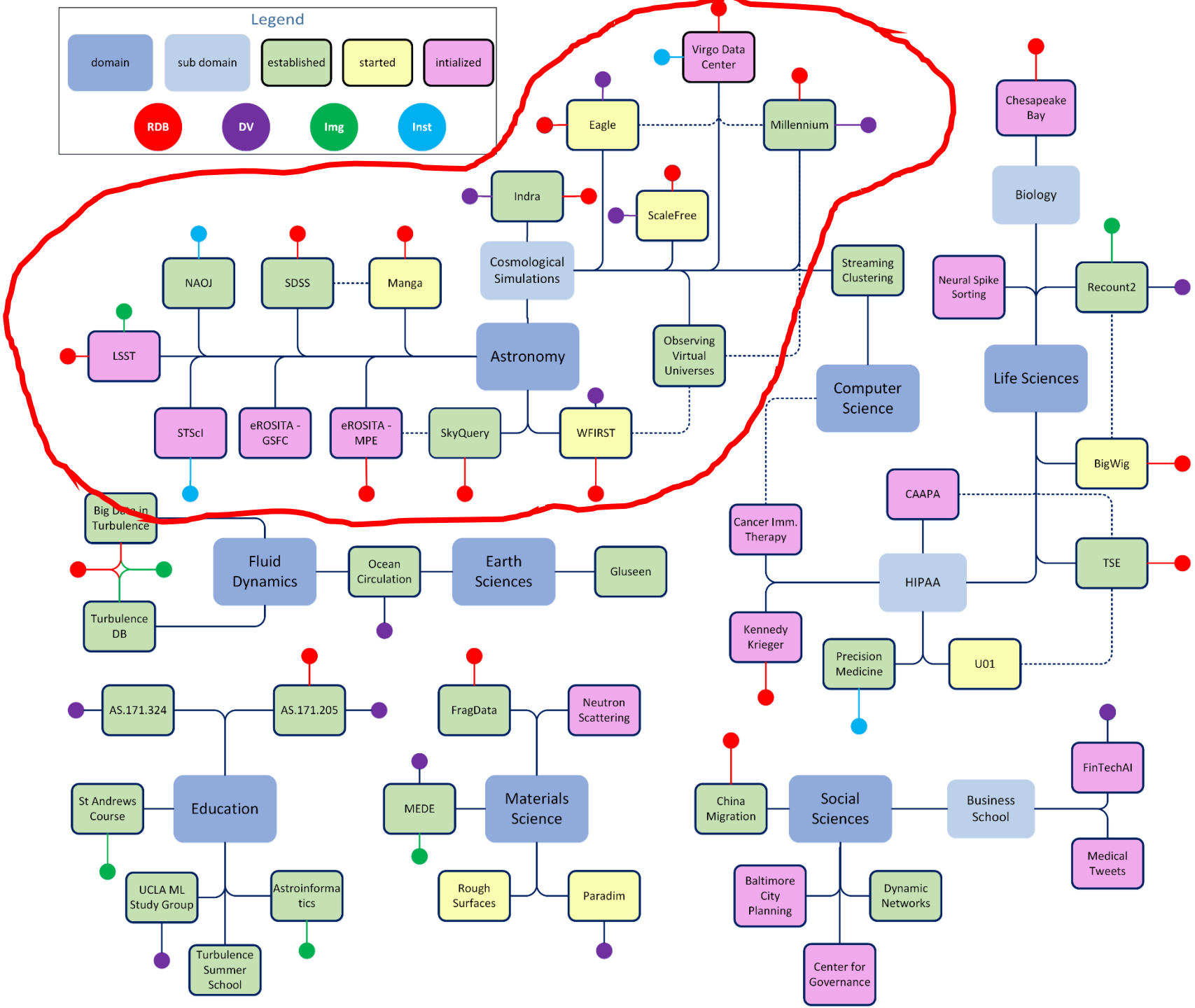




Overview

- SciServer, a science platform @ JHU
- Data
  - Databases
  - files
- Services
  - Applications
  - Special database functions
- Use in WFIRST Archive Science Investigation Team
  - E.g. STIPS (http://www.stsci.edu/wfirst/science-planning-toolbox/stips)
- Possible role for theory IG?


Graphical User Interfaces
Resource Access Control with Groups
Storage: databases and files
SciServer Portal
Registration / Login
Dashboard
Files
Groups
Resources
Application UIs
SkyServer
CasJobs
Compute UI
API
User Management & Sharing
Users
Groups
Workspaces
Resource Access Controls
RACM
Applications
FileService
CASJobs API
Interactive Compute
Jobs Queue
Resources
Data
Personal Databases
MyDB
Scratch DB
Personal Filesystems
Storage backed up
Temporary scratch
HOSTED Science Databases
HOSTED Science Data Files
Compute Domains
Programming Interfaces
Compute in python and R


Legend
domain
sub domain
established
started
intialized
RDB
DV
Img
Inst
Virgo Data Center
Eagle
Millennium
Indra
ScaleFree
Cosmological Simulations
NAOJ
SDSS
Manga
LSST
Astronomy
Observing Virtual Universes
STScI
eROSITA - GSFC
eROSITA - MPE
SkyQuery
WFIRST
Streaming Clustering
Computer Science
Chesapeake Bay
Biology
Neural Spike Sorting
Recount2
Life Sciences
BigWig
CAAPA
Cancer Imm. Therapy
HIPAA
TSE
Kennedy Krieger
Precision Medicine
U01
Big Data in Turbulence
Turbulence DB
Fluid Dynamics
Ocean Circulation
Earth Sciences
Gluseen
AS.171.324
AS.171.205
St Andrews Course
Education
UCLA ML Study Group
Astroinformatics
Turbulence Summer School
FragData
Neutron Scattering
MEDE
Materials Science
Rough Surfaces
Paradim
China Migration
Social Sciences
Business School
FinTechAI
Medical Tweets
Baltimore City Planning
Dynamic Networks
Center for Governance

