



# Towards New Science with the LSST

*Envisioning Platforms for Science with Large  
Datasets*

**Prof. Mario Juric**

DIRAC Institute | eScience Institute | UW Astronomy

With Eric Bellm, Robert Lupton, Zeljko Ivezic, Andy  
Connolly, Colin Slater, and many, many, many  
other colleagues in the  
LSST Project!

DATA INTENSIVE RESEARCH IN ASTROPHYSICS AND COSMOLOGY  
COLLEGE OF ARTS & SCIENCES | UNIVERSITY of WASHINGTON



# Introducing the LSST

# LSST: A Deep, Wide, Fast, Optical Sky Survey

8.4m telescope

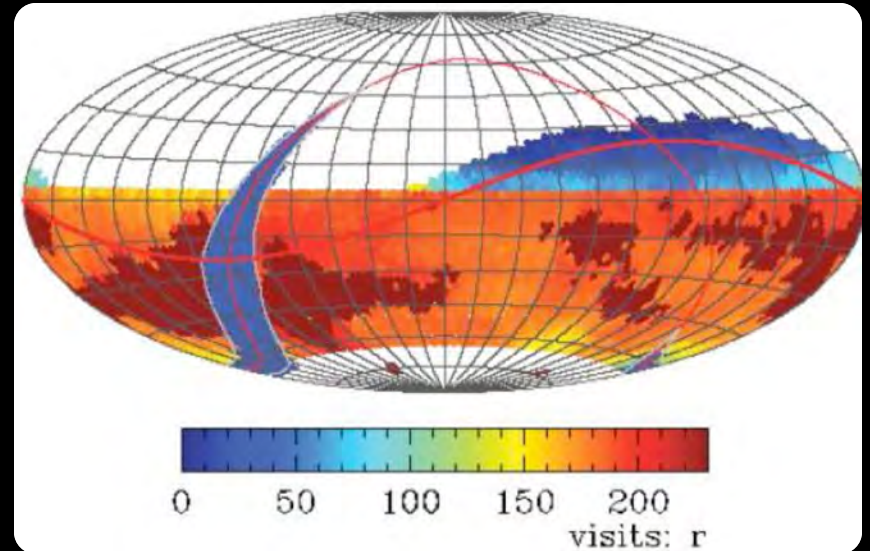
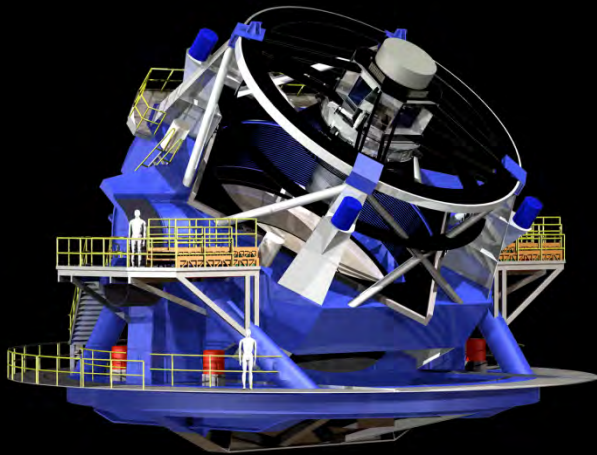
18000+ deg<sup>2</sup>

10mas astrom.

$r < 24.5$  ( $< 27.5$  @ 10yr)

ugrizy

0.5-1% photometry



3.2Gpix camera

30sec exp/4sec rd

15TB/night

37 B objects

Imaging the visible sky, once every 3 days, for 10 years (825 revisits)  
Alerts public, other data proprietary for two years, delivered in bulk afterwards.



# Location: Cerro Pachon, Chile

LSST Site

La Serena

Santiago

## Cerro Pachón – Future site of the LSST



## Leveling of El Peñón (the summit of Cerro Pachón)







*First Stone Ceremony, April 14th 2015*





*LSST Site, April 14th 2015*



May 2018



*LSST Site, May 2018*



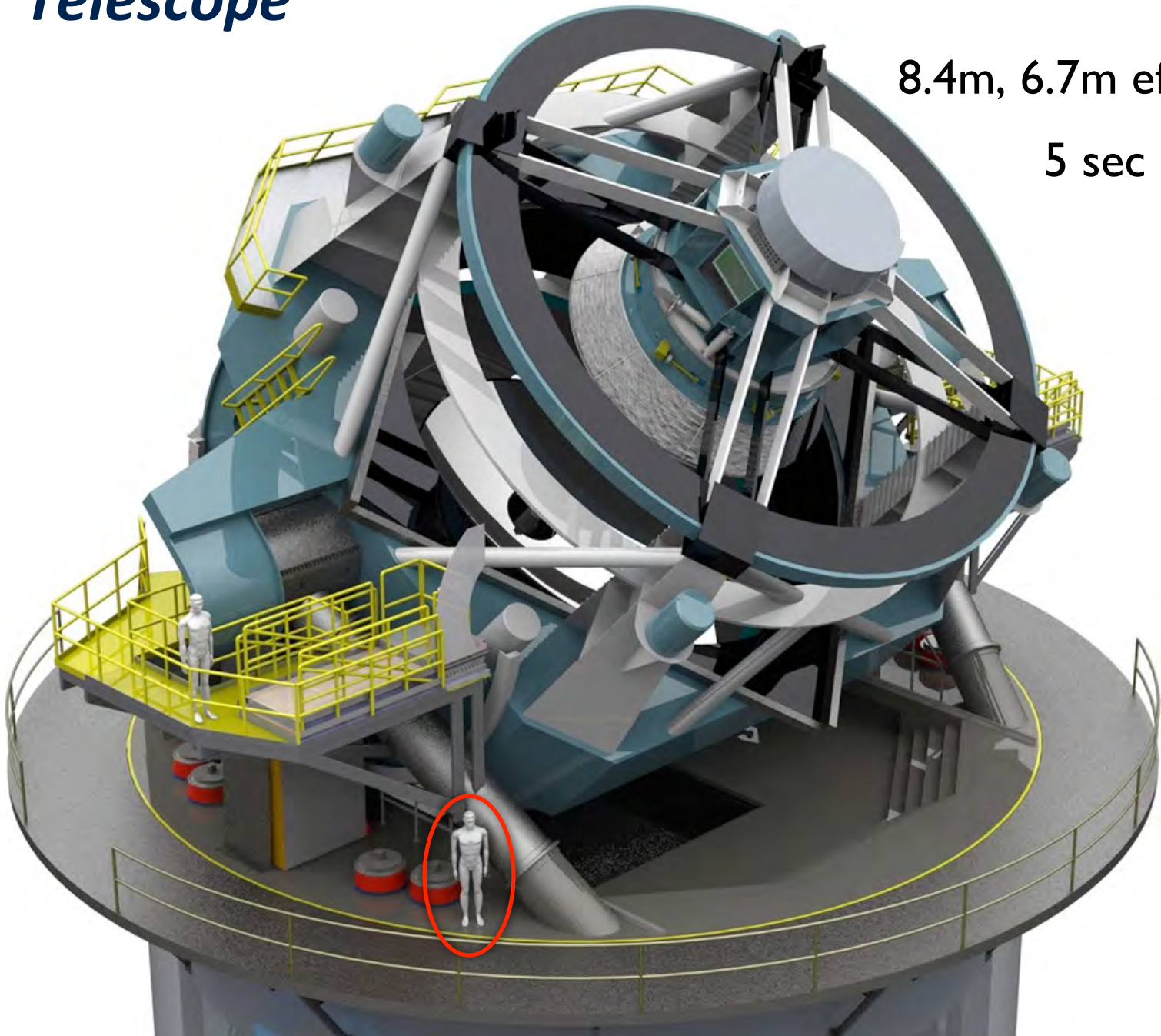
*First Light: 2020*



# *LSST Telescope*

8.4m, 6.7m effective

5 sec slew &  
settle



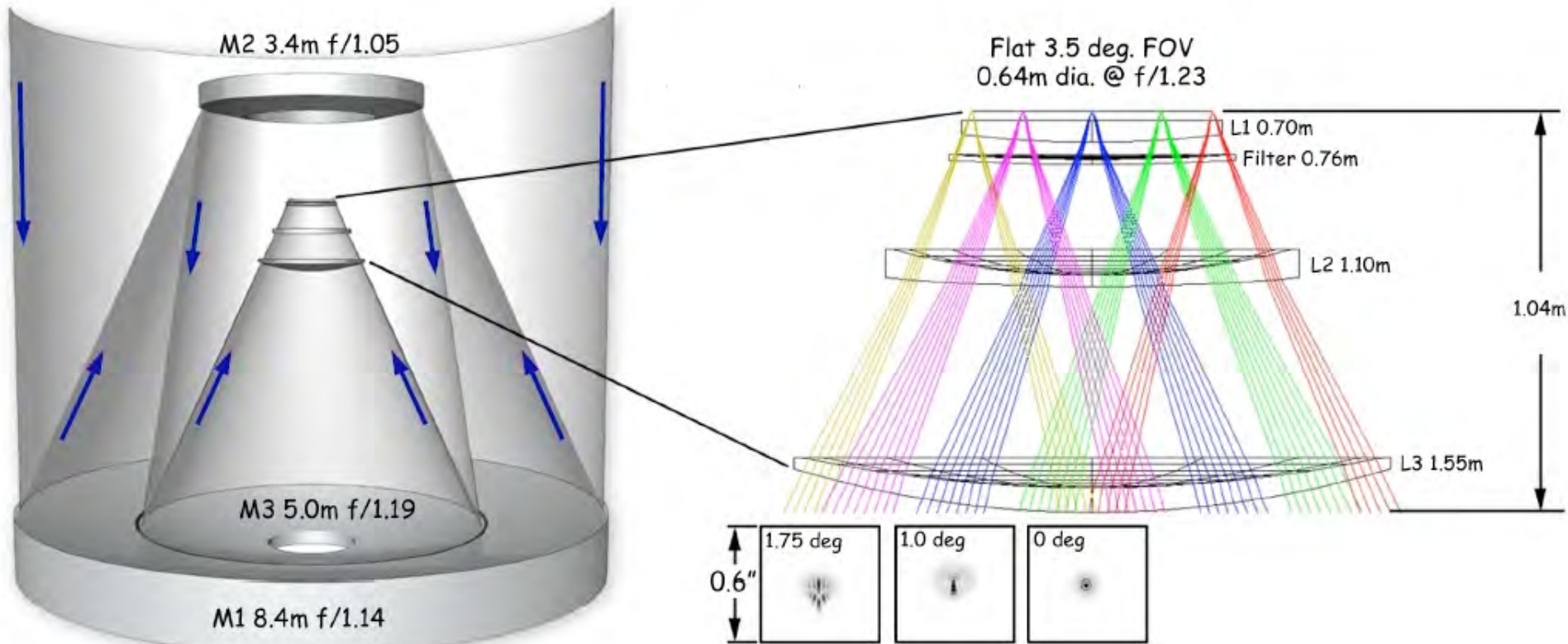




Telescope and Mount Assembly @ Asturfeito in Spain



# Optical Design



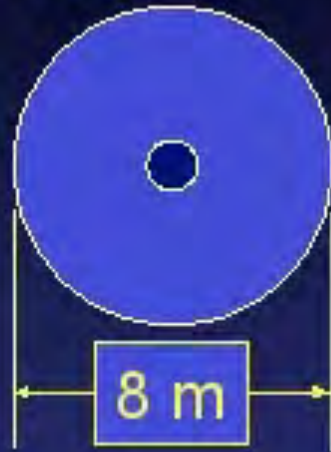
Three-mirror design (Paul-Baker system)  
enables large field of view with excellent image quality:  
delivered image quality is dominated by atmospheric seeing

