Rubin Science Platform on Google Cloud: The story so far

William O'Mullane *(Rubin)*
Frossie Economou *(Rubin)*
Hsin-Fang Chiang (*Rubin)*
Flora Huang *(Google)*
Dan Speck *(Burwood)*

... AND ...

VERA C. RUBIN OBSERVATORY

It takes a village

Rubin and LSST in a nutshell

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

- 90% of time spent on uniform survey: every 3-4 nights, the whole observable sky scanned twice per night
- 100 PB of data: about a billion 16 Mpix images, enabling measurements for **40 billion objects!**

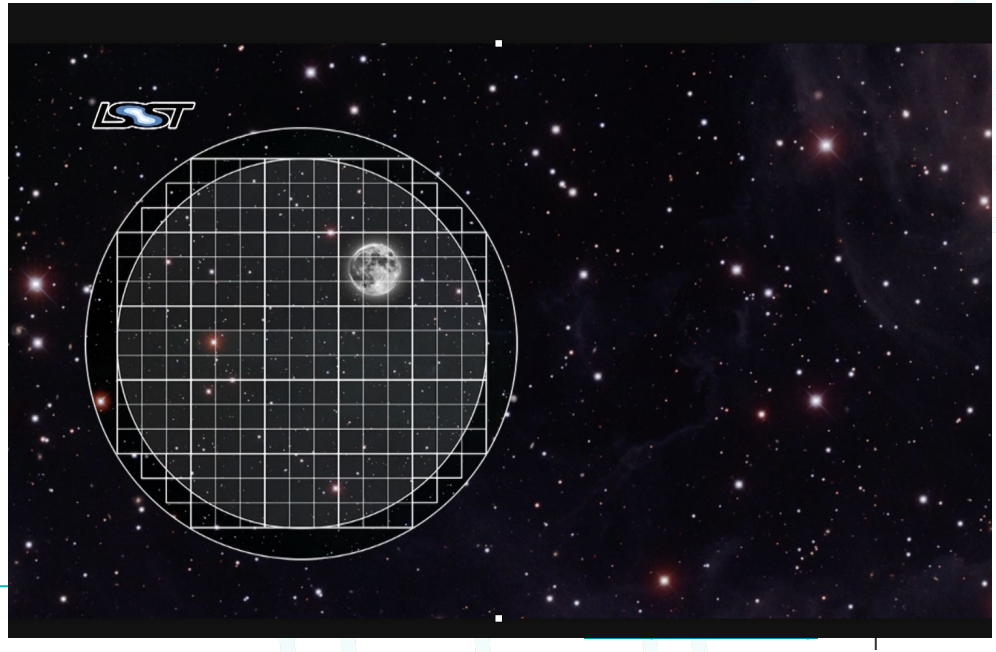
see also http://www.lsst.org

Ivezic´et al. (2019)-arXiv:0805.2366

https://www.lsst.org/scientists/keynumbers

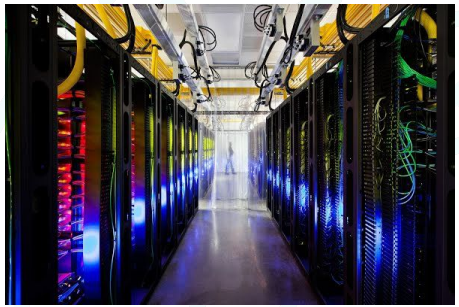
Situated on Cerro Pachón Chile (2647m)

Largest Camera ever (3.2 Gpixels)