Links

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

Astronomy

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=865819437896092880 9) 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]:

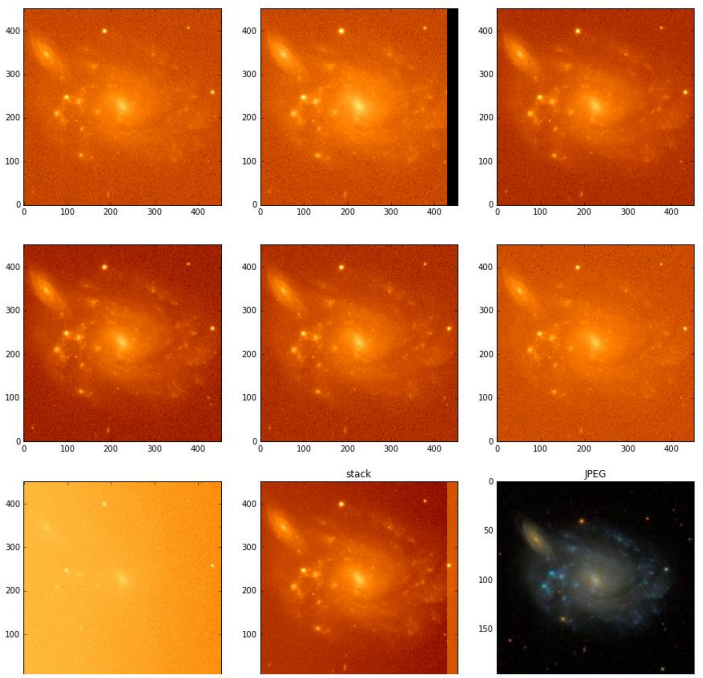
	head	match	matchcount	head_field	match_field	fits_g
0	8658194378960928809	8658194430499553320	57	8658194378960928768	8658194430499553280	http://das.sdss.org/imaging/5622/40/corr
1	8658194378960928809	8658194477742948377	57	8658194378960928768	8658194477742948352	http://das.sdss.org/imaging/5633/40/corr
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>