# The field-of-view comparison: Gemini vs. LSST

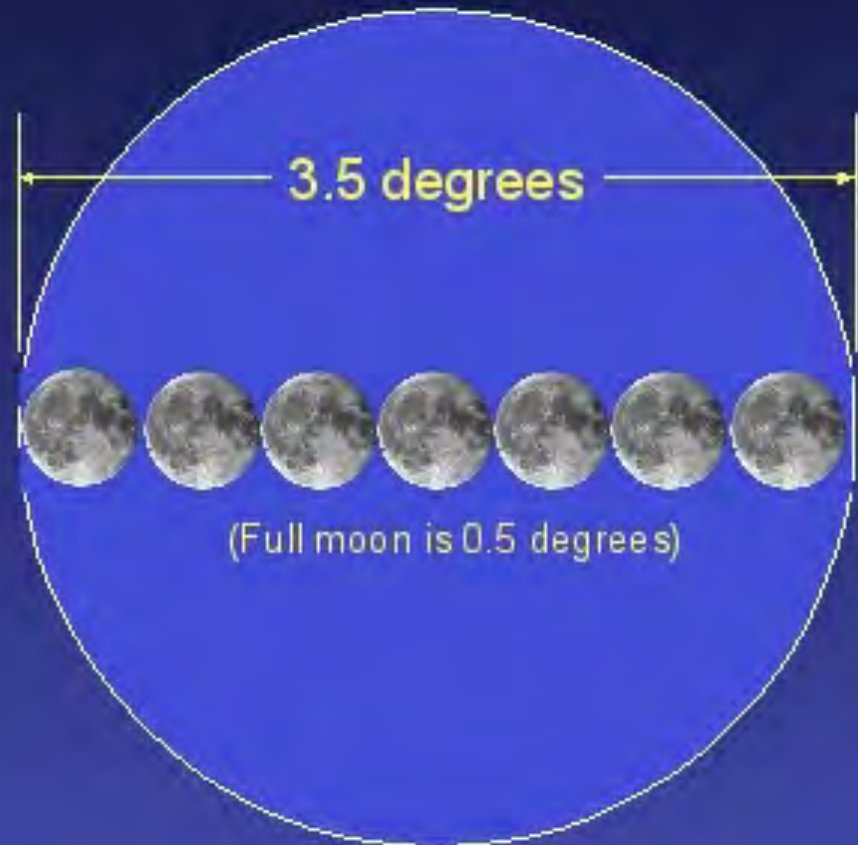


Gemini South Telescope

Primary Mirror Diameter

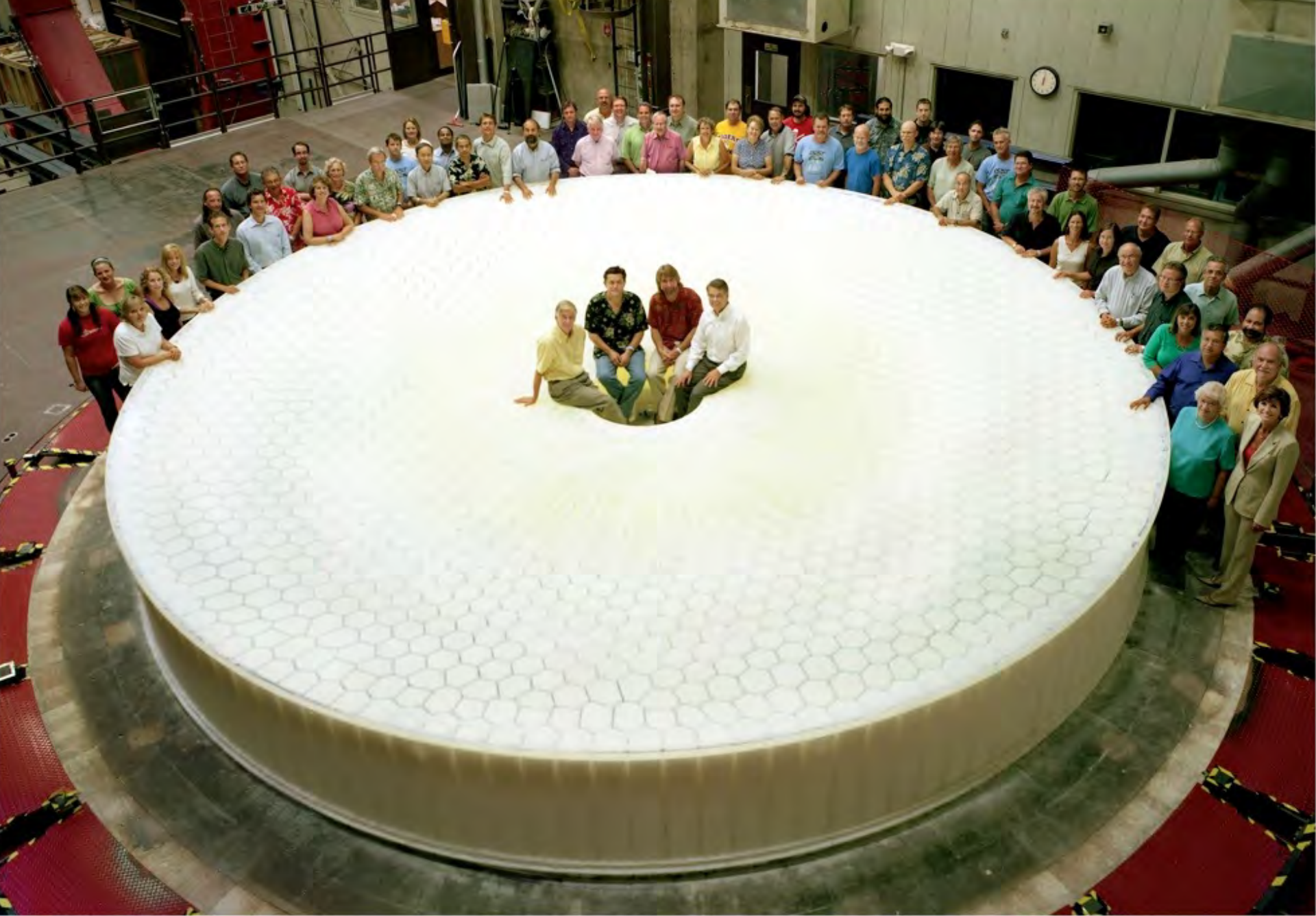


Field of View



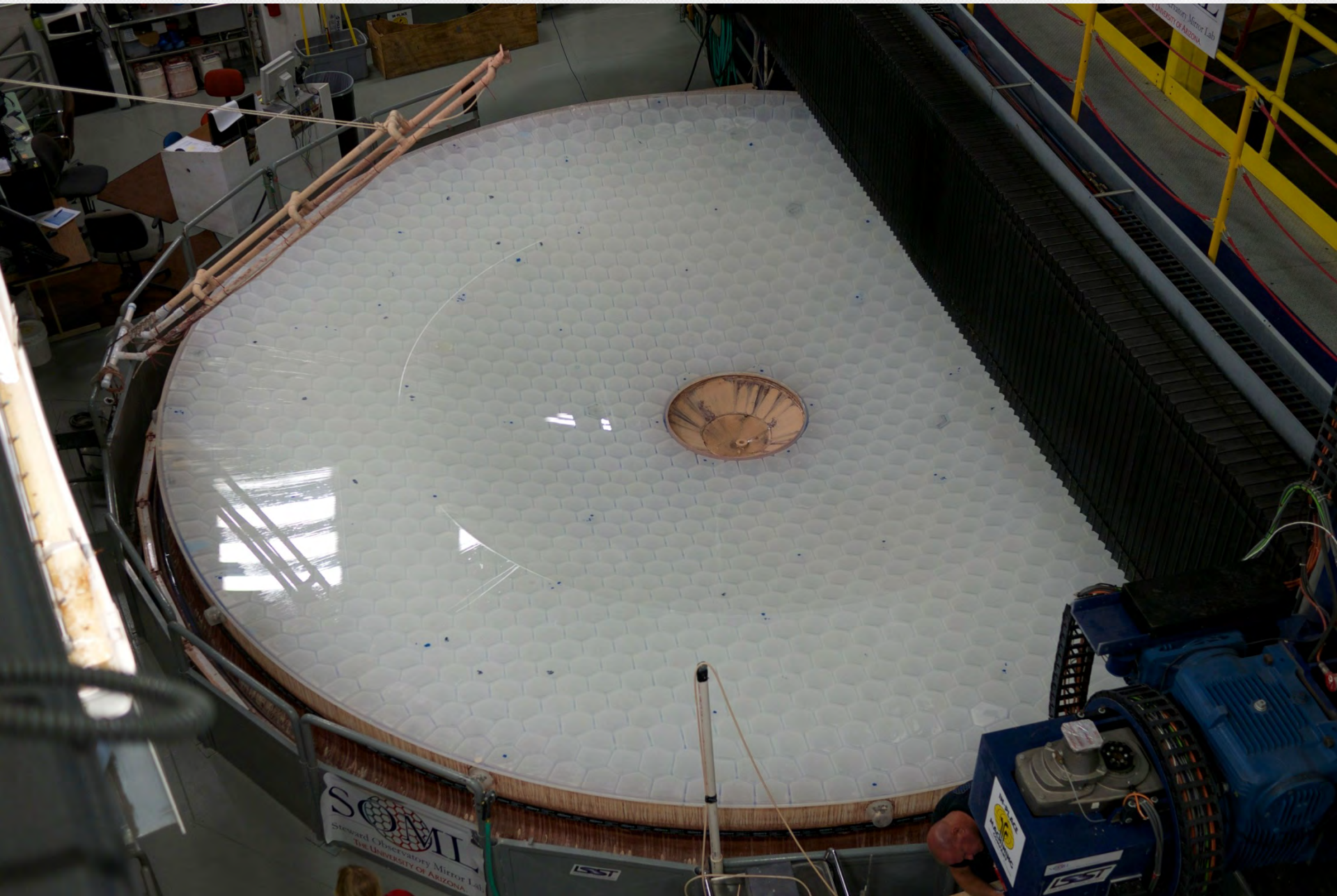
LSST





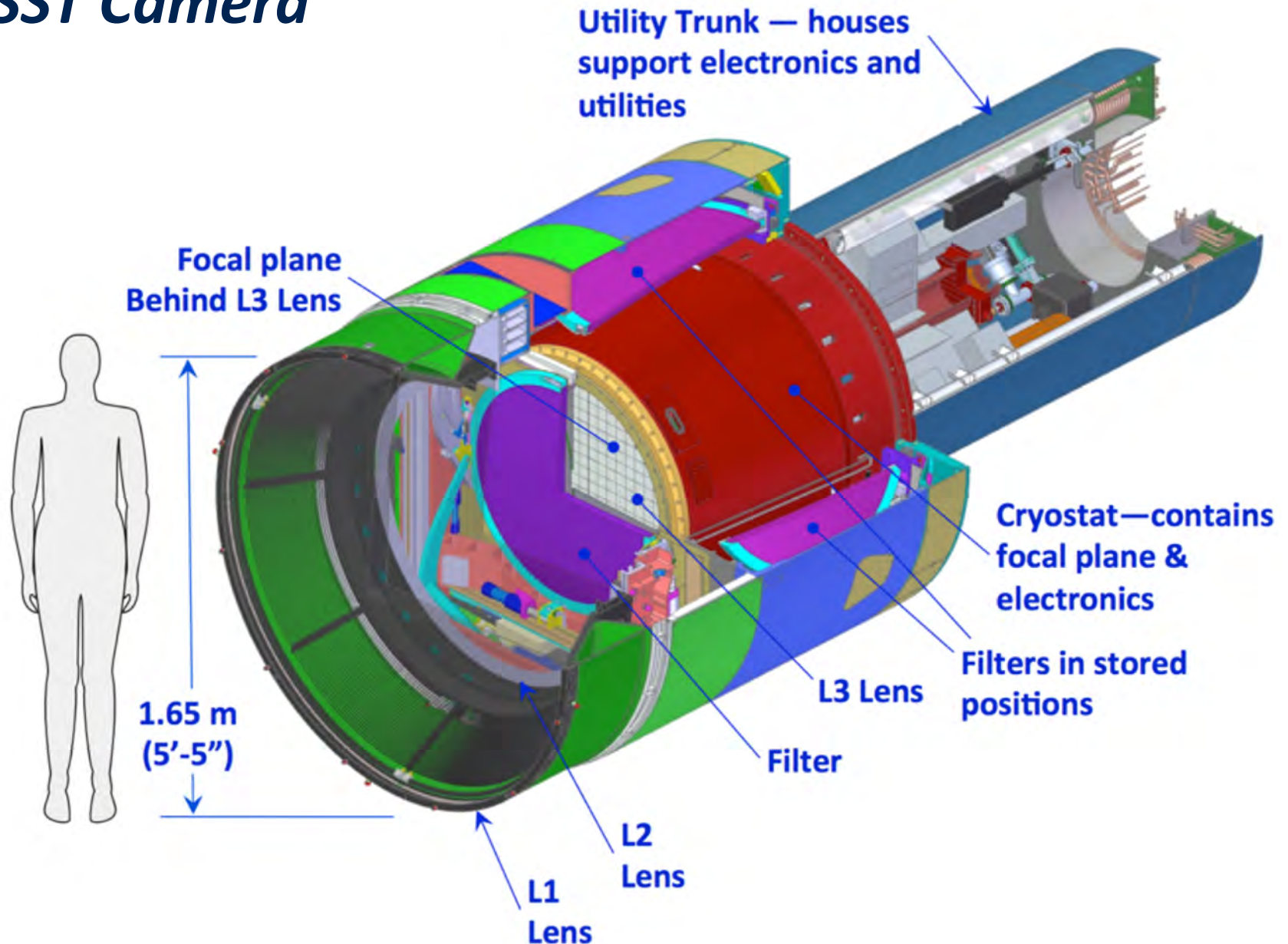


Done!





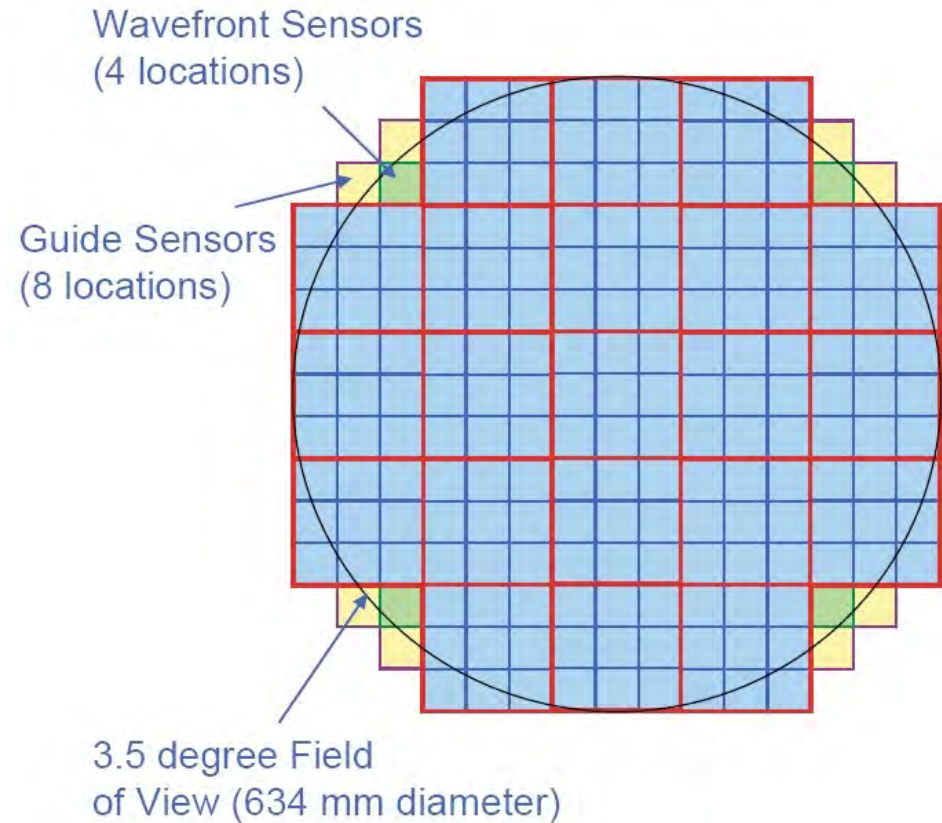
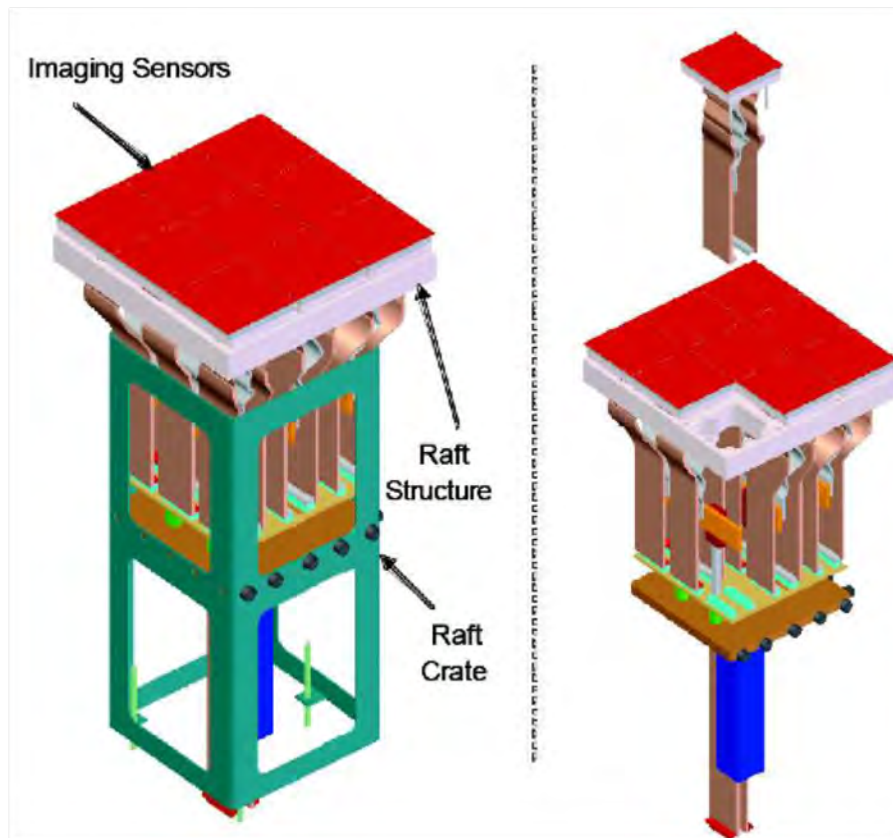
# LSST Camera



A very large (and massive) camera: 2800 kg, 3.2 Gpix

# LSST Camera

189 CCDs with 16Mpix each, assembled in 3x3 “raft” modules



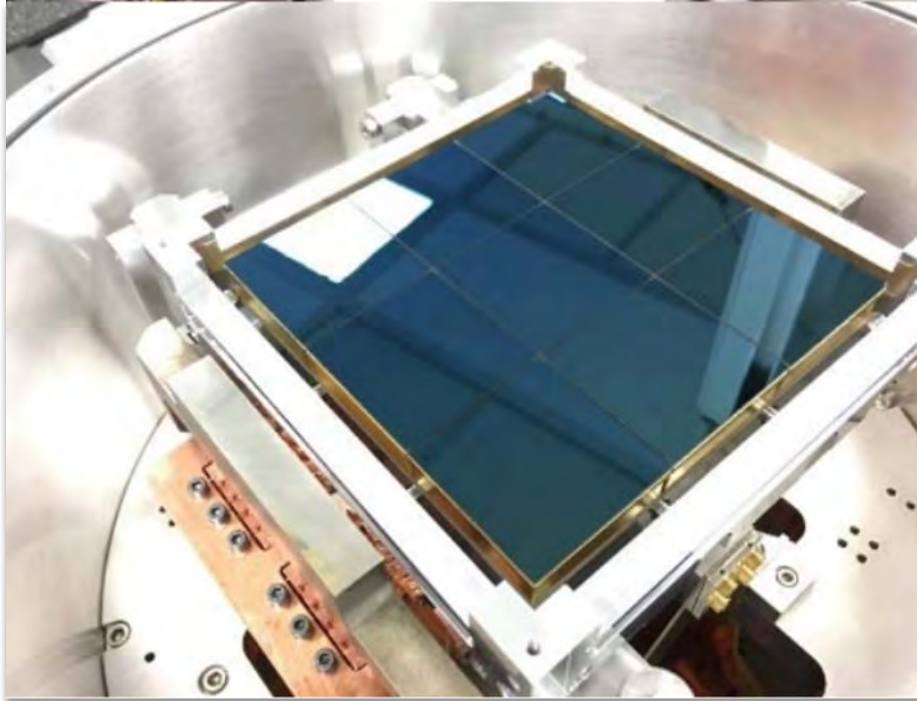
4k x 4k CCDs

3 x 3 CCD “rafts”

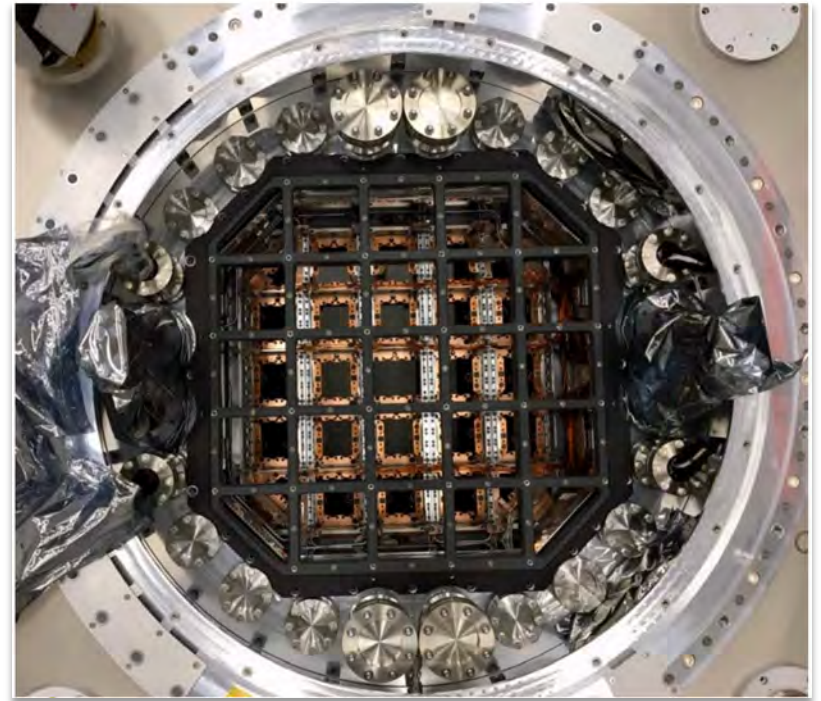
21 Raft Camera



# *Almost there!*



- 264 Science and Science Reserve Sensors delivered
- Need 208



- 12 Rafts SLAC Accepted
- 1.6 Gpixels Ready!
- Many in progress
- Need: 21 Science Rafts and 4 corner Rafts