Interim Data Facility on Google Cloud

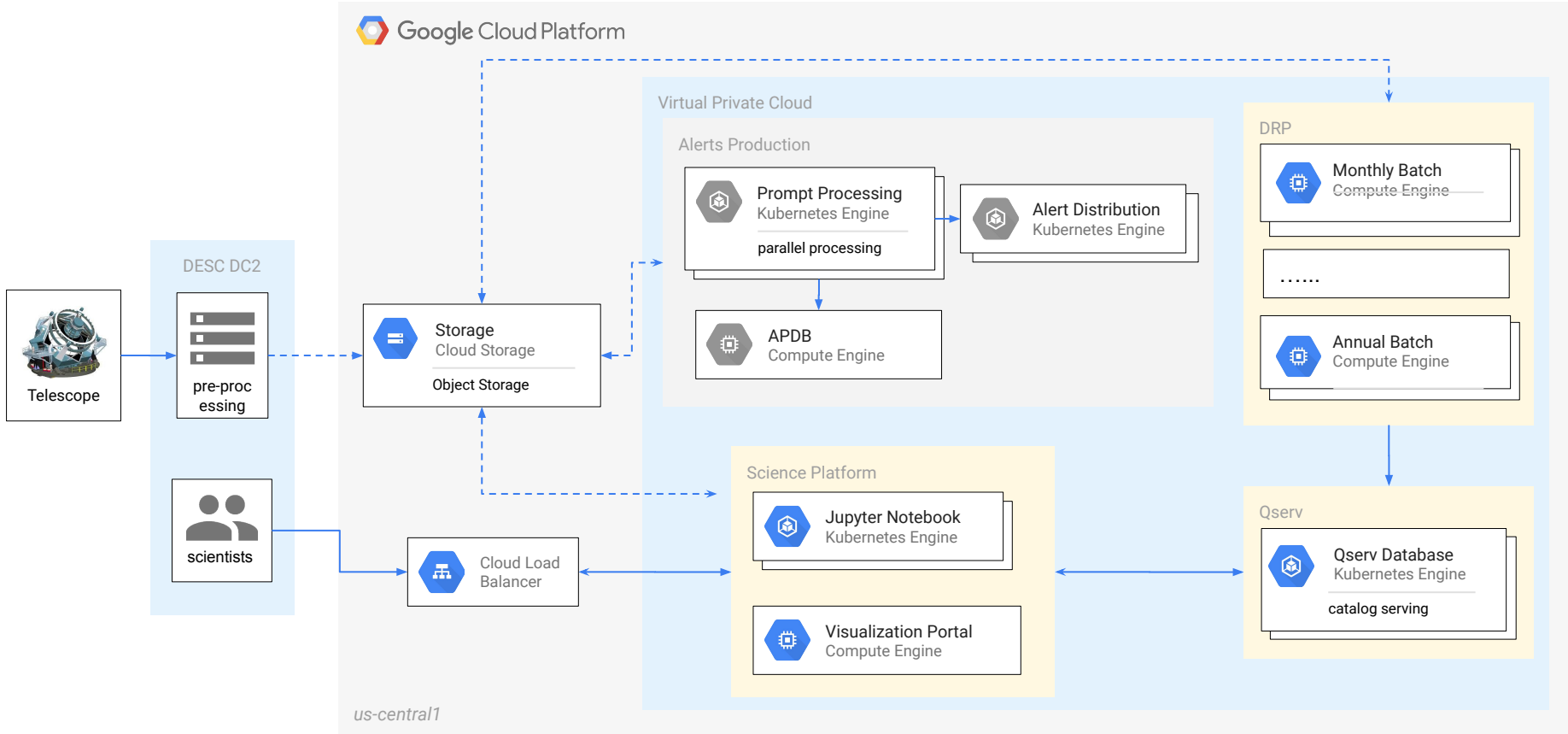
- Due to a change in the funding landscape, what was conceived as the LSST Data Facility in construction became the US Data Facility in operations - only for a while we had no idea where that would be
- We set up an Interim Data Facility (IDF) on Google Cloud as a way of bridging the gap between on-prem data facilities and servicing early users through our Data Preview program
- **This is not a toy or proof of concept; it is a full-scale production environment** without fall-back or dependance on on-prem for its released into production functions
- Contract signed October 2020, infrastructure essentially complete March 2021, production services aimed at external users released for the first time in June 2021
- Running services at scale allows for a high degree of readiness for Data Production activity in full survey operations