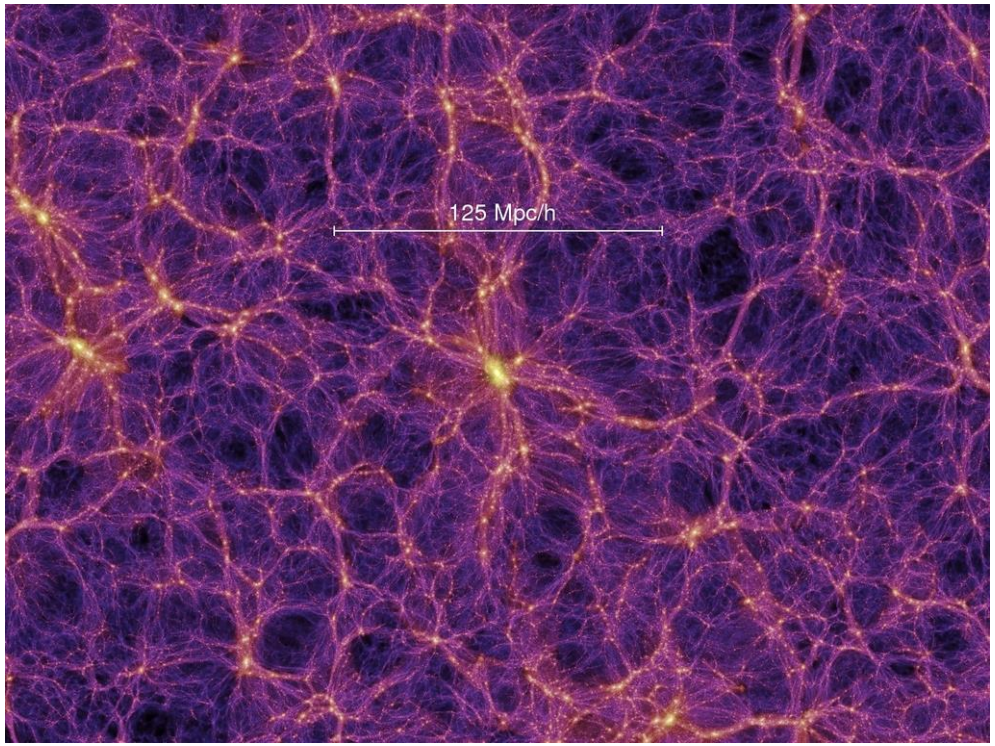


Cosmological Simulations

E.g. Millennium Run

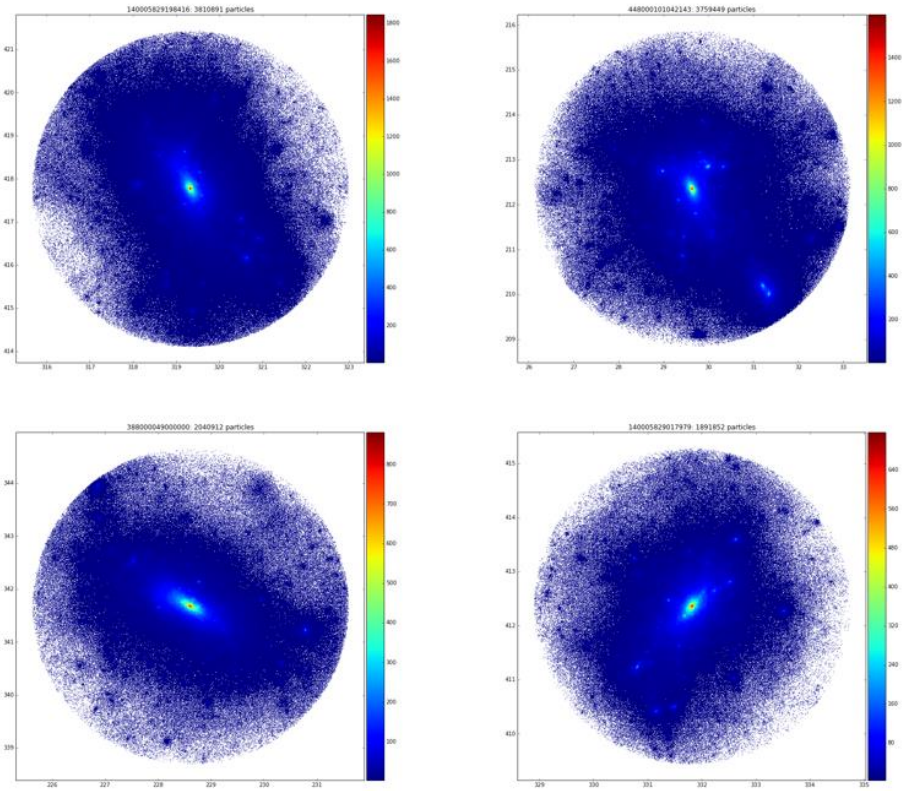
- 500 Mpc/h
- 10^{10} particles
- 8.6 10^{8} M_{sun}/h
- ~18 million halos
- ~300GB/snapshot

Springel etal 2005



```
    ax.set_title(title)
    divider = make_axes_locatable(ax)
    # Append axes to the right of ax3, with 20% width of ax3
    cax = divider.append_axes("right", size="5%", pad=0.05)
    # Create colorbar in the appended axes
    # Tick locations can be set with the kwarg `ticks`
    # and the format of the ticklabels with kwarg `format`
    cbar = plt.colorbar(im, cax=cax)#, ticks=MultipleLocator(0.2), format="%.2f")
```

```
CPU times: user 17.4 s, sys: 3.87 s, total: 21.3 s
Wall time: 27 s
```



In [ ]:

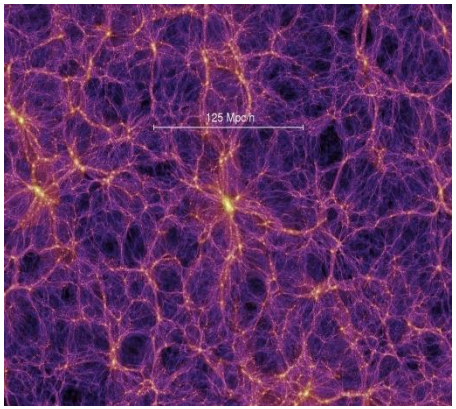
In [ ]:

OLD OLD OLD OLD OLD OLD OLD

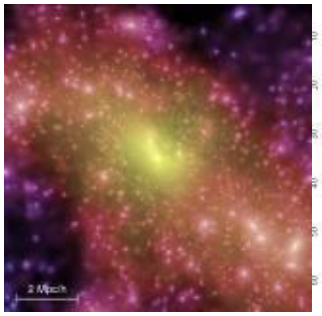
In [ ]:
```
def getParticles(url,shift,gr,shape):
    headers={"Range":"bytes=%s" % ",".join("%d-%d" % (g[4],g[4]+g[5]) for g in gr.itertuples())}

    resp=requests.get(url,headers=headers,stream=True)
    if resp.status_code != 206:
```