# SDSS vs. LSST comparison: $LSST = d(SDSS)/dt$ , LSST=SuperSDSS

3 arcmin  
is 1/10  
of the full  
Moon's  
diameter

3x3 arcmin, gri

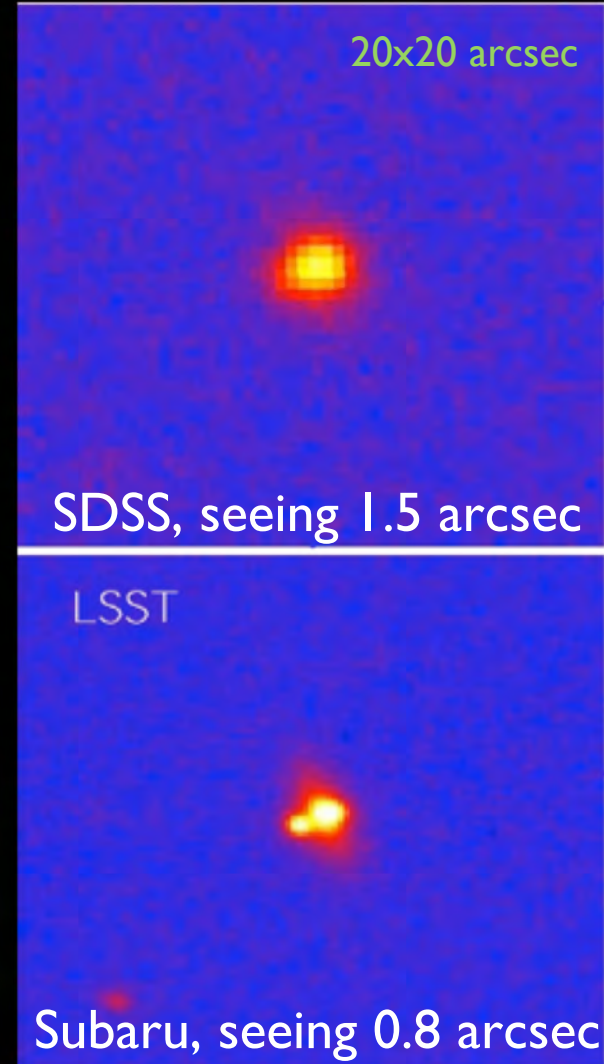


SDSS



Deep Lens Survey ( $r \sim 26$ )

20x20 arcsec; lensed SDSS quasar  
(SDSS J1332+0347, Morokuma et al. 2007)



20x20 arcsec

SDSS, seeing 1.5 arcsec

LSST

Subaru, seeing 0.8 arcsec

→  
(almost)  
like LSST  
depth (but  
tiny area)



# LSST Operations: Sites and Data Flows



## Satellite Processing Center

(CC-IN2P3, Lyon, France)

Data Release Production (50%)



## Archive Site

### Archive Center

Alert Production

Data Release Production (50%)

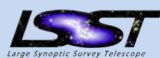
EPO Infrastructure

Long-term Storage (copy 2)

### Data Access Center

Data Access and User Services

## HQ Site



Science Operations

Observatory Management

Education and Public Outreach

## Chilean Sites

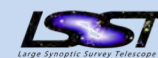
Telescope and Camera

Data Acquisition

Crosstalk Correction

Long-term storage (copy 1)

Chilean DAC Entry-point





- A stream of ~10 million time-domain events per night, detected and transmitted to event distribution networks within 60 seconds of observation.
- A catalog of orbits for ~6 million bodies in the Solar System.
- A catalog of ~37 billion objects (20B galaxies, 17B stars), ~7 trillion observations (“sources”), and ~30 trillion measurements (“forced sources”), produced annually, accessible through online databases.
- Reduced single-epoch, deep co-added images.
- User-produced added-value data products (deep KBO/NEO catalogs, variable star classifications, shear maps, ...)

Prompt

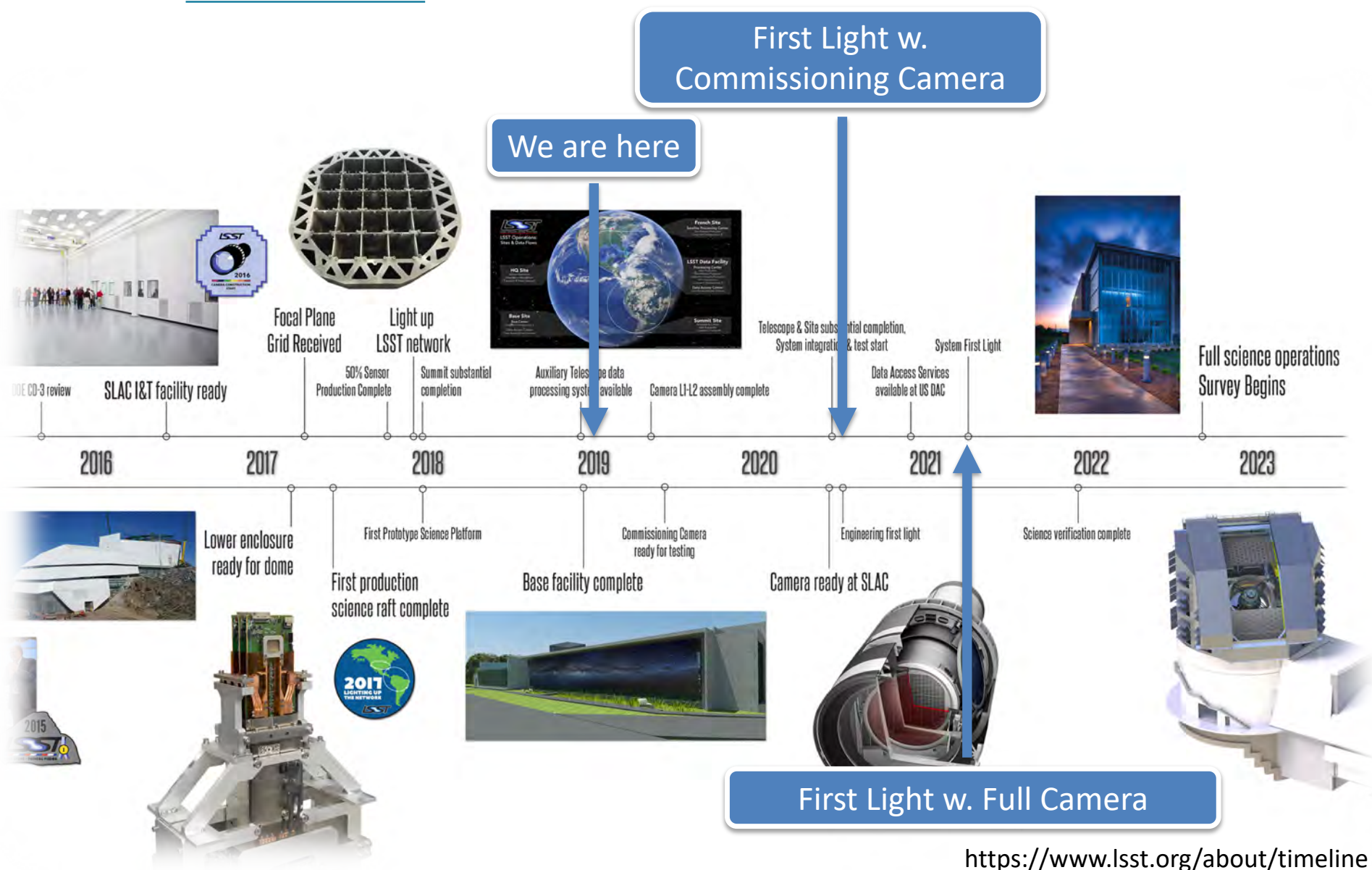
Data Rel.

User  
generated

For more details, see the “**Data Products Definition Document**”, <http://ls.st/lse-163>



# LSST Is Almost Here!



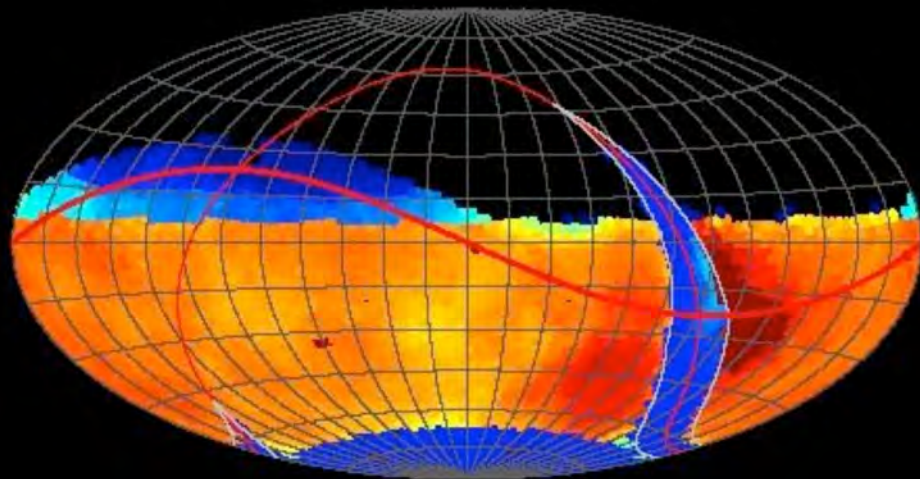
<https://www.lsst.org/about/timeline>

# LSST Science



# Basic idea behind LSST: **a uniform sky survey**

- 90% of time will be spent on a uniform survey: every 3-4 nights, the whole observable sky will be scanned twice per night
- after 10 years, half of the sky will be imaged about 1000 times (in 6 bandpasses, ugrizy): a digital color movie of the sky
- ~100 PB of data: about a billion 16 Mpix images, enabling **measurements for 40 billion objects!**



0 50 100 150 200  
acquired number of visits: r

## **LSST in one sentence:**

An optical/near-IR survey of half the sky in ugrizy bands to  $r \sim 27.5$  (36 nJy) based on 825 visits over a 10-year period: **deep wide fast.**

**Left:** a 10-year simulation of LSST survey: the number of visits in the r band (Aitoff projection of eq. coordinates)

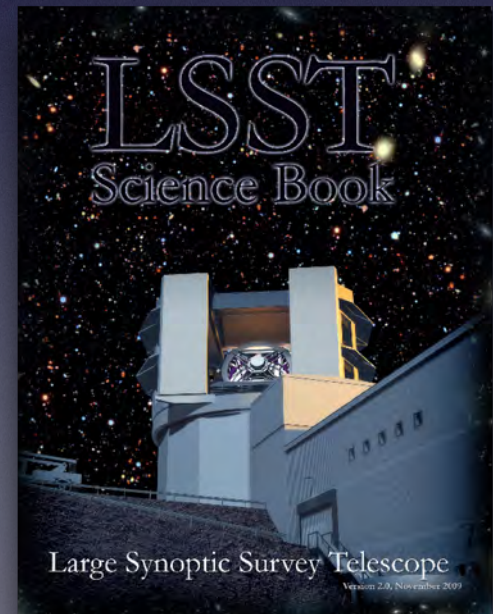
# Key Science Themes Enabled by LSST

- Dark matter, dark energy, cosmology  
(spatial distribution of galaxies, gravitational lensing, supernovae, quasars)
- Time domain (cosmic explosions, variable stars)
- The Solar System structure (asteroids)
- The Milky Way structure (stars)

## LSST Science Book: [arXiv:0912.0201](https://arxiv.org/abs/0912.0201)

Summarizes LSST hardware, software, and observing plans, science enabled by LSST, and educational and outreach opportunities

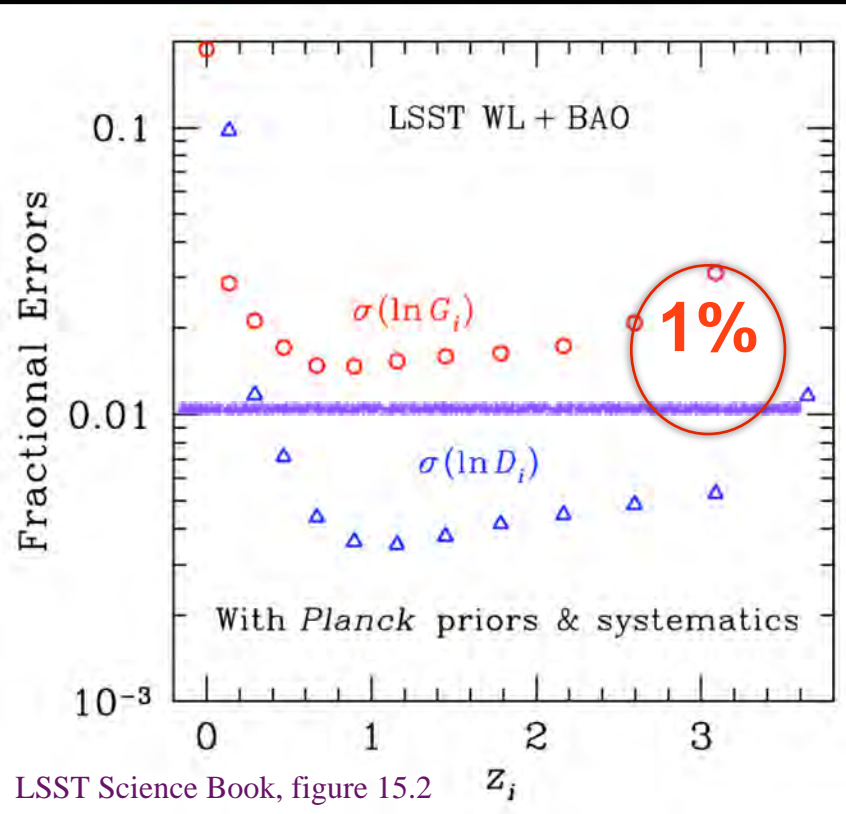
**245 authors, 15 chapters, 600 pages**



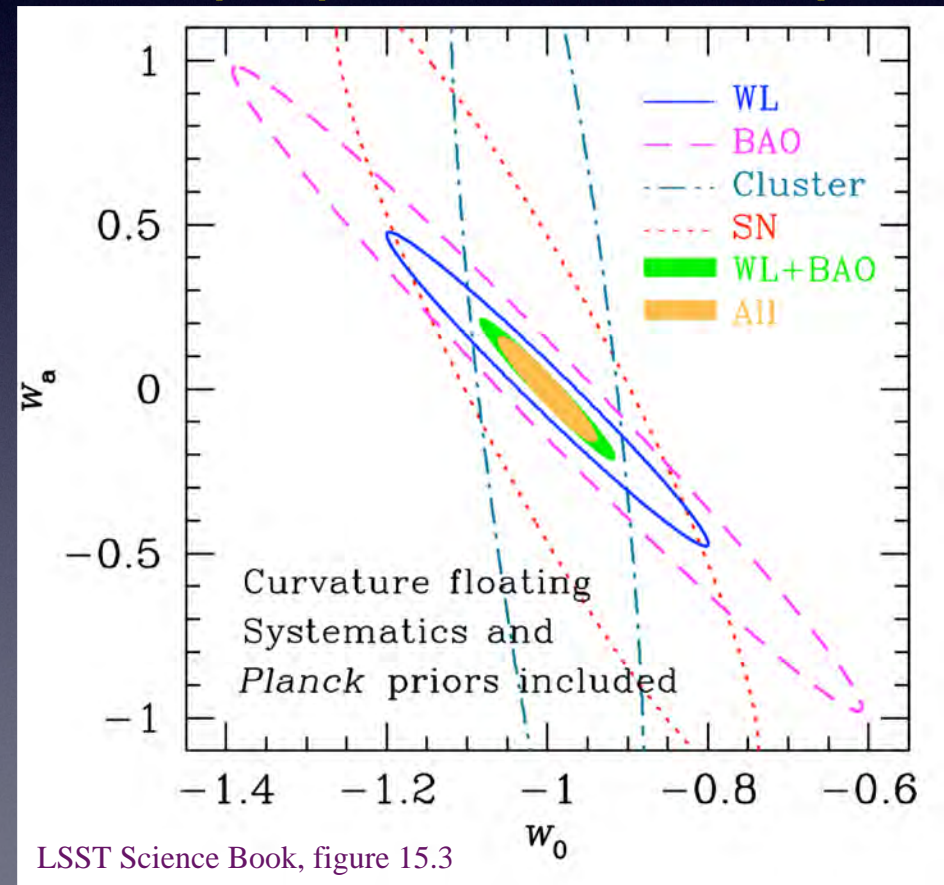


# Cosmology with LSST: high precision measurements

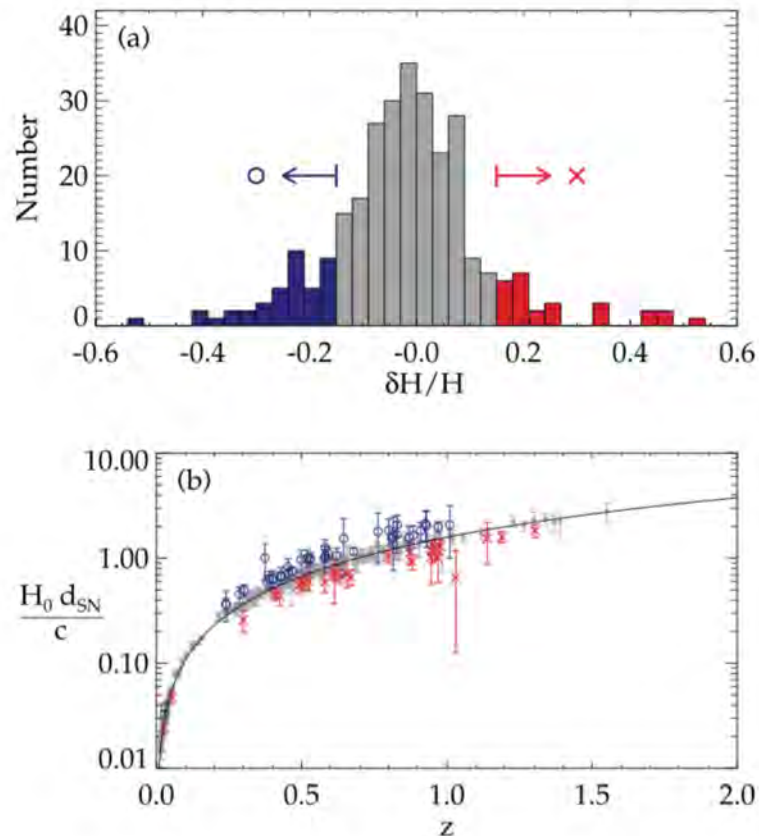
- Measuring distances,  $H(z)$ , and growth of structure,  $G(z)$ , with a percent accuracy for  $0.5 < z < 3$
- Multiple probes are the key!