Slide: Hsin-Fang Chiang

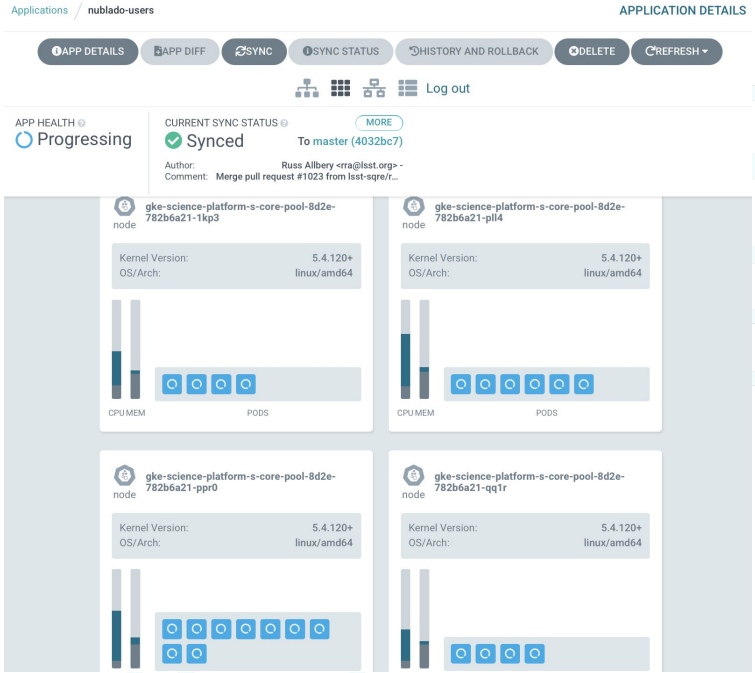
IDF - Architecture

Slide: Flora Huang



IDF deployment of Rubin Science Platform + friends

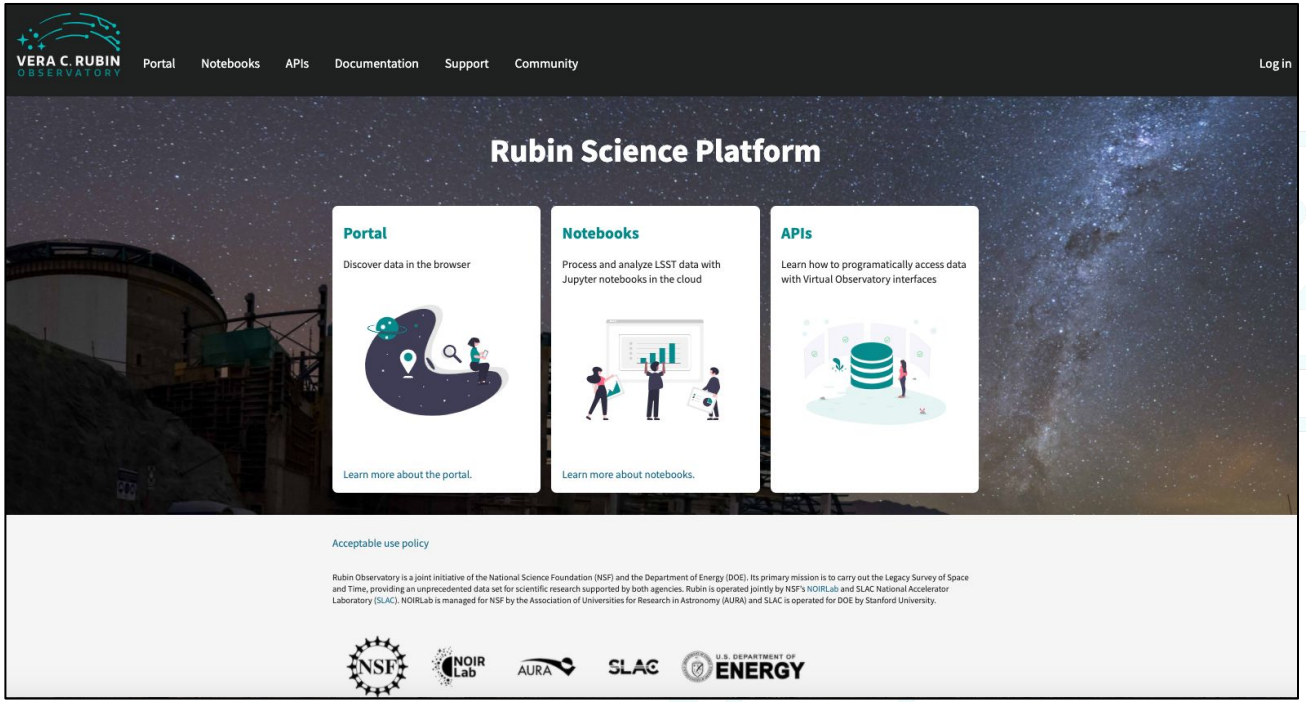
- No Google console operations: clusters managed by terraform for reproducibility and traceability (Infrastructure as Code)
- Services running on Kubernetes and managed by ArgoCD
- Only developers have infrastructure (Google) accounts; users authenticate to services through the Rubin Science Platform authentication and authorization service (currently backed by Github Oauth for this deployment)
- Data access is through our Gen3 Butler data abstraction layer, at the IDF this is backed by the Google Cloud Storage object store; posix home spaces via Google Filestore
- Our in-house high performance database service (Qserv) is also deployed on Kubernetes; Gen3 registry uses CloudSQL