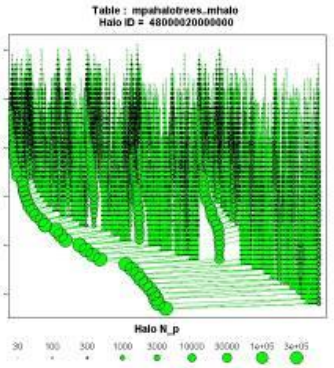
Millennium Run Observatory

Raw data:
Particles

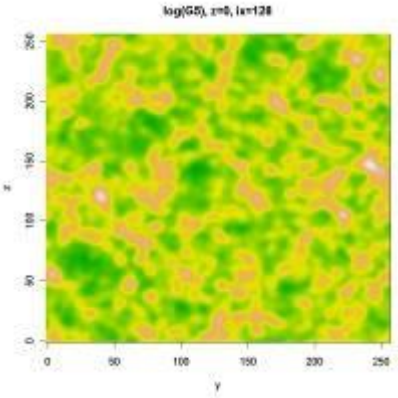


Subhalo
merger trees



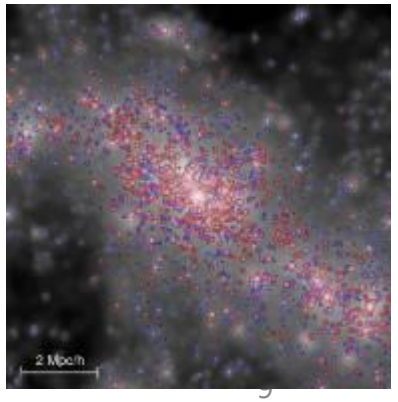


Density fields

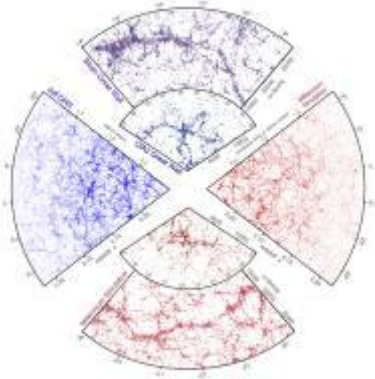


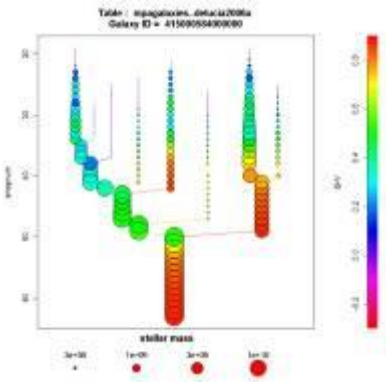
Synthetic
images

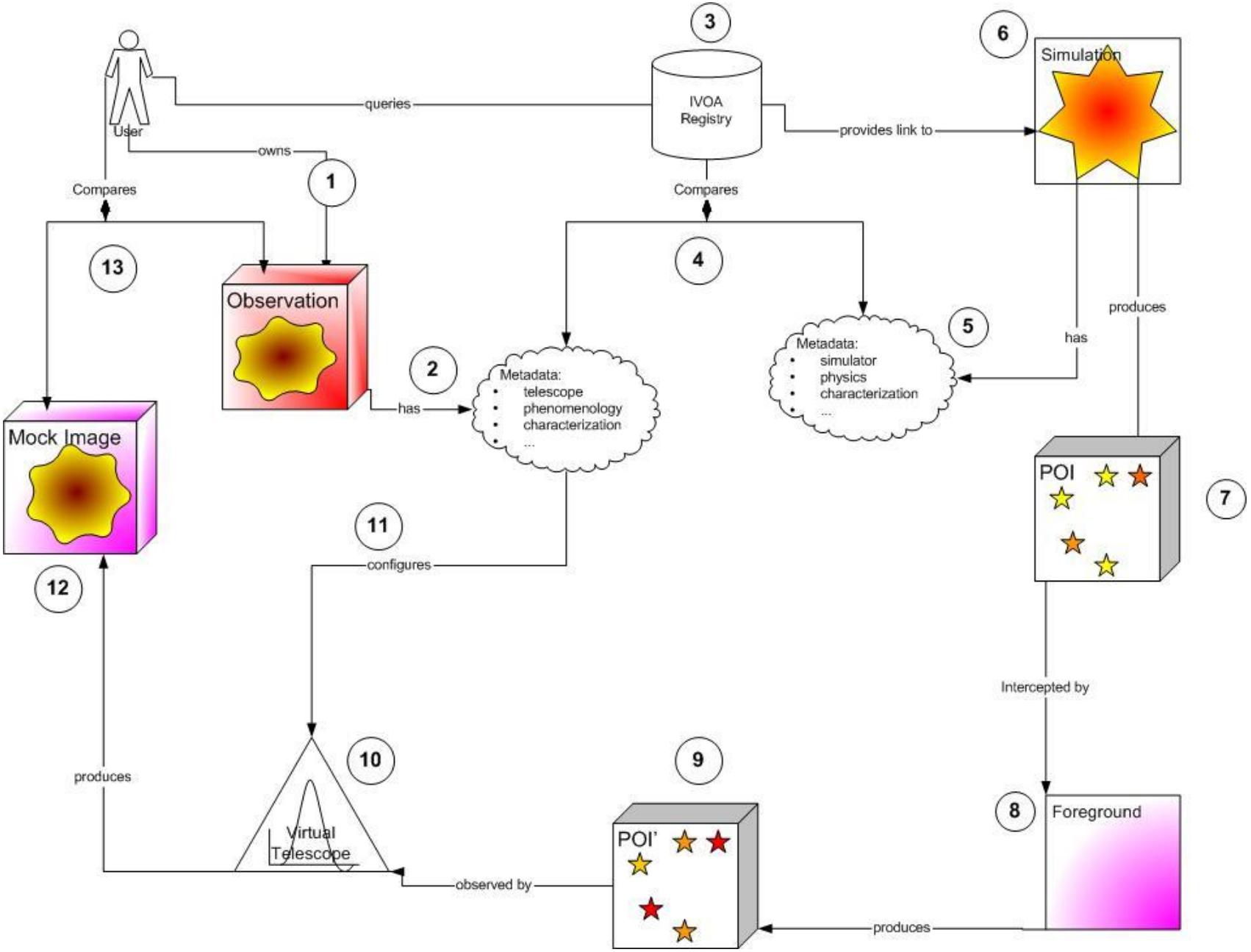
Synthetic galaxies



Mock catalogues

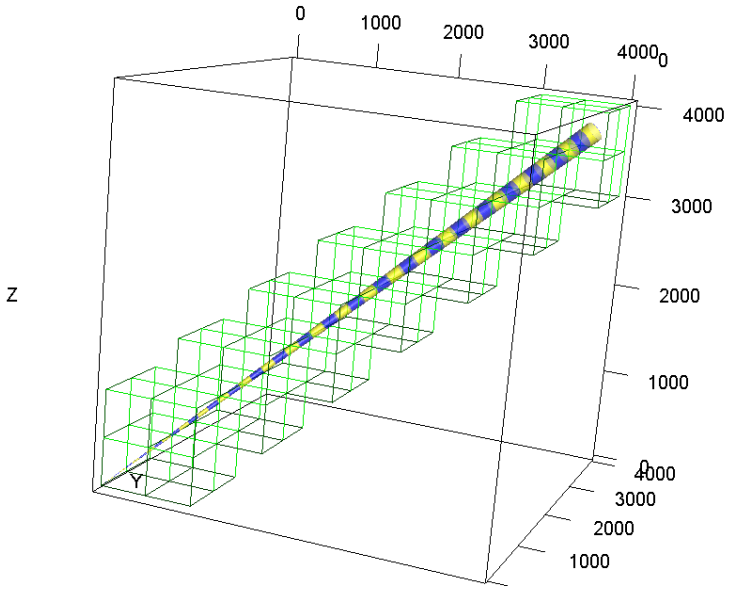


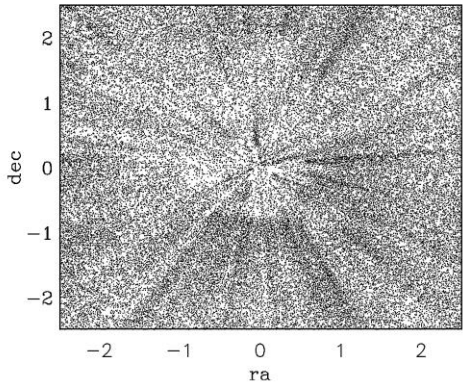


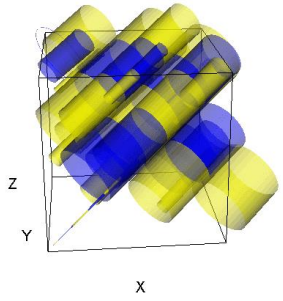

3
6
