Rubin's Hybrid On Premises-Cloud Data Access Center

William O'Mullane[1], Yusra AlSayyad[2], James Chiang[3], Richard Dubois[4], Frossie Economou[5], Fabio Hernandez[6], Flora Huang[7], Tim Jenness[5], Kian-Tat Lim[4], Yee-Ting Li[4], Fritz Mueller[4], Dan Speck[8], and Wei Yang[4]

[1]Vera C. Rubin Observatory, Avenida Juan Cisternas #1500, La Serena, Chile
[2]Department of Astrophysical Sciences, Princeton University, Princeton, NJ 08544, USA
[3]Kavli Institute for Particle Astrophysics and Cosmology, SLAC National Accelerator Laboratory, Stanford University, Stanford, CA 94025, USA
[4]SLAC National Accelerator Laboratory, 2575 Sand Hill Rd., Menlo Park, CA 94025, USA
[5]Rubin Observatory Project Office, 950 N. Cherry Ave., Tucson, AZ 85719, USA
[6]CNRS, CC-IN2P3, 21 avenue Pierre de Coubertin, CS70202, F-69627 Villeurbanne cedex, France
[7]Google LLC, 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA
[8]Burwood Group, 125 South Wacker Drive Suite 2950, Chicago, IL 60606, USA

ABSTRACT

Cloud computing offers unparalleled flexibility, a constantly increasing set of "Infrastructure As A Service'' capabilities, resource elasticity and security isolation. One of the most significant barriers in astronomy to wholesale adoption of cloud infrastructures is the cost for hot storage of large datasets - particularly for Rubin, a Big Data project sized at 0.5 Exabytes (500 Petabytes) over the duration of its 10-year mission. We are planning to reconcile this with a "hybrid" model where user-facing services are deployed on Google Cloud with the majority of data holdings residing in our on-premises Data Facility at SLAC. We discuss the opportunities, status, risks, and technical challenges of this approach.

1. INTRODUCTION

In 2019 the funding of Vera C. Rubin Observatory[1] operations changed with the Department of Energy (DOE) increasing its contribution to 50%, with the bulk of that funding the US Data Facility (USDF) at a site to be determined. This led to changes in how and where we would operate Rubin Data Management. We used the opportunity and uncertainty to propose the Interim Data Facility (IDF), a cloud-based solution, thus alleviating the immediate need to know the location of the USDF. The IDF has been very successful and supported three data previews with simulated data.[2] When SLAC National Accelerator Laboratory was selected as the USDF we maintained the interim solution to overlap the startup with the USDF; however, as we discussed the architecture a hybrid solution emerged. Keeping all science users on the cloud has certain security and scalability advantages while keeping the bulk of the data at SLAC has some cost advantages. DOE has committed funds for three years of the US cloud-based Data Access Center (DAC) on Google, which should bring us to 2027. The interim cloud will transition to become the US DAC. For the first two data previews all data was on Google; the third data preview had the database at SLAC and users on Google. The intention is to have most data at SLAC with the users accessing databases using IVOA protocols and images using the client/server Butler.[3] Thus the users do not have SLAC accounts and do not require approval through more detailed institutional processes. The system is built on terraform and kubernetes deployed with ArgoCD using our own Phalanx configuration system.[*]

[*] https://phalanx.lsst.io

2. SYSTEM REQUIREMENTS

Rubin Observatory has a rigorous approach to requirements management.[4] Most relevant requirements for the USDF are in the Data Management Subsystem Requirements (DMSR) document.[5] Having to switch data facilities encouraged us to pull system requirements which affect the data facility into one document which was used to scope the SLAC operations,[6] which we will not enumerate here although the tables of requirements broadly matching this section may be found there. The USDF is responsible for significant functionality in several areas as outlined below.

2.1 Networking

USDF must arrange 100Gbit/s, path redundant, network capacity to the Energy Sciences Network (ESNet) to connect to the Rubin Observatory facility in Chile. USDF must ensure their contribution to network latency is maximum 3s. Enough bandwidth must also be available to exchange files with France and the UK for annual Data Release processing.