LSST is designed to be a Stage IV Dark Energy Experiment (DETF)

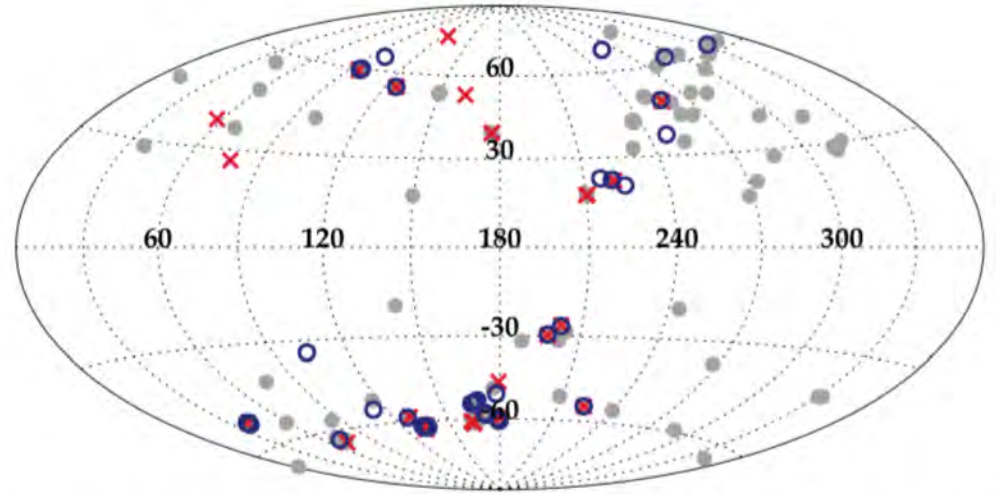


# Cosmology with LSST SNe: is the cosmic acceleration the same in all directions?



Cooke & Lynden-Bell (2009, MNRAS 401, 1409)

Is there spatial structure in the SNe distance modulus residuals for the concordance model?



**Figure 1.** A projection of the spatial distribution of the Union SNe Ia sample in Galactic coordinates. Note the relative uniformity of the points, except around the Galactic plane. The symbols correspond to those in Fig. 2, and are explained in Section 3.1.

- Even a single supernova represents a cosmological measurement!
- LSST will obtain light curves for several million Type Ia supernovae!



# Extragalactic astronomy: low surface brightness objects

Sloan Digital Sky Survey



3x3 arcmin, gri

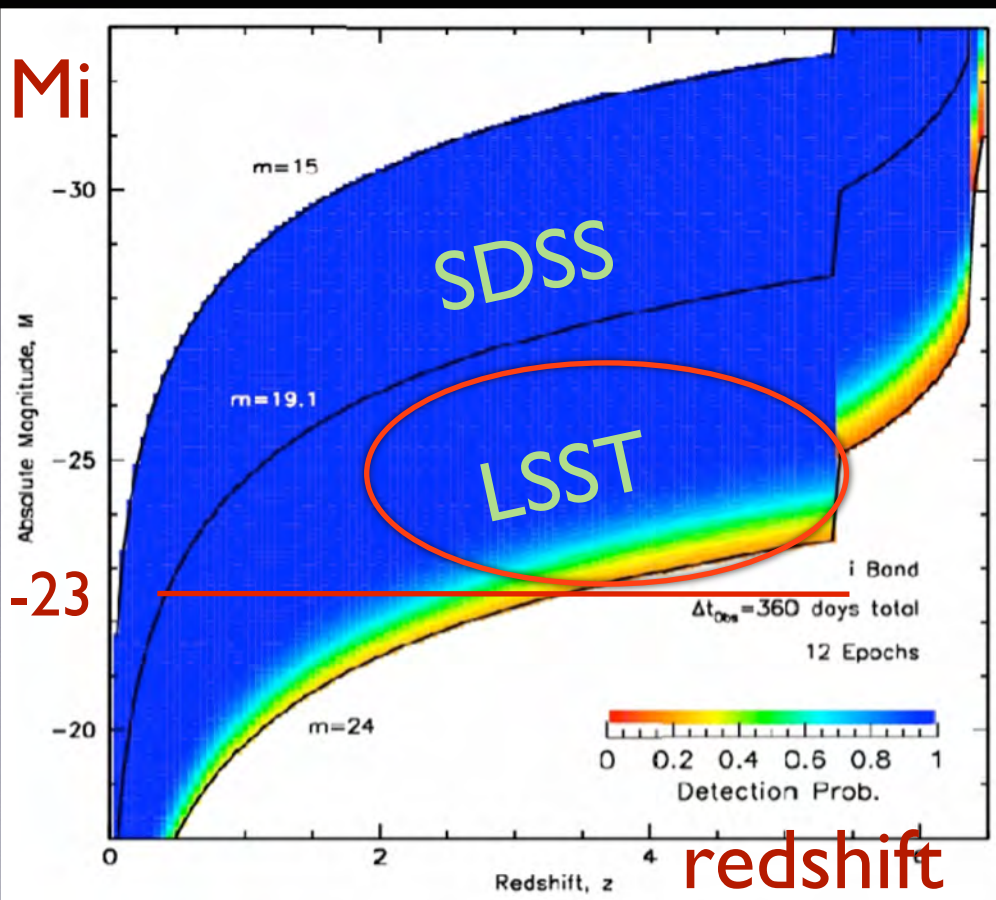


MUSYC: (almost) like LSST  
depth (but tiny area)

$r \sim 26$

Gawiser et al

# Extragalactic astronomy: Quasars



**Top:** absolute magnitude vs. redshift diagram for quasars

Today: ~100s of quasars with  $6 < z < 7.5$

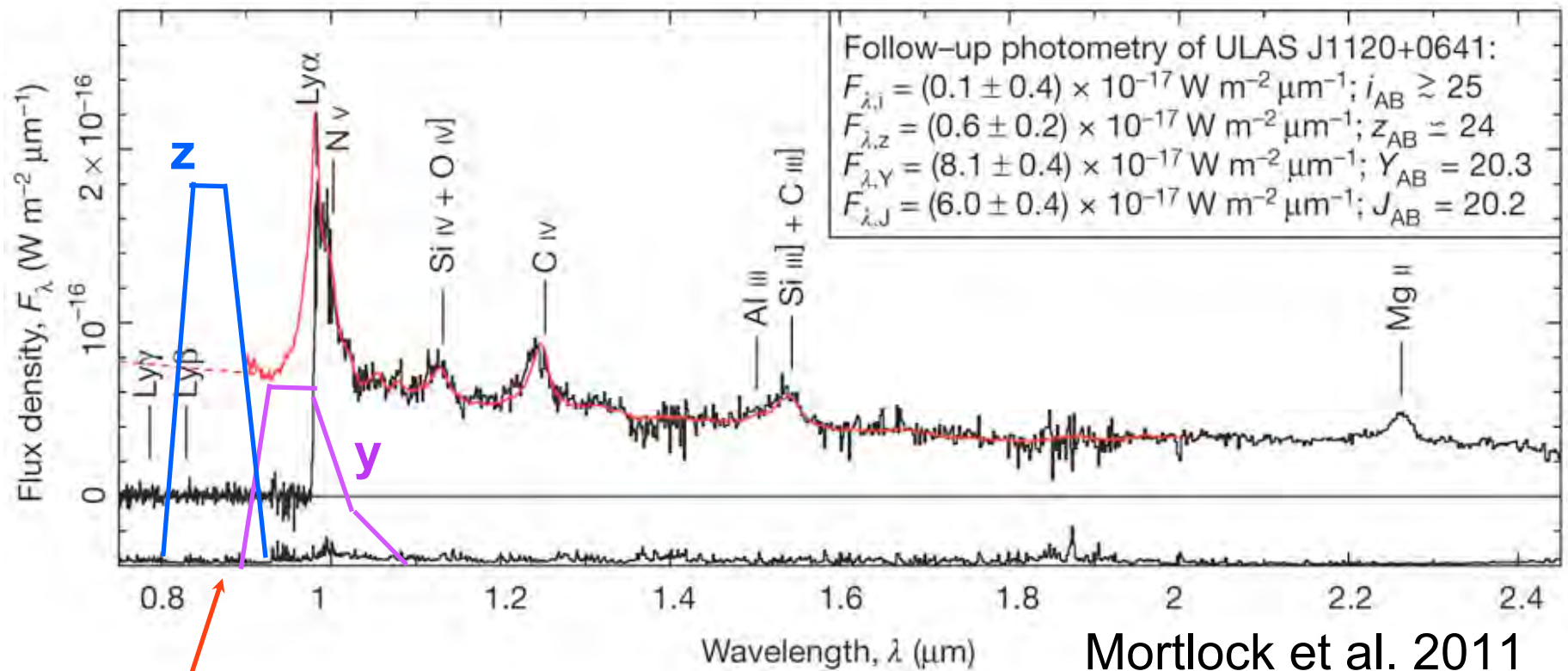
**Reionization studies!**

**LSST will detect ~10,000 quasars with  $6 < z < 7.5$ !**

- About 10 million quasars will be discovered using variability, colors, and the lack of proper motions
- The sample will include  $M_i = -23$  objects even at redshifts beyond 3
- Quasar variability studies will be based on millions of light curves with 1000 observations over 10 yrs



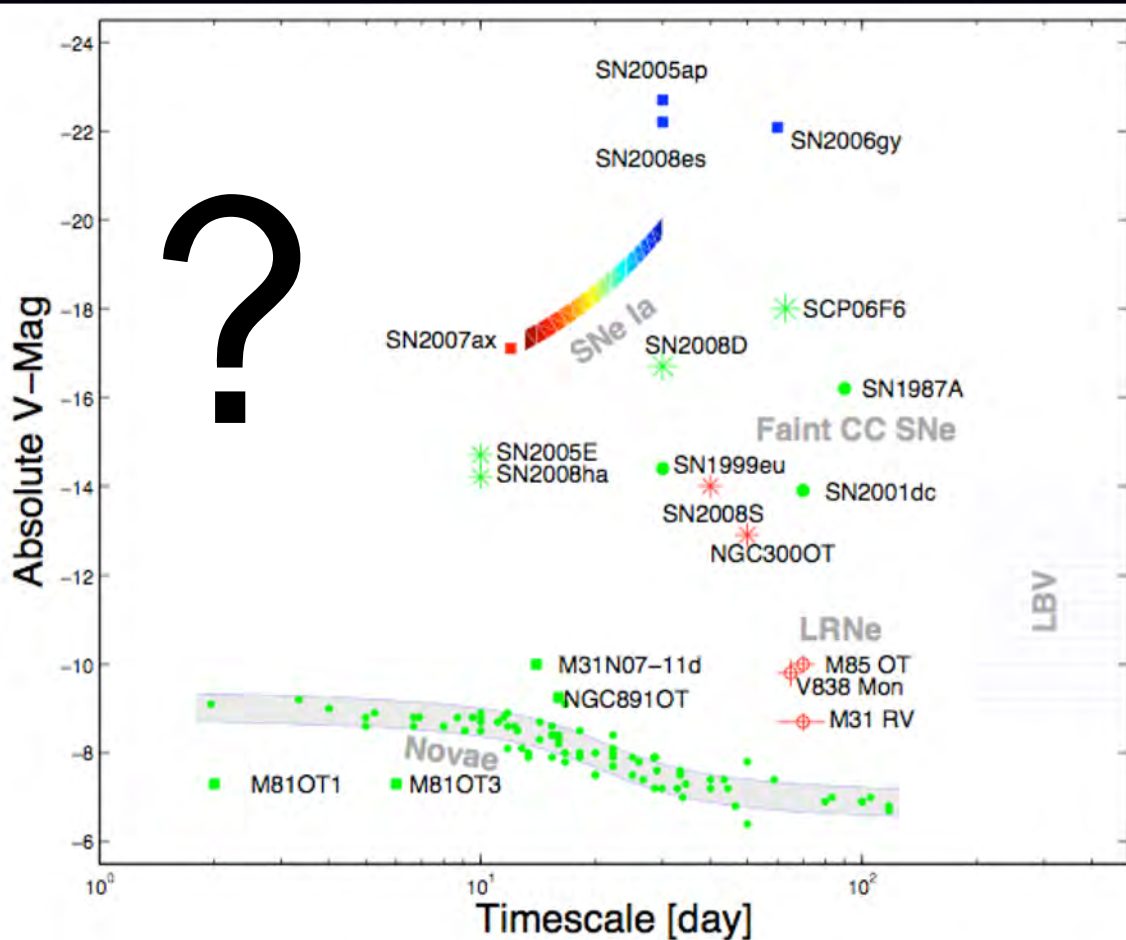
# The Highest Redshift Quasar at $z=7.085$ from UKIDSS



**Such a quasar would be detected by LSST as a z-band dropout**  
(multi-epoch data will greatly help with false positives)

**LSST should discover about 1,000 quasars with  $z > 7$**   
**Today (2016): one quasar with  $z > 7$**

**Time Domain:** objects changing in time  
**positions:** asteroids and stellar proper motions  
**brightness:** cosmic explosions and variable stars



LSST will extend  
time-volume space  
a thousand times  
over current  
surveys (new  
classes of object?)

known unknowns  
unknown unknowns

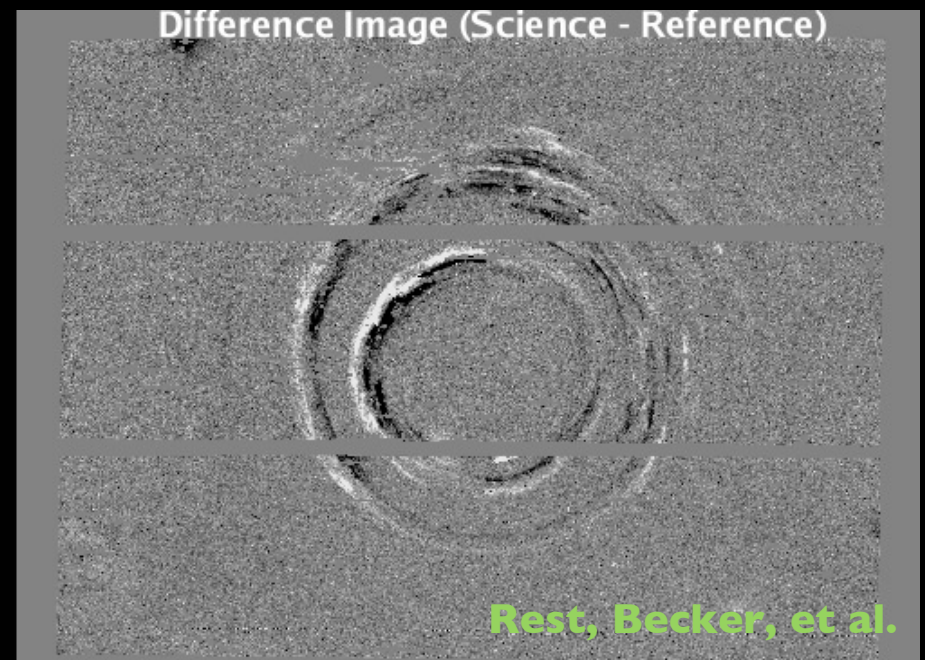
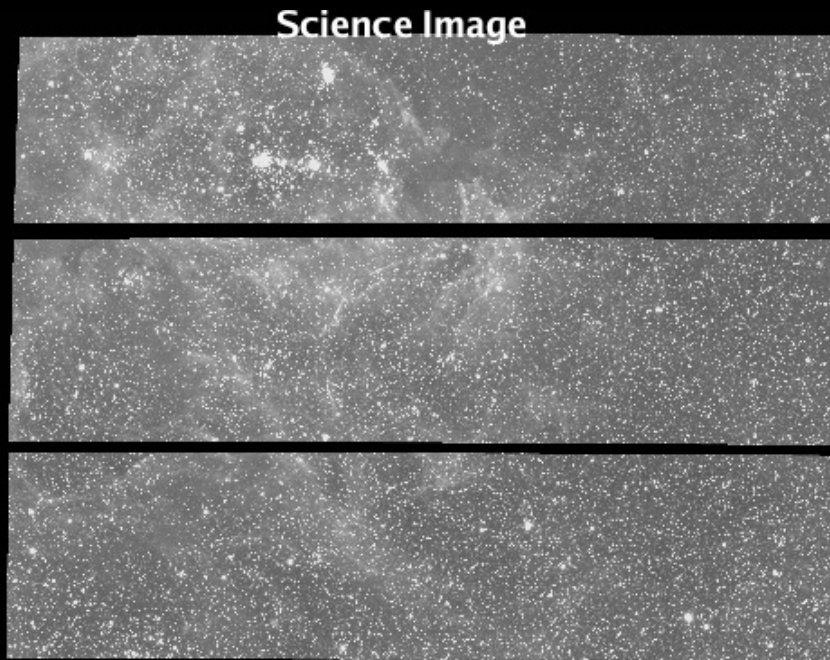
Note: There will be as many  
variable stars from LSST, as  
all stars from SDSS !

FIG. 29.— The phase space of cosmic explosive and eruptive transients as represented by their absolute  $V$  band peak brightness and the event timescale (adapted from Kulkarni et al. 2007).