Simulation
queries
IVOA
Registry
provides link to
User
owns
1
Compares
Compares
13
4
Observation
produces
5
has
Metadata:
simulator
physics
characterization
...
2
Metadata:
telescope
phenomenology
characterization
...
has
Mock Image
POI
7
11
configures
12
Intercepted by
produces
10
9
8
Foreground
Virtual
Telescope
POI'
observed by
produces

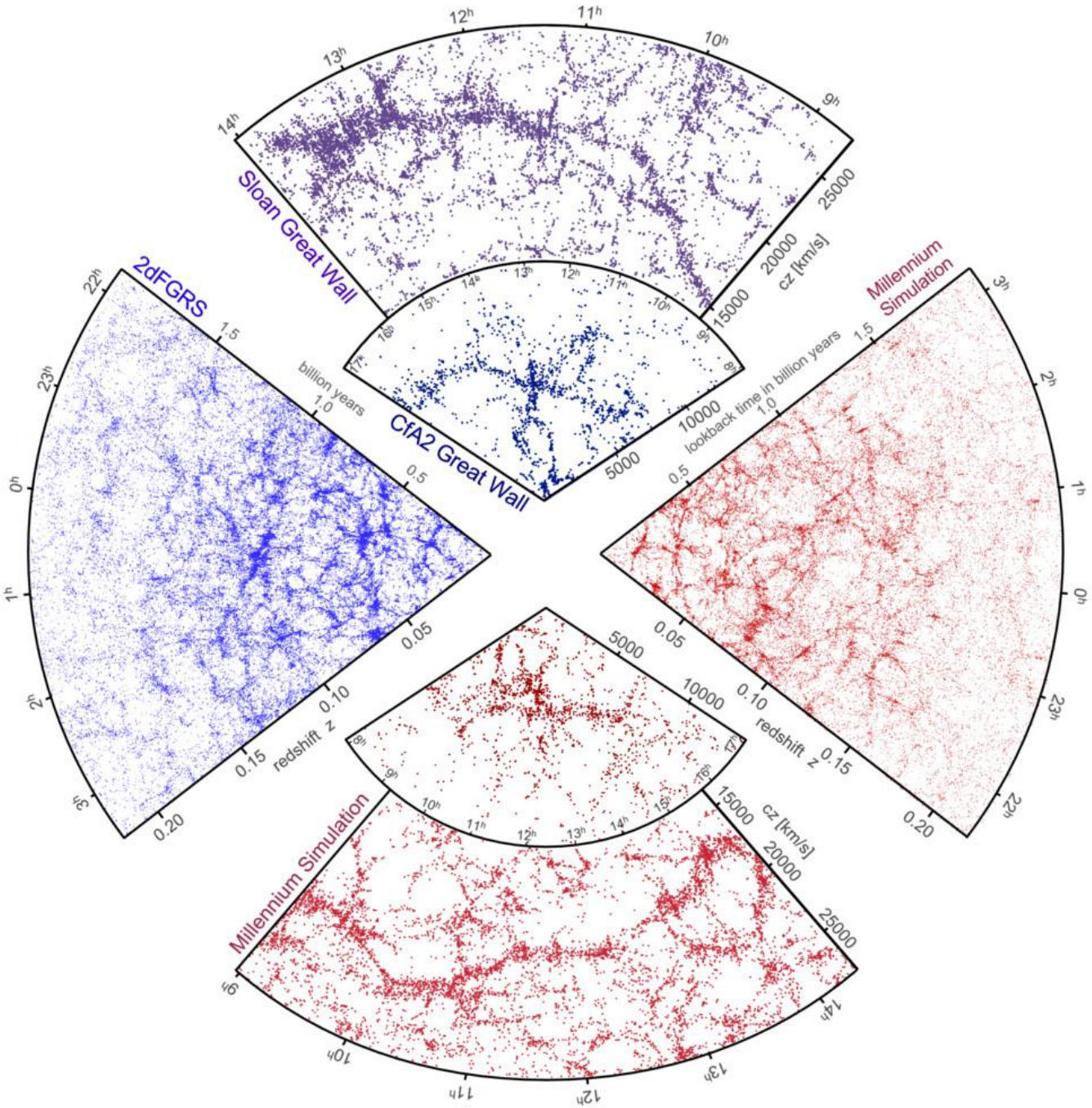
Mock light-cone catalogues

- Use periodic boundary conditions
  - Blaizot etal (2005)
  - Kitzbichler & White (2006)
  - Henriques etal (2012, 2015)