Slide: Hsin-Fang Chiang

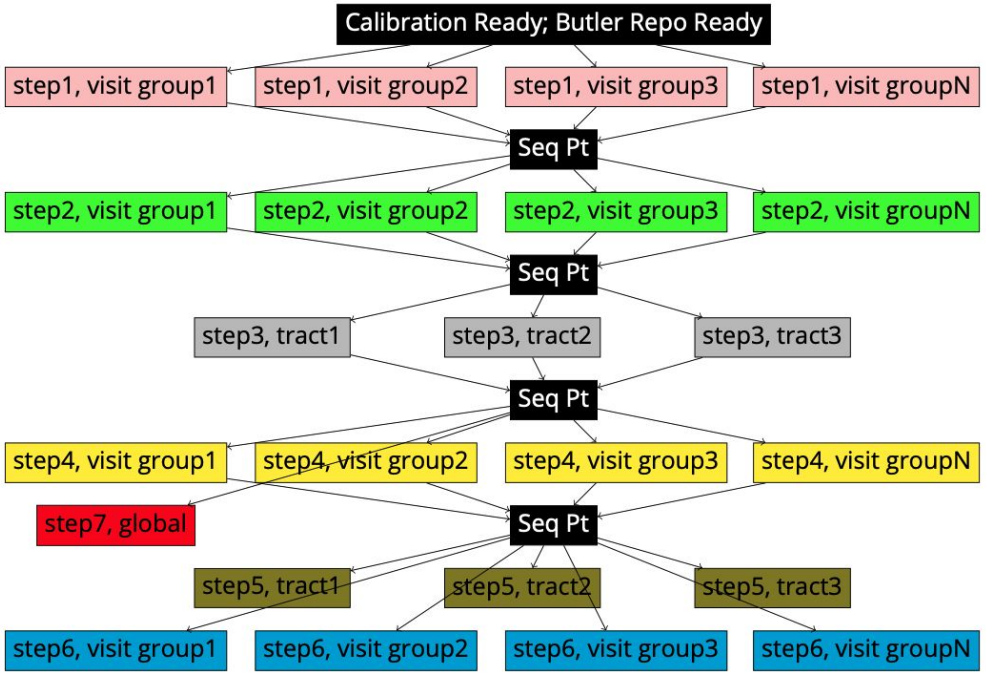
Data Preview 0.1! June 2021

- Pre-Operations Activity for Rubin and the community based on DESC simulations
- Science platform is up as planned on Google Cloud IDF data.lsst.cloud
- Data product documentation from Community Engagement Team (CET): dp0-1.lsst.io
- Delegates (users) gained access in June!