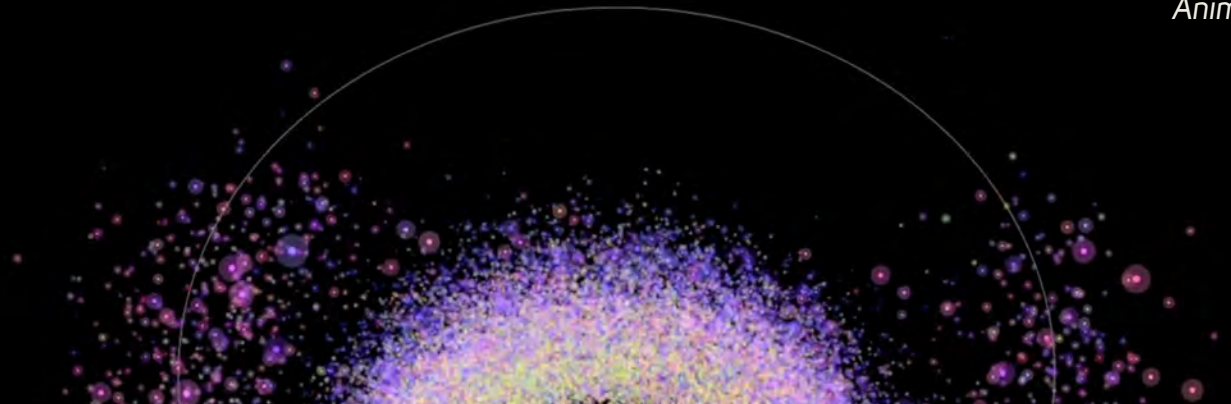
**Time Domain:** objects changing in time  
**positions:** asteroids and stellar proper motions  
**brightness:** cosmic explosions and variable stars

Not only point sources - echo of a supernova explosion:



# Census of the Solar System

Animation: SDSS Asteroids  
(Alex Parker, SwRI)



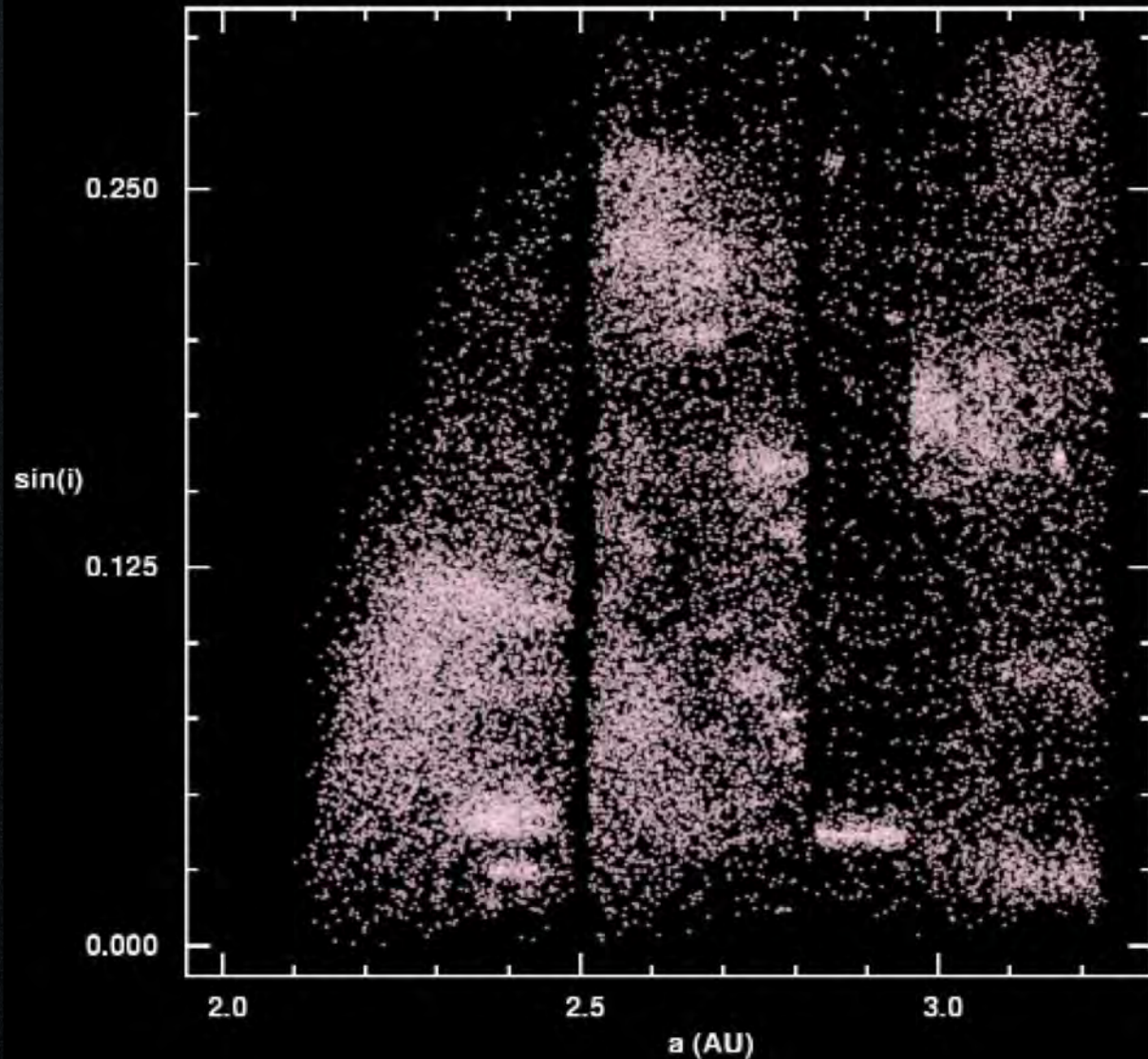
	Currently Known*	LSST Discoveries**	Median number of observations+	Observational arc length+
Near Earth Objects (NEOs)	14,500	100,000	(D>250m) 60	6.0 years
Main Belt Asteroids (MBAs)	650,000	5,500,000	(D>500m) 200	8.5 years
Jupiter Trojans	6000	280,000	(D>2km) 300	8.7 years
TransNeptunian Objects (TNOs) + Scattered Disk Objects (SDOS)	2000	40,000	(D>200km) 450	8.5 years

Estimates: Lynne Jones et al.



- **Large statistical samples of asteroids and comets**
- Serendipitous discoveries of rare events or objects
  - Asteroid collisions (P/2010 A2)
  - Retrograde TNO (2008 KV42)
- Discover new, incoming comets even before they become active
- Model shapes of asteroids from measurements of their brightness
- Discover links between different populations
  - How are NEOs and Main Belt asteroids related?
  - Are irregular satellites actually captured TNOs?
- Expand our knowledge of all small bodies to provide a better understanding of the formation and evolution of our Solar System

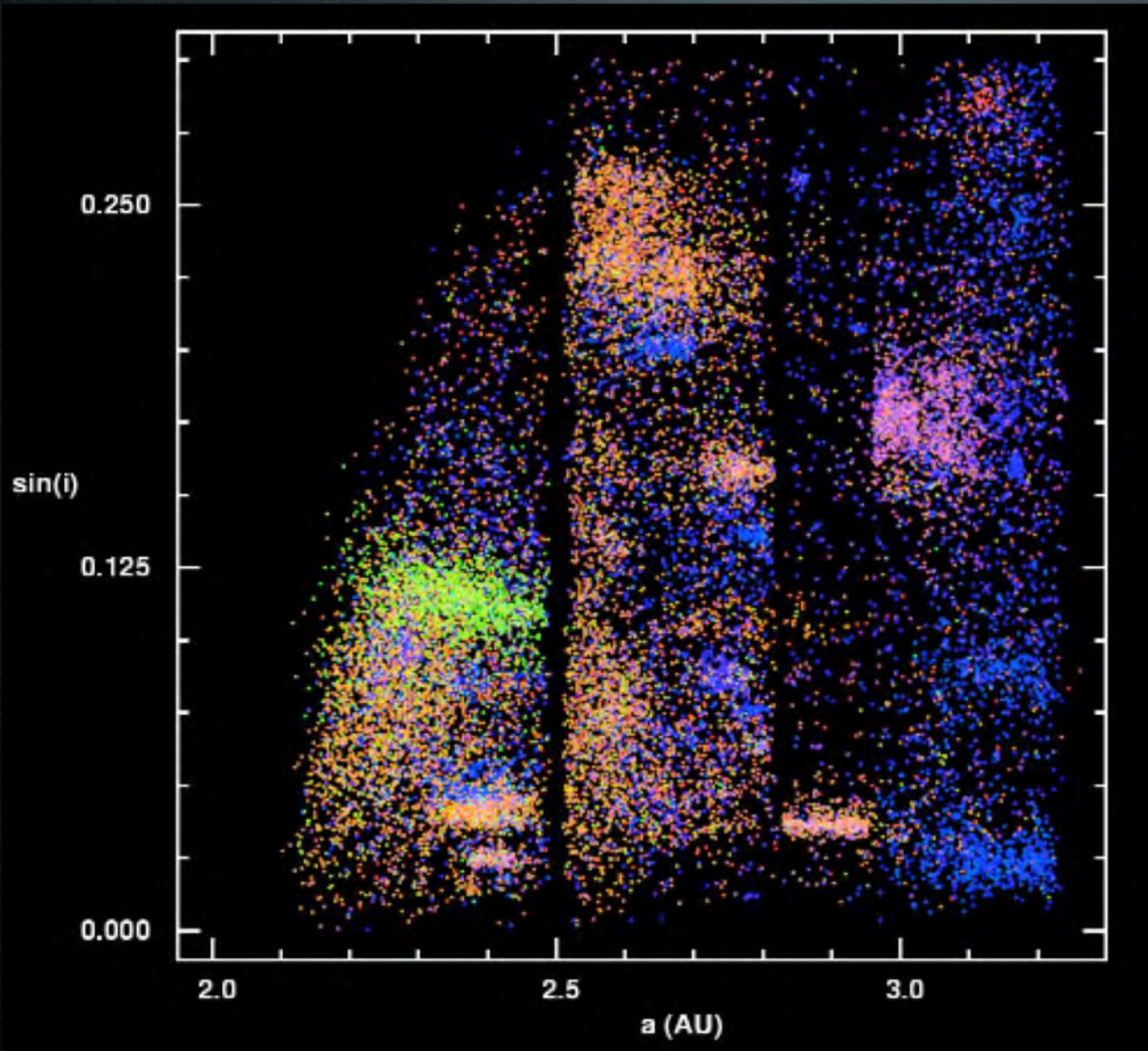
# Main-belt Inventory



30,000  
Asteroids with  
SDSS colors and  
proper  
orbital elements  
(Ivezic, Juric, Lupton 2002)



# Main-belt Inventory



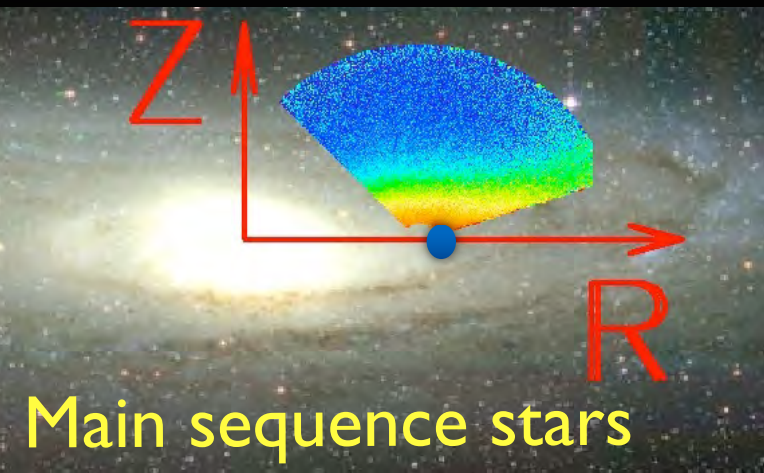
30,000  
Asteroids with  
SDSS colors and  
proper  
orbital elements  
(Ivezic, Juric, Lupton 2002)

Color-coded with  
SDSS colors

Colors help with the definition of asteroid families.  
LSST will also provide color light curves!

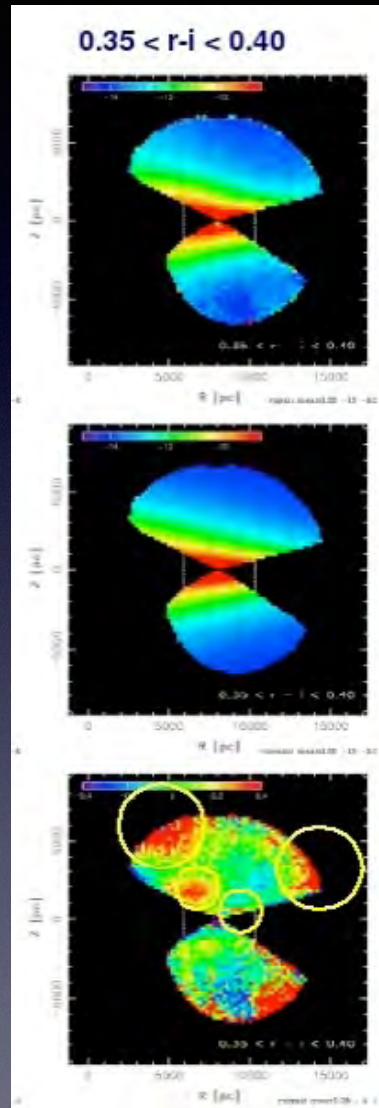
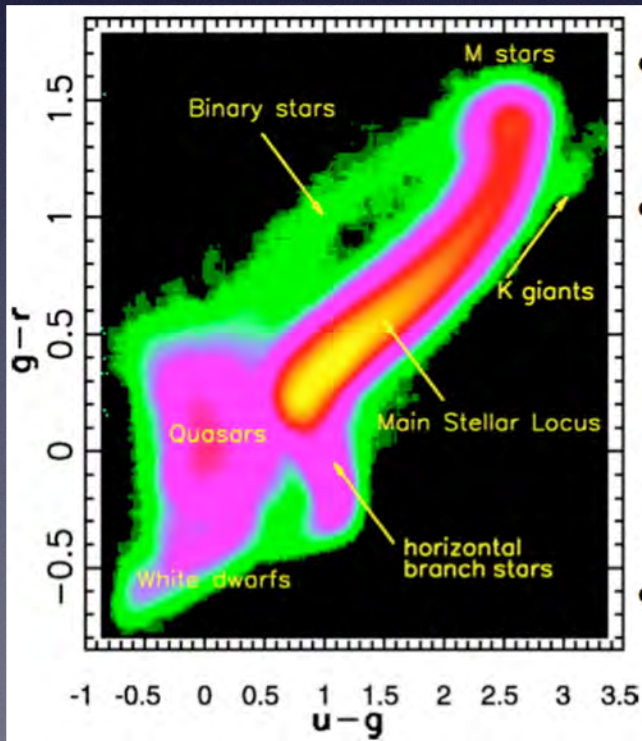


# The Milky Way structure: 20 billion stars, time domain massive statistical studies!



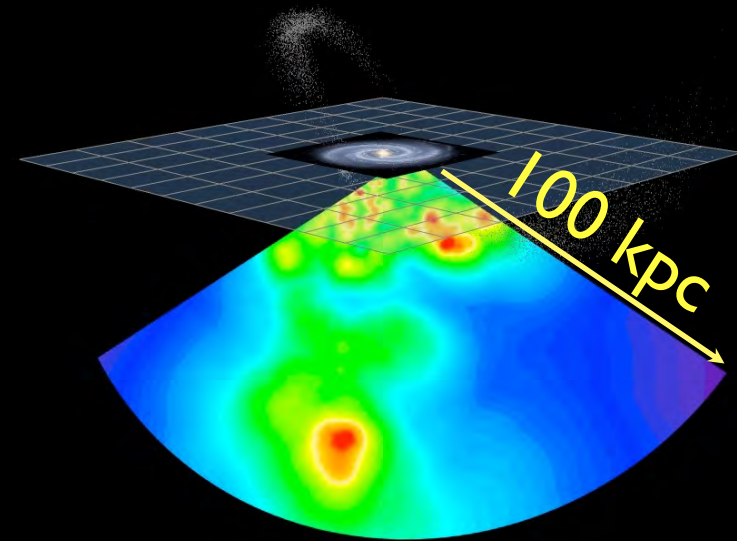
Main sequence stars

Distance and  $[Fe/H]$ :



Juric et al. (2008)

Compared to SDSS:  
LSST can “see” about  
40 times more stars,  
10 times further away  
and over twice as  
large sky area