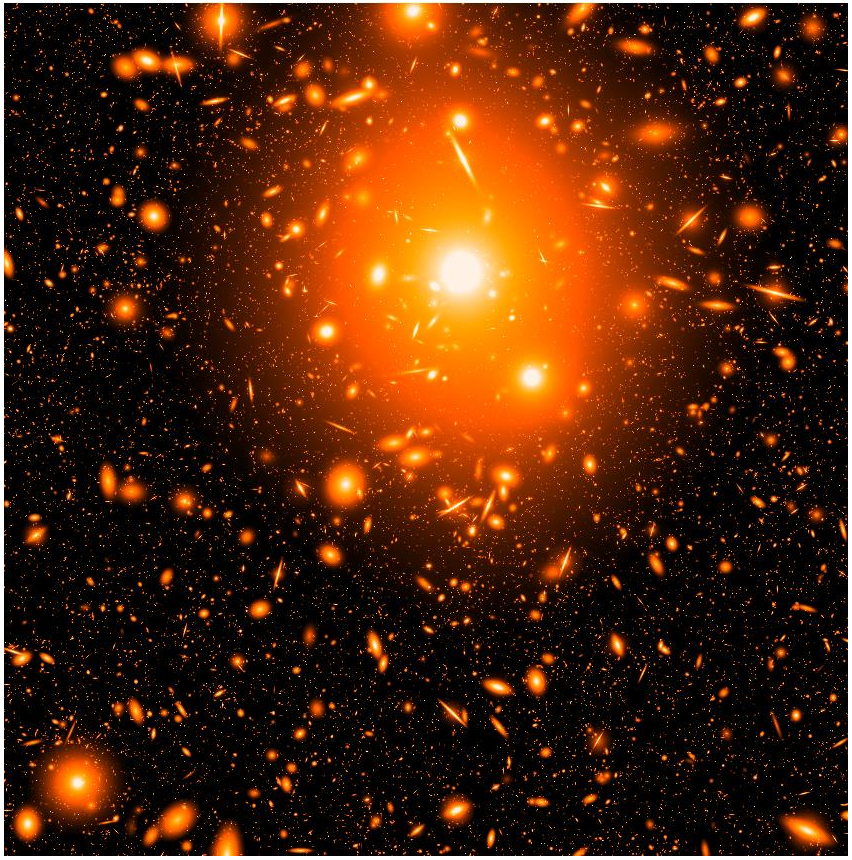


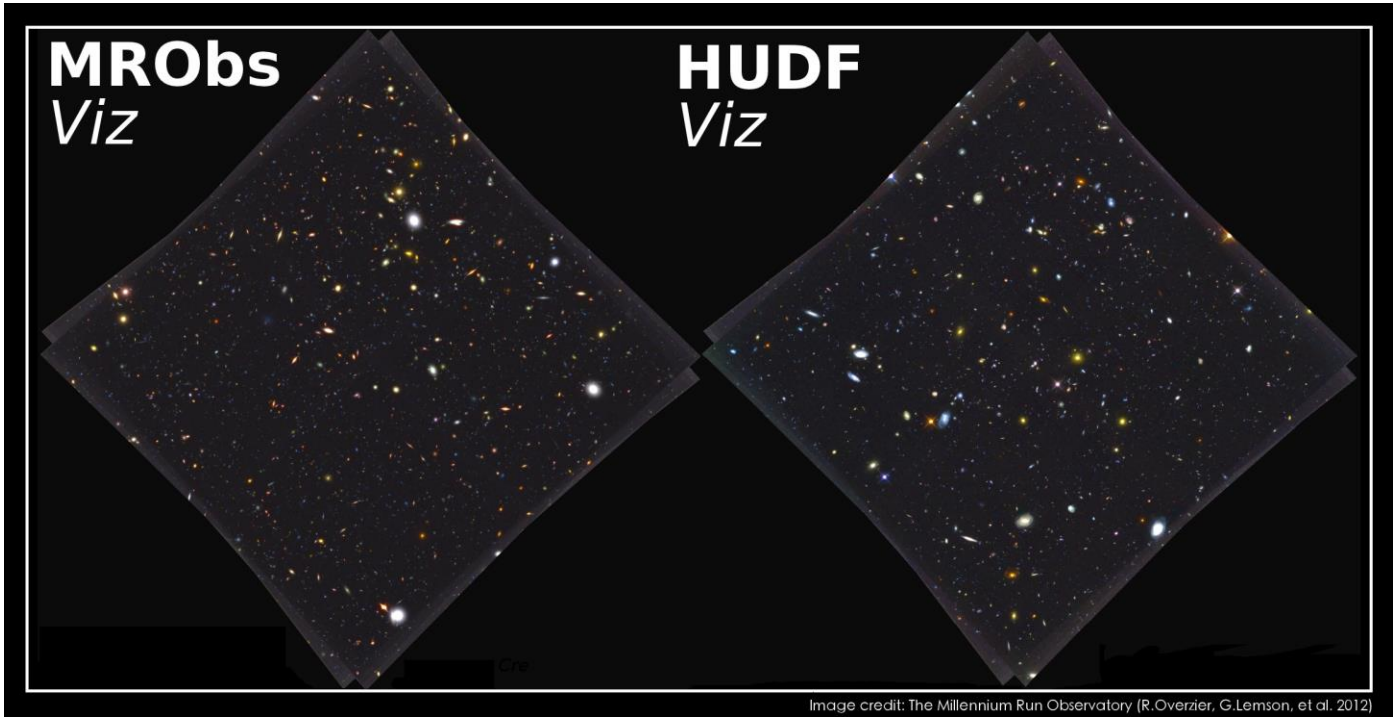



Sloan Great Wall
2dFGRS
CfA2 Great Wall
Millennium Simulation
Millennium Simulation
cz [km/s]
billion years
lookback time in billion years
redshift z