DP0.2 - June 2022

- Reprocess the DESC data (DP0.1) with Pipelines V23
  - generate a fully self-consistent data release
  - Demonstrate portable set of cloud enabled tools based on Butler Gen3 and PanDA
  - Produce and load Catalogs
  - FIrst Operations Rehearsal for Data Release Production
- Pipelines to run stepwise
  - This is an experiment
  - Flow chart on right.
- our middleware with PanDA
  - Data processing split between USDF🇺🇸, FRDF🇫🇷, UKDF🇬🇧

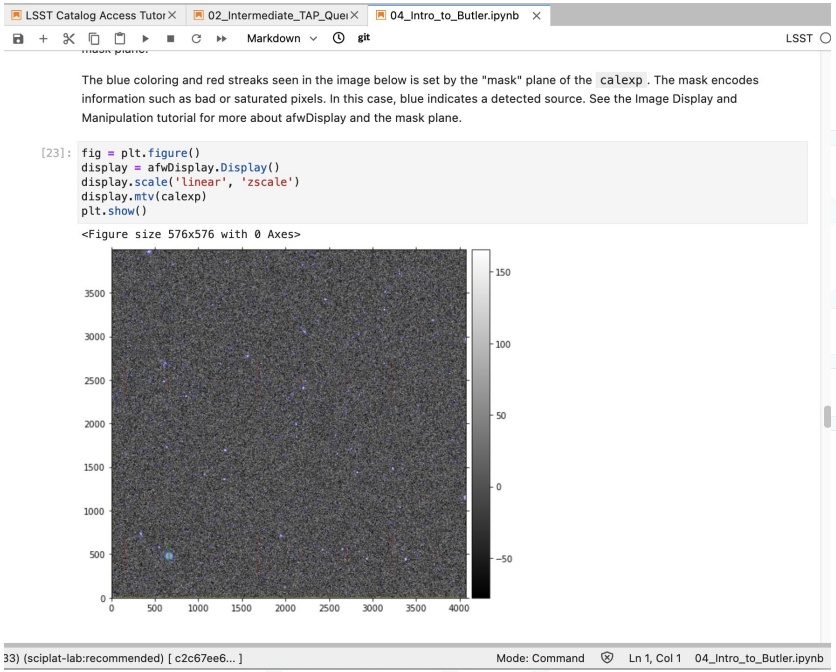


Community Engagement in Data Preview 0

~ 250 “DP0 Delegates”, scientists and students from the Rubin community, are participating in DP0.1 (ops users estimated at ~ 10K)

Resources and activities provided by the Rubin Community Engagement Team (CET), such as:

- biweekly virtual “Delegate Assemblies” with hands-on demonstrations
- the full suite of DP0 documentation available at ls.st/dp0-1 (and tutorial Notebooks)
- a dedicated Community.lsst.org category: “Support -- Data Preview 0”
- all of which is publicly accessible



Slide:Melissa Graham

Things we learned along the way

Users need a lot of guidance if we are to be efficient:

- Qserv (ADQL) which is SQL is powerful but not everyone knows wha you can do
  - E.g. can build great histograms (minutes for full dataset) in SQL (Qserv)
  - Users in notebooks tend to do this in python by pulling in all the data and binning in python
- Users will find ways to consume all your available resources; have a plan!
- AutoScale really works - but its not really fast, takes **minutes** to spin up new node - **eons** as far as users concerned!

You need management buy-in :

- Proof of concepts helped a lot
  - Six months with Google 2018/2019 (DMTN-125)
  - Six months with Amazon 2019 (DMTN-137)
- Still prepare for lots of explaining and cost modeling.

Summary

- Where we were prepared by having design and built cloud-ready services, transition was fast and painless (Kubernetes Will Save Astronomy ™)
- Tangible benefits to working with a highly popular toolchain
- Ideal way to mitigate schedule risk for on-prem computing delivery
- Not cheap but great value for money
- Developers love the self-serve aspect (and their velocity shows it)
- Vendor lock-in is not the issue; the working style to which you become accustomed (and not wanting to give it up) is the issue
- We are seriously evaluating whether an on-prem/cloud hybrid model is actually the best way forward permanently
- From a technical perspective, use of commodity computing is a no-brainer

The End

Blast 20 Cerro Pachón April 2011

http://community.lsst.org

Rubin Observatory Sept 2019

http://www.lsst.org