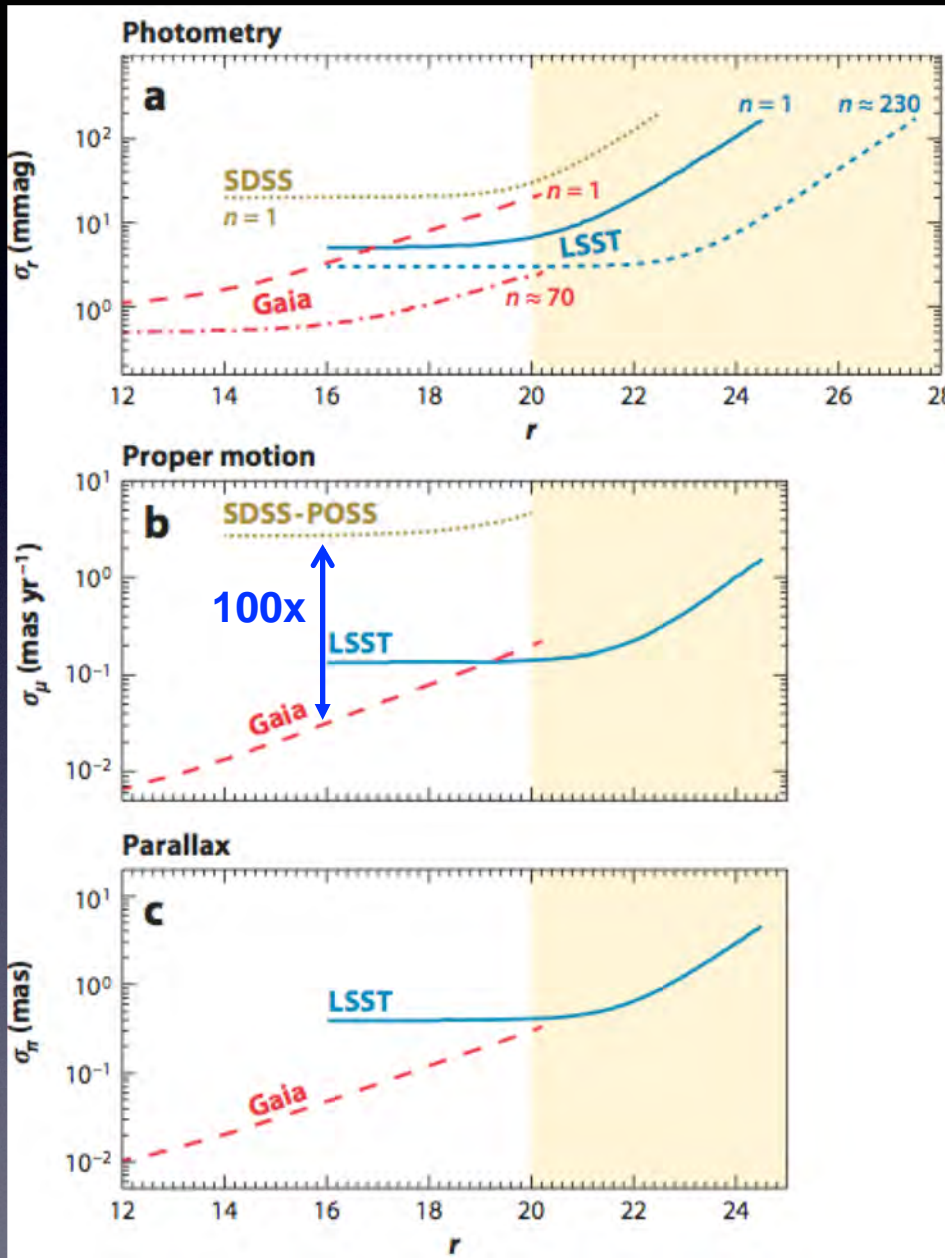


SDSS RR Lyrae

Sesar et al. (2009)



# Gaia vs. LSST comparison



- **Gaia:** excellent astrometry (and photometry), but only to  $r < 20$

- **LSST:** photometry to  $r < 27.5$  and time resolved measurements to  $r < 24.5$

- Complementarity of the two surveys: photometric, proper motion and trigonometric parallax errors are similar around  $r=20$

The Milky Way disk “belongs” to Gaia, LSST will be excellent for the halo (plus very faint and/or very red sources, such as white dwarfs and LT(Y) dwarfs).

The large blue circle: the  $\sim 400$  kpc limit of future LSST studies based on RR Lyrae

The large red circle: the  $\sim 100$  kpc limit of future LSST studies based on main-sequence stars (and the current limit)

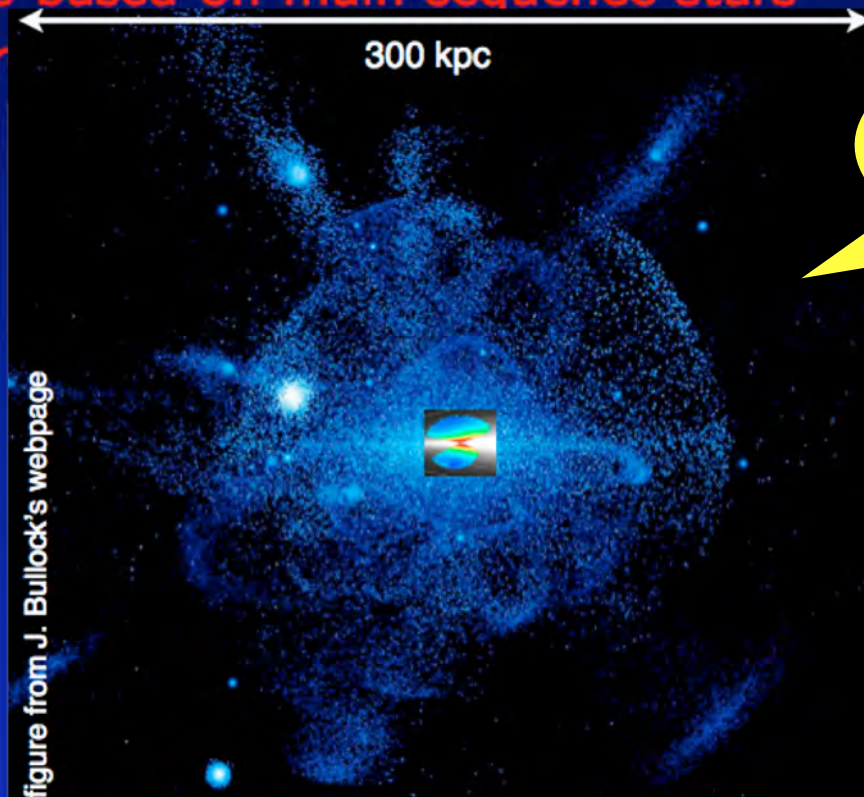


figure from J. Bullock's webpage

LSST limit for RR Lyrae: 400 kpc

200 million stars from LSST!

The small insert:  
 $\sim 10$  kpc limit of SDSS and future Gaia studies for kinematic &  $[Fe/H]$  mapping with MS stars



# Doing Science With LSST

# LSST Data Products

- A stream of ~10 million time-domain events per night, detected and transmitted to event distribution networks within 60 seconds of observation.
- A catalog of orbits for ~6 million bodies in the Solar System.
- A catalog of ~37 billion objects (20B galaxies, 17B stars), ~7 trillion observations (“sources”), and ~30 trillion measurements (“forced sources”), produced annually, accessible through online databases.
- Reduced single-epoch, deep co-added images.
- User-produced added-value data products (deep KBO/NEO catalogs, variable star classifications, shear maps, ...)

Prompt

Data Rel.

User  
generated

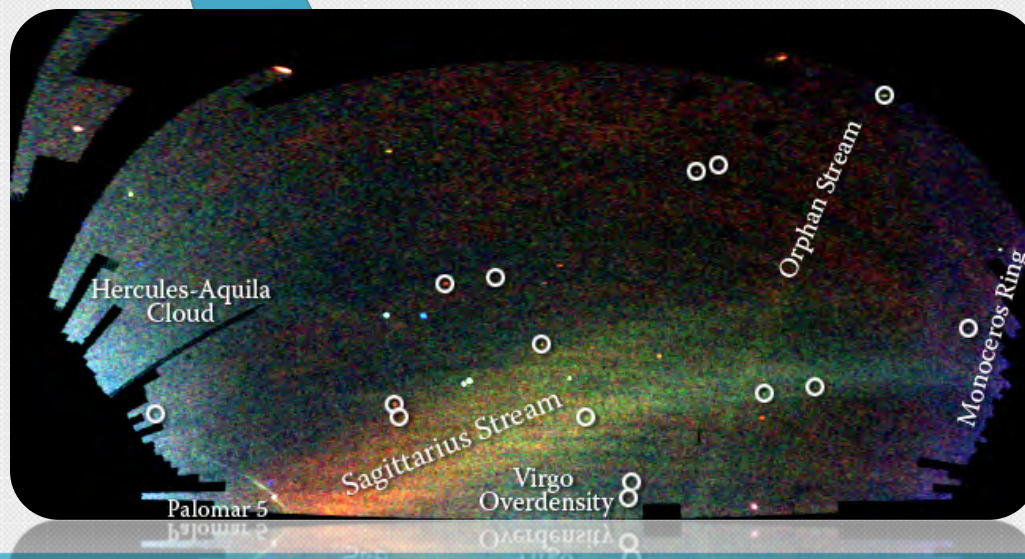
For more details, see the “**Data Products Definition Document**”, <http://ls.st/lse-163>



# Analysis Paradigms: Subset – Download – Analyze

```

7757,301,1.74,186,6,8,13544435186418,26,6361772894736,17,84889,11,16535,0,01654885,0,02145229
7757,301,1.74,187,6,8,12783867355789,26,62724593921,17,37487,17,02873,0,01844421,0,03348013
7757,301,1.74,188,3,8,127323232524192,26,6251199416623,26,1466,21,35297,0,3883744,0,3302762
4288,301,1.39,682,3,24,5161178422385,-1,16579446393527,22,97832,24,3259,0,2672399,0,5248437
4288,301,1.39,683,3,24,5179486515354,-1,17578698922485,22,62952,25,09109,0,1858479,0,6585885
4288,301,1.39,684,6,24,5188443723148,-1,159158816881,23,4247,23,4432,0,06688635,0,1368172
4136,301,1.61,931,6,36,4715522759892,-1,06893938826789,22,71782,23,14112,0,1580147,0,1799857
4136,301,1.61,936,3,36,4717581013136,-1,1378448767726,22,81863,23,88123,0,1742272,0,3266685
4136,301,1.61,937,3,36,47175812434391,-1,13784497192974,22,81147,23,87586,0,1734457,0,3247895
4288,301,1.48,311,3,24,6839281339872,-1,23631696217547,21,7862,21,67521,0,05694364,0,06316777
4288,301,1.48,312,3,24,68486881592246,-1,21784918327087,26,38287,21,64576,0,02972161,0,04622911
4288,301,1.48,313,6,24,6848216698377,-1,08292727289886,24,52163,25,72778,0,68427,0,5471938
5598,301,1.61,792,3,351,787958113487,6,14573538435867,22,43574,22,83793,0,1125768,0,2867745
5598,301,1.61,793,3,351,787958113414,6,14573538316193,22,43573,23,77753,0,1174541,0,2741905
5598,301,1.61,794,6,351,787349187439,6,14481613145227,24,4781,24,8587,0,4675894,0,4845864
5598,301,1.48,527,3,12,8768018016488,-3,32677418219699,22,18116,23,27577,0,326546,0,2278169
2699,301,1.48,528,6,12,8778827529666,-3,32913243326258,22,12757,23,79366,0,1215217,0,3481472
2699,301,1.48,529,3,12,8832778187538,-3,52539818226738,22,29741,23,32088,0,1377919,0,2253349
94,301,1.38,279,6,348,514768659138,-0,84398883870374,20,75028,21,13886,0,04838822,0,04644733
94,301,1.38,280,6,348,525793546678,-0,965218498356883,24,23321,26,67633,0,8858552,0,7762702
94,301,1.38,281,6,348,53257887691,-1,02365835629542,21,10688,21,15248,0,66685774,0,04672186
4288,301,1.76,766,3,38,0899167388738,-1,25189466335681,22,57854,22,91144,0,1887321,0,1584752
4288,301,1.76,767,3,38,0899962733195,-1,14314301954175,21,95419,23,78533,0,1894794,0,4162939
4288,301,1.76,768,6,38,0899156247815,-1,13236548812718,23,64196,22,91776,0,1537483,0,282232
7937,301,1.84,354,3,5,59332226478183,26,6130638818974,23,61619,23,76798,0,7888025,0,5886199
7937,301,1.84,355,3,5,58984744314193,26,6157353187021,21,27889,22,13992,0,98888753,0,1498587
7937,301,1.84,356,6,5,59897889167375,26,5783184153732,24,52393,26,33231,1,068187,0,6528299
3996,301,1.81,615,6,216,223187682876,11,947228647793,19,59735,21,8716,0,8224876,0,83478354
3996,301,1.81,616,3,216,23835672188,12,853224512884,28,23574,21,24497,0,02192515,0,03942162
3996,301,1.81,617,3,216,219378232888,11,9171482188482,22,68762,23,85279,0,1416738,0,1555459
6354,301,1.29,2997,6,332,382758261878,41,2179102291468,28,95895,22,51458,0,04865189,0,1163793
6354,301,1.29,2998,6,332,3557889585822,41,2447881315887,28,78153,23,35575,0,93573929,0,89754679
6354,301,1.29,2999,3,332,38781488452,41,2613114889198,21,68856,22,75476,0,04217186,0,1153268
2986,301,1.59,251,6,127,546395164415,2,57627835997822,24,42951,26,48598,0,6853675,0,9818285
2986,301,1.59,252,6,127,585997787459,2,58267453869978,25,42876,25,86479,0,5991669,0,8493288
2986,301,1.59,253,6,127,588134654789,2,58377339888894,22,25137,23,16562,0,6728882,0,7848868
5598,301,1.31,1254,6,347,389819727191,0,88342418898853,24,31121,25,55253,0,4788885,0,4634795
5598,301,1.31,1255,6,347,382888888886,4,84284817791926,23,21882,25,48826,0,4833483,0,474572
5598,301,1.31,1256,6,347,294399617648,0,88821292496797,24,812,25,48189,0,5272257,0,4741835
    
```



## Data Volumes

	ZTF	LSST
Number of detections	1 trillion	7 trillion
Number of objects	1 billion	37 billion
Nightly alert rate	1 million	10 million
Nightly data rate	1.4 TB	15 TB
Alert latency	< 20 minutes	60 seconds

Science analysis code

~50kb



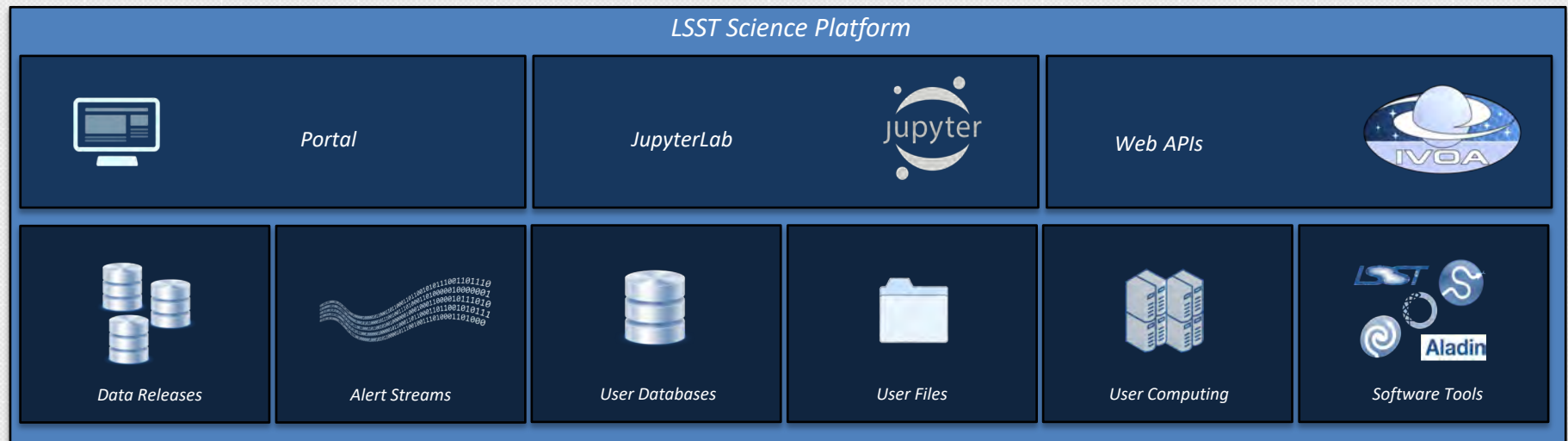
***If the data is big...***

***... bring the code to the data.***

# The LSST Science Platform: Accessing LSST Data and Enabling LSST Science

Juric, Dubois-Felsmann & Ciardi (2016)