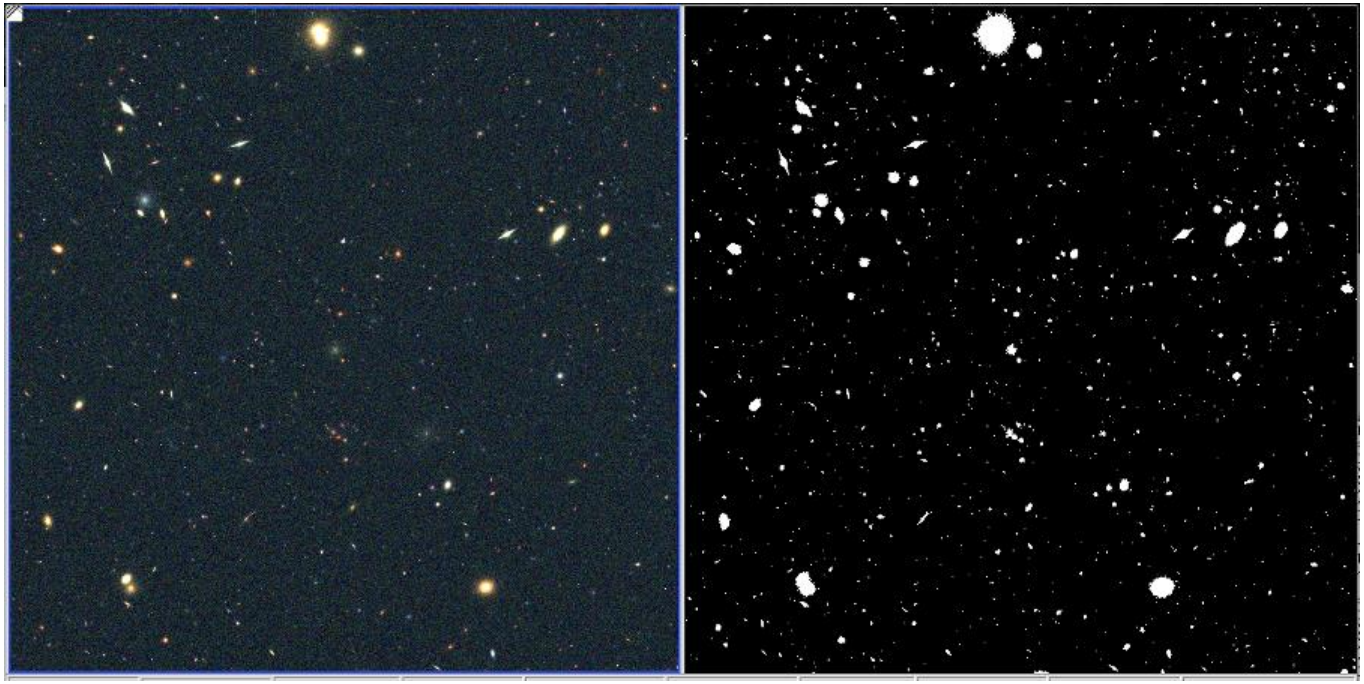
SkyMaker (E. Bertin, 2008)

- Ideal image: bulges and disks
- Inclination and PA from model
- Adding own PSF, noise etc




MRObs
Viz
HUDF
Viz
Image credit: The Millennium Run Observatory (R.Overzier, G.Lemson, et al. 2012)