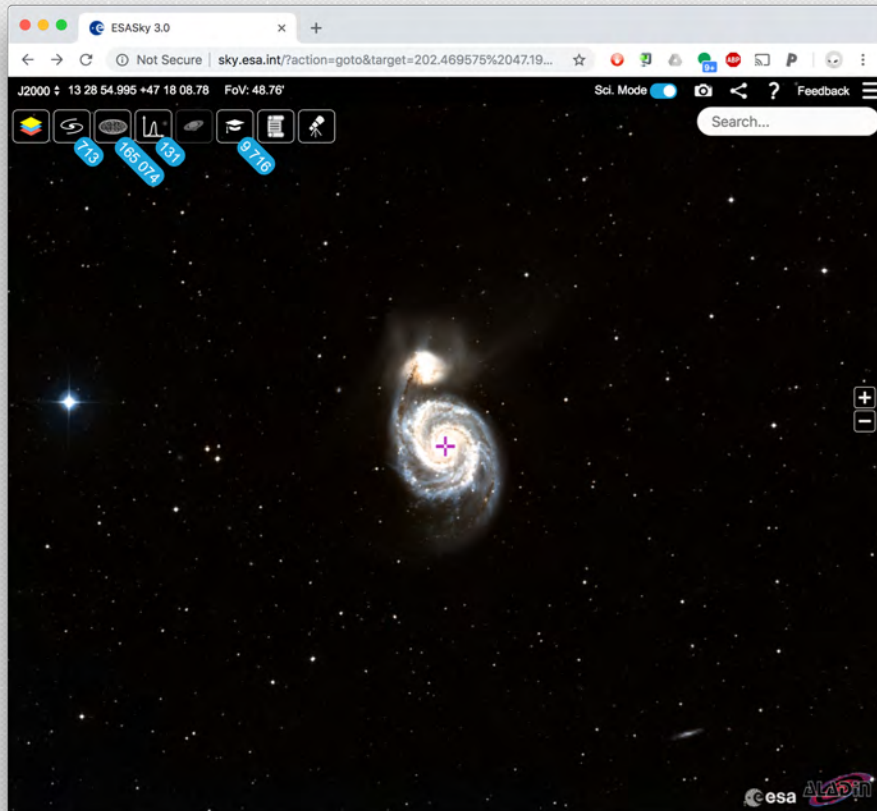
<http://ls.st/lse-319>



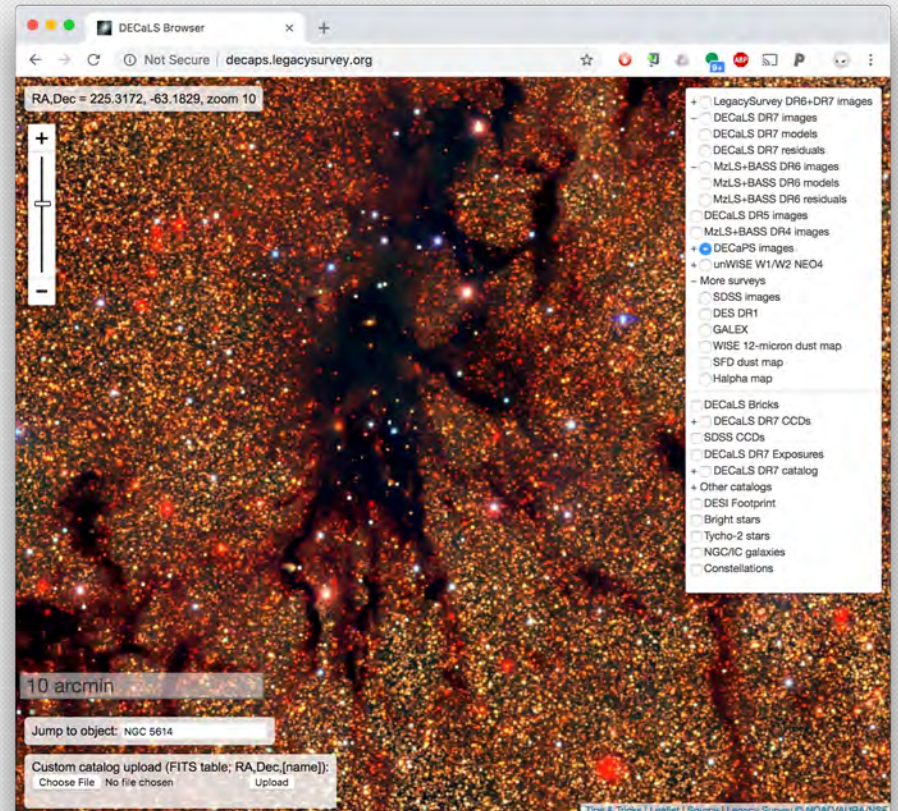
The **LSST Science Platform** is a set of integrated web applications and services deployed at the LSST Data Access Centers (DACs) through which the scientific community will access, visualize, subset and perform next-to-the-data analysis of the data.



# What to expect



<http://sky.esa.int/>



<http://decaps.legacysurvey.org/>

# JupyterLab: Next-to-the-data Analysis

The screenshot displays the JupyterLab interface. On the left, a file browser shows a directory structure with files like 'analysis', 'singlechip\_sample', 'test.ipynb', and 'singlechip.tar.gz'. The main area shows a Jupyter notebook with Python code for astronomical data analysis. The code includes imports for 'last.afw.geom' and 'afw.geom', and uses 'Butler' to retrieve data from a specific observation. A scatter plot is shown, displaying a distribution of points with a central cluster. The bottom part of the notebook shows code for plotting the data, including 'plt.scatter' and 'plt.xlim'.

```
In [41]: butler = datamodels.Butler('singlechip_sample')
exp = butler.get('calexp', visit=410877, ccd=28, filter='r')

In [42]: import last.afw.geom as afw_geom
bbox = afw_geom.Box2D(afw_geom.Point2D(1024, 1024), afw_geom.Extent2D(512, 512))
sources = butler.get('src', visit=410877, ccd=28, filter='r')
overlay_masks(exp, bbox=bbox, sources=sources)

In [ ]: x = sources.getPhysicalFlux()
y = sources.getModelFlux()/sources.getPhysicalFlux()
extend = sources.get('base_classificationExtendedness_value')
x = numpy.array(x)
y = numpy.array(y)
stars = numpy.where(extend==0.0)[0]
galaxies = numpy.where(extend>0.0)[0]
plt.scatter(x[stars], y[stars], alpha=0.3)
plt.scatter(x[galaxies], y[galaxies], c='r', alpha=0.3)
plt.xlim(10000, 10)
plt.ylim(-1, 5)
plt.show()
```

YouTube demo of the LSST JupyterLab Aspect Demo: <http://ls.st/bgt>



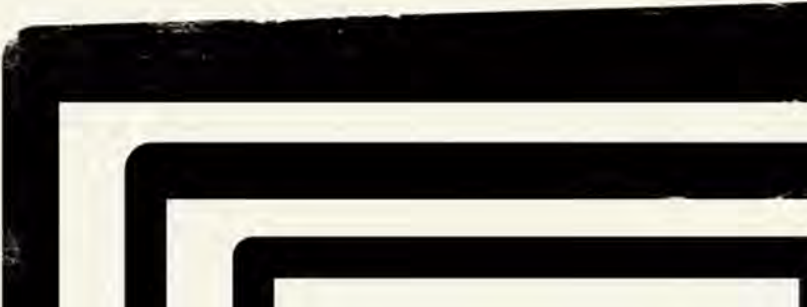
Why Jupyter is data scientists' computational notebook of choice

TOOLBOX • 30 OCTOBER 2018

*An improved architecture and enthusiastic user base are driving uptake of the open-source web tool.*

Jeffrey M. Perkel


Twitter Facebook Email



**Nature Careers**  
@NatureCareers

Follow

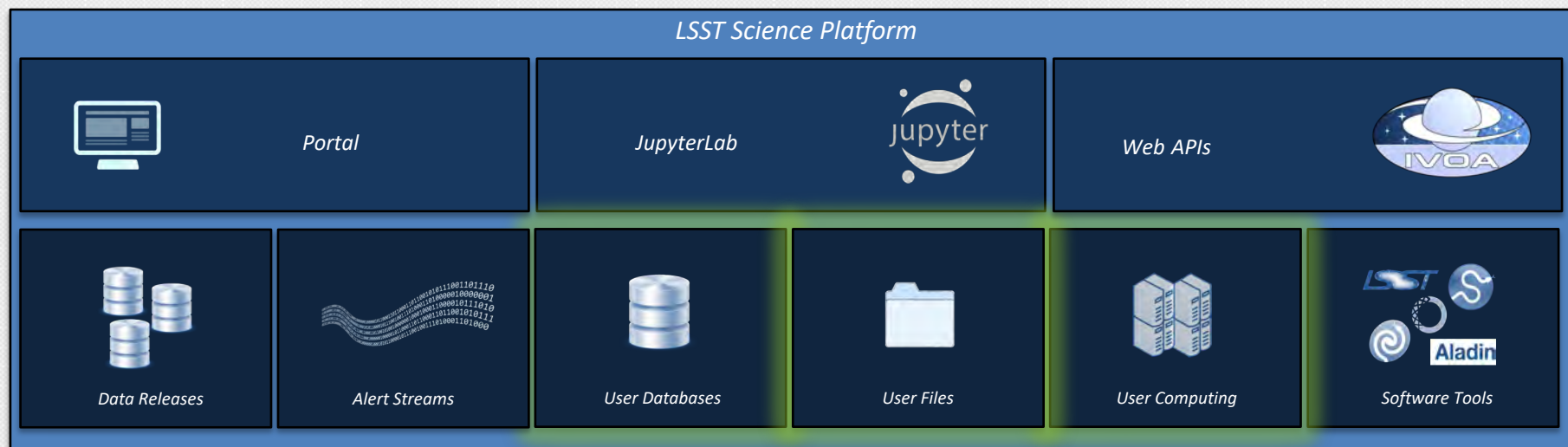
"I've never seen any migration this fast. It's just amazing." -- @mjuric on the rise of @ProjectJupyter in data science



**Why Jupyter is data scientists' computational notebook of choice**  
An improved architecture and enthusiastic user base are driving uptake of the open-source web tool.  
nature.com

10:30 PM - 5 Nov 2018

# Computing, Storage, and Database Resources



Computing, file storage, and personal databases (the “*user workspace*”) will be made available to support the work via the Portal and within the Notebooks.

An important feature is that no matter how the user accesses the DAC (Portal, Notebook, or VO APIs) they always “see” the same workspace.



# How big is the “LSST Science Cloud” (@ DR2)?

## – Computing:

- ~2,400 cores
- ~18 TFLOPs

**This is shared by all users.** We’re estimating the number of potential DAC users not to exceed 7500 (relevant for file and database storage).

Not all users will be accessing the computing cluster concurrently. **We are estimating on order of a ~100.**

## – File storage:

- ~4 PB

Though this is a relatively small cluster by 2020-era standards, it will be **sufficient to enable preliminary end-user science analyses** (working on catalogs, smaller number of images) and creation of some added-value (Level 3) data products.

## – Database storage

- ~3 PB

***Think of this as having your own server with a few TB of disk and database storage, right next to the LSST data, with a chance to use tens to hundreds of cores for analysis. It will be excellent for enabling early science!***

This kind of approach will become increasingly common for *all* big data archives.

# Large-Scale Science in the LSST Era

(my concerns and some potential solutions)



## Challenges (part 1)

Better Together

(joining datasets is powerful)

I Want it All

(science demands whole dataset operations)

3D Dust Mapping  
with Pan-STARRS 1

Query Map Usage Notes Read Papers

## The Map

Interstellar dust attenuates ultraviolet, optical and near-infrared light. Because the extent of this attenuation is wavelength-dependent, dust both dims and reddens the light of stars and galaxies before it can reach our telescopes. In many areas of astrophysics, an accurate correction for the effects of interstellar extinction and reddening is critical. Historically, the most widely used maps of dust have been two-dimensional, tracing integrated dust reddening out to infinite distance. Here, we describe three-dimensional maps of interstellar dust reddening, which trace dust reddening both as a function of angular position on the sky and distance. These dust maps are based on [Pan-STARRS 1](#) photometry of 800 million stars, along with [2MASS](#) photometry of 200 million stars.

To read about how to download the map, or how to query it remotely, read our [usage notes](#). To explore our map in the browser, see our [interactive query page](#). To read in detail about our map, read our [published papers](#).



# Whole Dataset Operations

- Galactic structure: density/proper motion maps of the Galaxy
  - => forall stars, compute distance, bin, create 5D map
- Galactic structure: dust distribution
  - => forall stars, compute g-r color, bin, find blue tip edge, infer dust distribution
- Near-field cosmology: MW satellite searches
  - => forall stars, compute colors, convolve with spatial filters, report any satellite-like peaks
- Variability: Bayesian classification of transients and discovery of variables
  - => forall stars, get light curves, compute likelihoods, alert if interesting
- ...

## Challenges (part 2)

Scalability

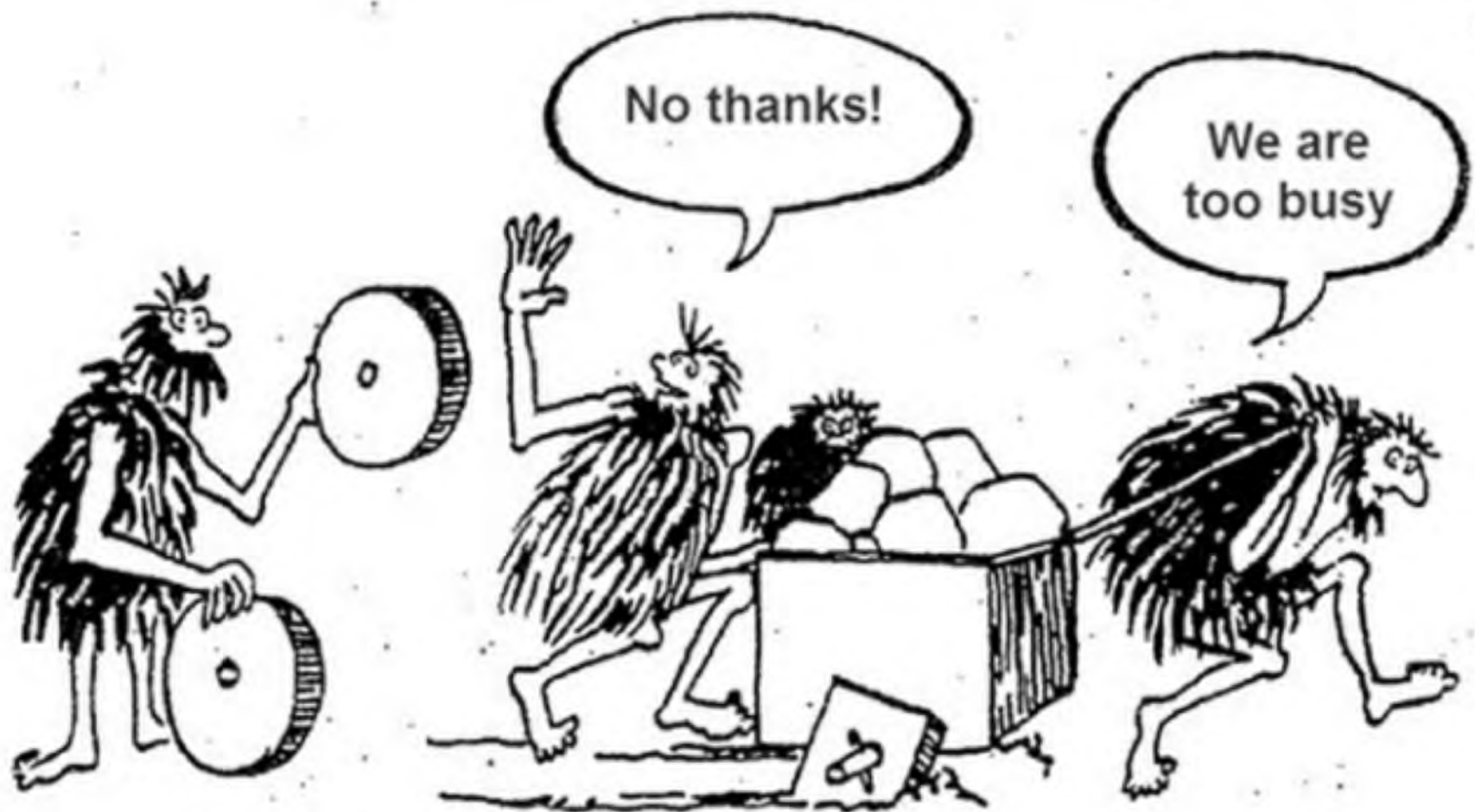
(how do I write an analysis code that will scale to thousands of machines and petabytes of data?)

Resources

(where are the resources to run this code?)



## Industry vs. Astronomy (sometimes)



# The Big Data Open Source Tools Landscape

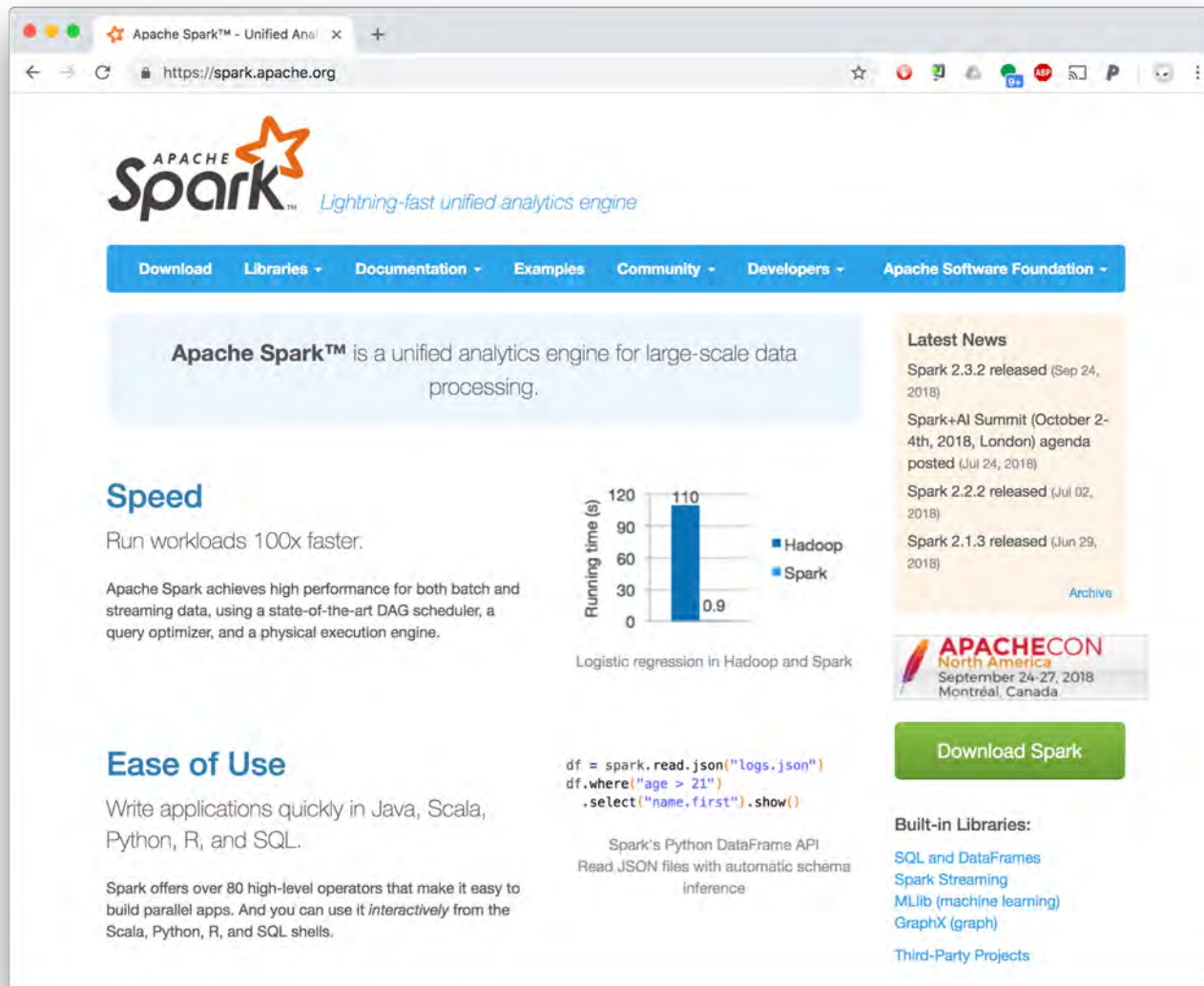
The Dataflop Open Source Landscape 2.0



<https://dataflop.com/big-data-open-source-tools/os-home/>



# Writing Scalable Applications: MapReduce and Apache Spark



The screenshot shows the Apache Spark website in a web browser. The page features the Apache Spark logo with the tagline "Lightning-fast unified analytics engine". A navigation bar includes links for Download, Libraries, Documentation, Examples, Community, Developers, and Apache Software Foundation. A main section describes Spark as a unified analytics engine for large-scale data processing. Below this, there are three main content areas: "Speed" with a bar chart comparing Hadoop and Spark running times for logistic regression, "Ease of Use" with a code snippet and description of the Python DataFrame API, and "Latest News" with a list of recent releases and an ApacheCon event announcement. A "Download Spark" button is prominently displayed.