Taking it one step further...
Source Extracting from Simulated Images

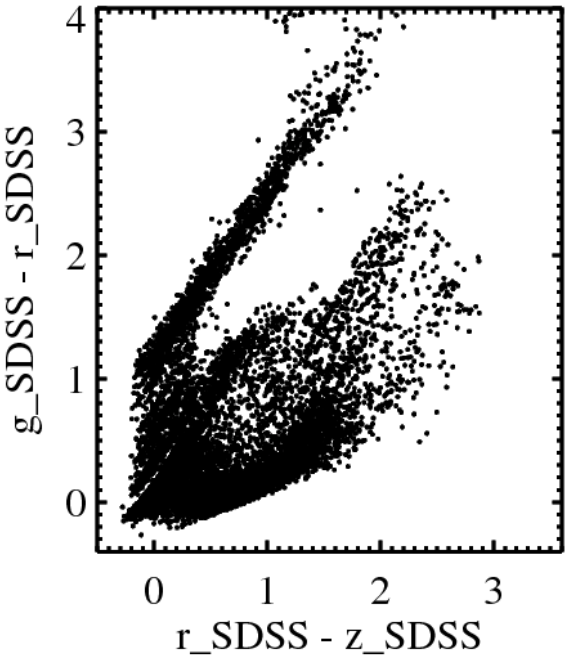


Simulated image

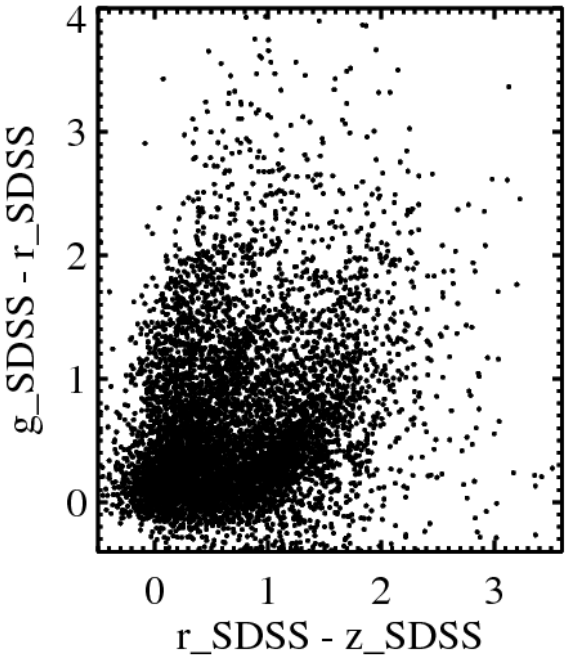
SExtractor segmentation image

courtesy Roderik Overzier

Source Extracting from Simulated Images
Color-color plot Lightcone vs. Recovered



MR Mock Lightcone



Extracted from
MR Mock Image

courtesy Roderik Overzier

skymaker-example

Secure | https://apps.sciserver.org/dockervm16/1daa5bdb-22e0-11e8-96b1-...

jupyter skymaker-example (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted

Markdown

Example to run skymaker

Input file: sdss_g.csv.
Sky command: sky.
Configuration file: sky_default.conf

sky sdss_g.csv -c sky_default.conf -IMAGE_SIZE 3571,3571 -PIXEL_SIZE 0.168 -BA(
21.8 -MAG_ZEROPOINT 27.1 -WAVELENGTH 0.4839 -PSF_NAME psf_g.fits -PSF_T
PSF_MAPSIZE 128 -SEEING_FWHM 0.0 -SEEING_TYPE NONE -PSF_OVERSAMP
-WELL_CAPACITY 0.0 -SATUR_LEVEL 0.0 -READOUT_NOISE 4.5 -IMAGE_NAME s
EXPOSURE_TIME 600.0 -IMAGE_TYPE SKY_NONOISE -STARCOUNT_ZP 0

SQL Query to generate the CSV file

```
declare @pixel real set @pixel=0.168 ;
with g as(
select *
,      case when ra > 200. then ra - 360. else ra end as ra_update
  from Henriques2015a.cones.MRscPlanck1_BC03_001
)
select 200 as [type]
,      ((-1)*(1.0*(g.ra_updated-(0.0)))+0.0833333333333)*3600./@pi
x_pix
,      ( 1.0*(g.dec-(0.0))+0.0833333333333)*3600./@pixel as y_pix
,      (g.sdss_g) as mag_total
,      case when -0.4*(g.sdss_g_bulge-g.sdss_g) < -10 then 0.
            else power(convert(real,10),-0.4*(g.sdss_g_bulge-g.sds
end as bulge_to_total
,      geom.bulgesize as re_bulge_arcsec
,      1.0 as bulge_proj_aspect_ratio
,      (90.-geom.pa) as bulge_pa
,      (geom.disksize/3.) as rh_disk_arcsec
,      case when cos(pi()*geom.inclination/180) < 0.1 then 0.1
            else cos(pi()*geom.inclination/180) end as disk_inclin
,      (90.-geom.pa) as disk_pa
,      g.galaxyid
  from g
  ,    Henriques2015a.cones.MRscPlanck1_BC03_001 geom
 where geom.galaxyid = g.galaxyid
   and 1.0*g.ra_updated between -0.0875 and 0.0875
   and 1.0*g.dec between -0.0875 and 0.0875
```

In [1]:
```
import os
import numpy as np
```

skymaker-example

Secure | https://apps.sciserver.org/dockervm16/1daa5bdb-22e0-11e8-96b1-...

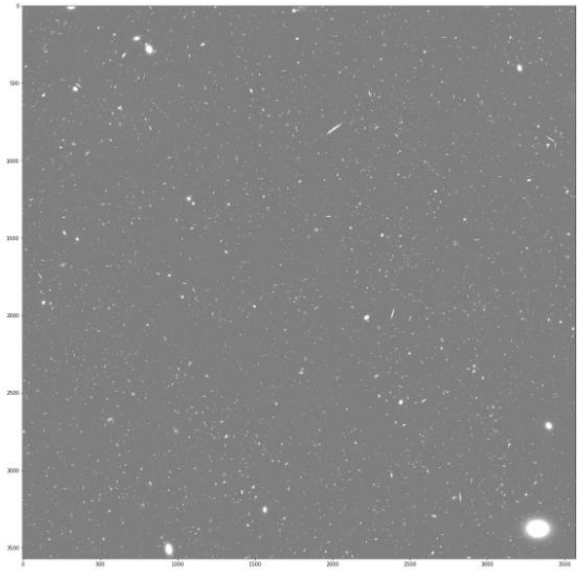
jupyter skymaker-example (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 2

Markdown

```
sdss_g.fits -EXPOSURE_TIME 600.0 -IMAGE_TYPE SKY_NONOISE -STARCOUNT_ZP 0
```

In [5]:
```
fitsFile = fits.open('./sdss_g.fits')
im = np.array(fitsFile[0].data)
plt.figure(figsize=(20,20))
image_median = np.median(im)
image_sd = np.std(im)
plt.imshow(im, cmap='gray', norm=Normalize(vmin=image_median-3*image_sd,vmax=imag
plt.show()
```



In [ ]:

Interoperability? Standardization?

- Many surveys use simulations for testing pipeline and software
- Generally produced by own teams
- Rarely reuse existing simulations
- Can these be used by others
- Same for “virtual telescopes” and other post-processing software

Proposal

- Contact big surveys
  - Some already active in IVOA
- Get info on their simulations and testing pipelines
- Would they be interested in sharing?
- If so, find commonalities in data and tools if any
  - Looking for interoperability
- Science platforms will likely play an important role
  - Analysis must be done close to the data
- Role for SimDB?