**Speed**  
Run workloads 100x faster.

Apache Spark achieves high performance for both batch and streaming data, using a state-of-the-art DAG scheduler, a query optimizer, and a physical execution engine.

Logistic regression in Hadoop and Spark

Framework	Running time (s)
Hadoop	110
Spark	0.9

**Ease of Use**  
Write applications quickly in Java, Scala, Python, R, and SQL.

Spark offers over 80 high-level operators that make it easy to build parallel apps. And you can use it *interactively* from the Scala, Python, R, and SQL shells.

```
df = spark.read.json("logs.json")
df.where("age > 21")
  .select("name.first").show()
```

Spark's Python DataFrame API  
Read JSON files with automatic schema inference

**Latest News**

- Spark 2.3.2 released (Sep 24, 2018)
- Spark+AI Summit (October 2-4th, 2018, London) agenda posted (Jul 24, 2018)
- Spark 2.2.2 released (Jul 02, 2018)
- Spark 2.1.3 released (Jun 29, 2018)

**APACHECON North America**  
September 24-27, 2018  
Montréal, Canada

**Download Spark**

**Built-in Libraries:**

- [SQL and DataFrames](#)
- [Spark Streaming](#)
- [MLlib \(machine learning\)](#)
- [GraphX \(graph\)](#)
- [Third-Party Projects](#)

Apache Spark is an open-source distributed general-purpose cluster-computing framework.

**Spark provides an interface for programming entire clusters with implicit data parallelism and fault tolerance.**

-- Wikipedia

## Examples

### Pi Estimation

Spark can also be used for compute-intensive tasks. This code estimates  $\pi$  by "throwing darts" at a circle. We pick random points in the unit square ((0, 0) to (1,1)) and see how many fall in the unit circle. The fraction should be  $\pi / 4$ , so we use this to get our estimate.

Python

Scala

Java

```
def inside(p):  
    x, y = random.random(), random.random()  
    return x*x + y*y < 1  
  
count = sc.parallelize(xrange(0, NUM_SAMPLES)) \  
    .filter(inside).count()  
print "Pi is roughly %f" % (4.0 * count / NUM_SAMPLES)
```

<https://spark.apache.org/examples.html>



## Map

$\{x_i\} \text{ ---map--> } \{y_i=f(x_i)\}$

Apply a function  $f$  to every element of dataset  $X$ , producing dataset  $Y$

## Reduce

$\{ (k_i, v_{ij}) \} \rightarrow \{ y_i=(k_i, f(\{v_{ij}\})) \}$  Apply a function  $f$  to all values with a common key

Example:

$\{ ("dog", 2), ("dog", 1), ("cat", 3), ("dog", 2), ("cat", 1) \}$

-> reduce w. *sum()* ->

$\{ ("dog", 5), ("cat", 4) \}$

## Examples

```
{ ("dog", 2), ("dog", 1), ("cat", 3), ("dog", 2), ("cat", 1) }
```

-> reduce w. *sum()* ->

```
{ ("dog", 5), ("cat", 4) }
```

## Word Count

In this example, we use a few transformations to build a dataset of (String, Int) pairs called counts and then save it to a file.

Python

Scala

Java

```
text_file = sc.textFile("hdfs://...")
counts = text_file.flatMap(lambda line: line.split(" ")) \
    .map(lambda word: (word, 1)) \
    .reduceByKey(lambda a, b: a + b)
counts.saveAsTextFile("hdfs://...")
```

<https://spark.apache.org/examples.html>



# Astronomy Example: Compute Light Curve Features

*This works on arbitrarily large datasets!*

```
In [10]: from pyspark.sql.types import ArrayType, FloatType, DoubleType
from pyspark.sql.functions import col, pandas_udf, explode
import pandas as pd

import cesium
from cesium.time_series import TimeSeries
from cesium.featurize import featurize_single_ts, featurize_time_series

#####

features_to_use = ["amplitude", "percent_beyond_1_std", "maximum", "max_slope",
                  "median", "median_absolute_deviation", "percent_close_to_median",
                  "minimum", "skew", "std", "weighted_average"]

ls_features = ["freq1_amplitude1", "freq1_amplitude2", "freq1_amplitude3",
              "freq1_amplitude4", "freq1_freq", "freq1_lambda", "freq1_rel_phase2",
              "freq1_rel_phase3", "freq1_rel_phase4", "freq1_signif", "freq2_amplitude1",
              "freq2_amplitude2", "freq2_amplitude3", "freq2_amplitude4", "freq2_freq",
              "freq2_rel_phase2", "freq2_rel_phase3", "freq2_rel_phase4"]

def featurize_udf(mjd, psfflux):
    feat_outs = []
    for row_mjd, row_psfflux in zip(mjd, psfflux):
        feat_out = featurize_time_series(np.array(row_mjd), np.array(row_psfflux),
                                         features_to_use=features_to_use + ls_features)
        feat_outs.append(feat_out.values.flatten())
    return pd.Series(feat_outs)

#####

feat_udf = pandas_udf(featurize_udf, returnType = ArrayType(DoubleType()))
spark_session.udf.register("FEATURIZE", feat_udf)

pdf = ztf.where("SIZE(mjd)>50").selectExpr("FEATURIZE(mjd, psfflux)").toPandas()
```



## The Result (with apologies for the appallingly poor visualization)

```
In [12]: pdf = ztf.where("SIZE(mjd)>50").limit(10).selectExpr("ADDMJ(mjd, psfflux)").toPandas()
```

```
Out[12]: [Row(ADDMJ(mjd, psfflux)=[1925.2211608886719, 0.13978494623655913, 3987.0869140625, None, 344
6.4375, 152.688720703125, 0.7419354838709677, 136.64459228515625, -2.4431908318547433, 631.64
34156713688, 3189.848529118364, 313.63354429378256, 14.639380553238073, 1.3644581456964708,
1.7500900946095723, 1.5124216699725148, 42.94394862399773, 2.5313200999890264, 0.635479246010
1448, -0.5061265295340045, 5.0589783305487925, 297.2971196481153, 49.95079199367568, 5.377033
053881004, 2.9892154859975197, 4.01272141380738, -0.11956697923508663, 0.44717905887839726, -
1.1813749683927528]),
Row(ADDMJ(mjd, psfflux)=[208.086181640625, 0.17857142857142858, 848.4930419921875, 57922.265
60191954, 561.7611083984375, 21.10723876953125, 0.7142857142857143, 432.3206787109375, 1.7943
474250678484, 58.001638792839145, 561.3877334594727, 41.09281808433971, 3.6566340247461695,
0.484343106761218, 0.3561043472427892, 1.9945387332853752, 36.82950672444141, -2.622502373517
709, 1.901872187975775, 1.3933600656842617, 3.208573392947105, 26.071430553383156, 1.26390776
08450804, 0.47282767919988666, 0.11065610514135481, 23.0800641543819, -2.85326837788731, 1.89
44949539332152, -0.3698982601601857]),
Row(ADDMJ(mjd, psfflux)=[491.52618408203125, 0.1320754716981132, 1122.197509765625, 32600.79
9800087658, 402.0052490234375, 31.34088134765625, 0.7924528301886793, 139.1451416015625, 2.88
3311815955483, 131.10471926999602, 416.95930855229216, 114.34083688267272, 10.76770241852664
3, 4.481966442155684, 1.2482773020089568, 1.0816044062094867, 24.405764071747054, -1.35716539
37044816, -2.4773879544783286, 2.77126228292522, 3.23567097891298, 60.04996221378995, 5.37096
0993975372, 0.8751468632528988, 0.3325519757277276, 0.6780459096769216, -0.8609512237261738,
-1.5228145978972758, 1.6412947076259528]),
Row(ADDMJ(mjd, psfflux)=[107.95289611816406, 0.17333333333333334, 303.4608154296875, 15769.0
08109654327, 146.07757568359375, 16.181289672851562, 0.64, 87.55502319335938, 1.9784213046179
848, 33.608624825862975, 152.7379244995117, 14.807230580579672, 0.2625194273012284, 0.1712371
6298087752, 0.1045290010068471, 29.250142510335483, 75.41574345145936, -1.005145004902362, 2.
1760284474899296, 1.1339028285326511, 2.980258224087021, 18.704151766762745, 1.06178361034372
75, 0.283697089687251, 0.18280423388998288, 0.965224836036898, 0.24573780352703498, -1.282736
1427620787, -2.016597738480162]),
Row(ADDMJ(mjd, psfflux)=[1531.0211791992188, 0.14285714285714285, 4344.10595703125, None, 39
23.2584228515625, 160.582275390625, 0.6587301587301587, 1282.0635986328125, -2.22651166992487
46, 582.6092968330478, 3688.5484958224824, 387.9816297071594, 48.53561353536267, 12.997457710
221171, 6.372125562326946, 3.0098298953254314, 13.648208330625398, 0.4783521707732701, -0.520
8550287735074, 0.9412748273621686, 6.5045045280108695, 236.76237021479648, 3.799074459364612,
```



Want to try it out?

```
conda install -c conda-forge pyspark
```



# Scaling with Spark

Today, Spark is being adopted by major players like Amazon, eBay, and Yahoo! Many organizations run Spark on clusters with thousands of nodes. According to the Spark FAQ, the largest known cluster has over 8000 nodes. Indeed, Spark is a technology well worth taking note of and learning about.



<https://www.toptal.com/spark/introduction-to-apache-spark>





+ government-sponsored private clouds (e.g., JetStream)

## Cloud services

- Essentially, companies that rent computers (or a few million of them)
  - The same for storage.
- Pay only for what you use (by the second/minute/hour)
- Scalable: ask for 1000 machines, get a 1000 machines
- Becoming cost effective (TCO)
  - Especially “spot” pricing



# Meeting the Challenges

Dataset Storage

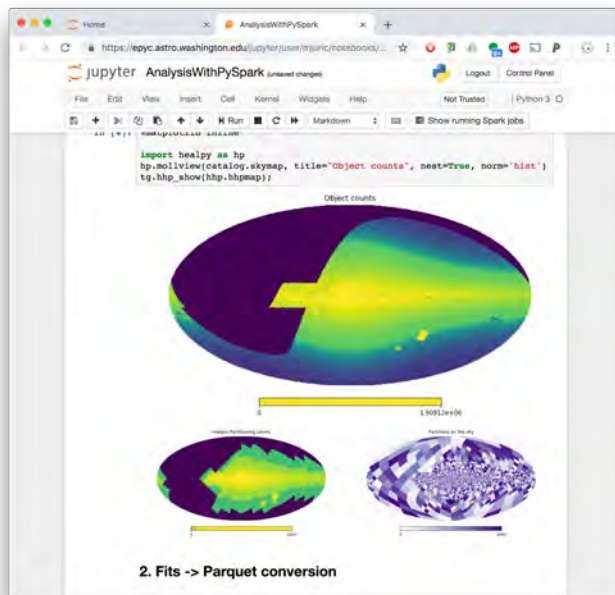
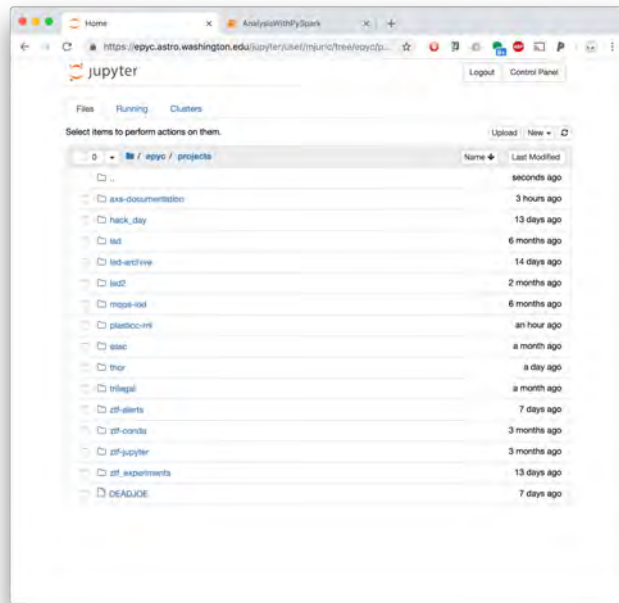
Resources

Scalable Analysis  
Code

Interface



# “Analysis 2025”





# A Number of Projects are Working to Make this Happen

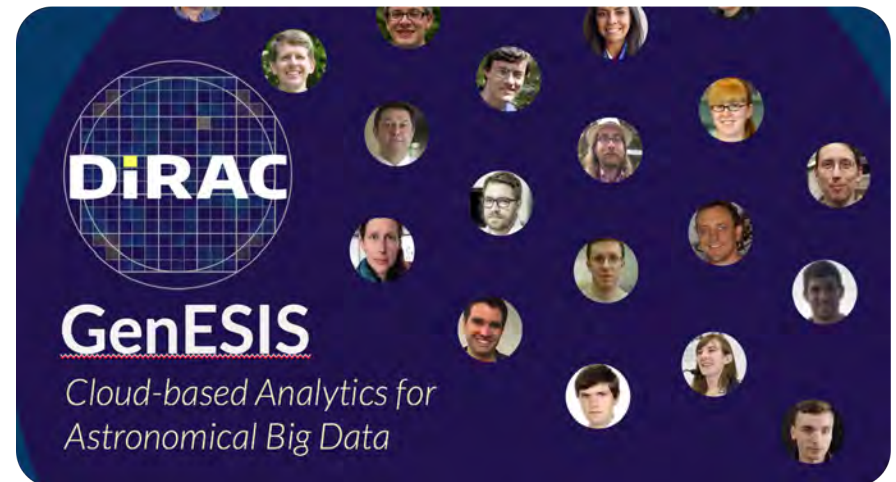


**PANGEO**

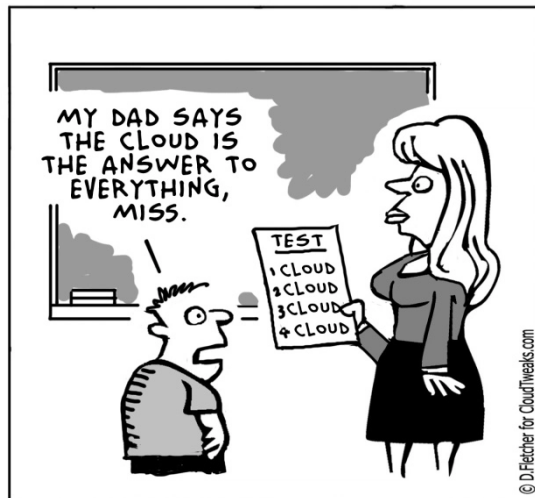
A community platform for Big Data geoscience

<http://pangeo.io/>

*Coming soon w. ZTF !*



# Some Words of Caution



Just like with machine learning / A.I., there's no need to throw cloud at everything.

Small datasets?  
Large-ish datasets?

But the *programming model* works across all scales.



The implementation of these technologies is still in its infancy. They change incredibly quickly.

Expect you may need to shift from framework to framework (e.g., Spark → Dask).

**That said, the *programming models* change on a much longer timescale (e.g., MR 2004 → ?).**





# Summary

- > After decades of planning and construction, LSST is coming soon! First light in 2020, science commissioning in 2021, start of operations in October 2022! LSST is around the corner.
- > Remote analysis platforms are being set up to “bring the code to the data” and lower the barrier to entry to working with the dataset. Remote access through Jupyter is becoming a standard.
- > Large scale (~PB), end-user analysis remain an unsolved problem (both software and resources) in academia. Adopting cloud-ready solutions from the industry is one way forward. MapReduce and related frameworks (Spark, Dask, etc.) will play a large role.