2.2 Prompt processing

The USDF will need to run the Prompt Processing framework in near-real-time in order to execute the Alert Production payload that generates prompt data products and alerts corresponding to changes in the sky. The processing is to be completed and alerts are to be distributed within two minutes of the end of readout of an image from the LSST Camera. Quality control metrics for the images also need to be generated and made available to staff. Prompt products, including both images and catalogs, and alerts are stored for retrieval by science users and staff.

2.3 Batch System

Every year, the accumulated images taken to date will be reprocessed. This extensive and complex Data Release Production runs in batch mode across the US, French, and UK Data Facilities. The USDF is responsible for providing infrastructure for executing 35% of the Data Release, coordinating the campaign to generate the annual Data Release, ensuring the quality of the data products, archiving a copy of 100% of those products, and making them available to science users through the US DAC. Certain products will also be distributed to Independent Data Access Centers and Science Collaborations.[7]

The batch system and associated interactive nodes are also used extensively by staff for development of future versions of the LSST Science Pipelines code.[8] Though most science user access will be via the Data Access Center (DAC) (see subsection 2.5) there are requirements for science users to have some access to batch-type processing for larger-scale, non-interactive computations on the data.[9] This will be controlled by a Resource Allocation Committee

2.4 Data transfer and preservation

The USDF is responsible for operating the systems that track all raw data and released data products, including managing their movement, backup, and lifetime. These systems must be fault-tolerant and able to catch up after failures.

2.5 US Data Access Center

The USDF hosts the US DAC. To science users, the DAC appears primarily as an installation of the Rubin Science Platform (RSP)[10] see Figure 1. This software includes the Portal Aspect, a web-based application for browsing, querying, and investigating the data products; the Notebook Aspect, which allows interactive, customized programs to retrieve and manipulate the data products; and the API Aspect, which provides community-standard protocols for automated retrieval of data. The RSP is deployed using Helm and ArgoCD as a suite of services on top of Kubernetes.

In the hybrid model, most of this will now be hosted in the cloud, with the underlying data, both prompt products and annual Data Release products, fed from the USDF at SLAC. But an RSP for staff will also be deployed on top of a Kubernetes cluster provided by the USDF.

This system is to be sized for around 10K users with perhaps 1K simultaneously accessing at any given time. Each user will have a limited amount of user space (order 100GB) for storage of output images and queries.

There is further functionality specified for the DAC such as precovery, product regeneration and special program support which imply invocation of batch processing. User generated products must be stored and potentially shared; catalog uploads will also be allowed. Access is not only to the current Release but also one previous Release of the data.

3. GENERAL CLOUD BENEFITS AND DRAWBACKS

Commercial cloud providers are generally competitive with on-premises facilities for provision of compute cycles, especially where the needs are time-varying. Storage costs, however, tend to be higher in the cloud than at research facilities, possibly due to higher durability and availability requirements for commercial data than for research data. In terms of networking, cloud providers do not charge for data movement to the cloud from elsewhere, but they do charge for egress from the cloud.

The hybrid model we have chosen uses the cloud for highly-elastic support of science users while keeping the more-constant near-real-time nightly compute load and high-throughput annual reprocessing load at on-premises facilities. The bulk of the storage is also maintained on-premises, with relatively small caches and user data living in the cloud. This design ensures that the bulk of the data transfer is from SLAC to the cloud, qualifying as free ingress. Science users are expected to process the data in the cloud, reducing its volume prior to retrieval to their own systems. Thus we attempt to use the strengths of each provider.

Egress charges can be a significant worry when dealing with unconstrained users accessing popular data products. We believe these can be controlled through three means: contractual waivers and discounts, throttling of Science Platform applications and interfaces with quota allocations by a Resource Allocation Committee, and potentially the purchase of a fixed-cost, fixed-bandwidth exit network connection to be used by all science users.

One of the major advantages of this approach beyond simple cost considerations is the separation of security concerns. Users at SLAC require extra checks which take some time because it is a DOE facility. SLAC may have difficulty processing accounts for our many thousands of users. By putting all the science users on Google we can streamline access using InCommon* which will allow us to identify most US academic users.

Another benefit that is not directly financial is that the commercial cloud providers are incentivized to provide excellent managed infrastructure tooling such as Kubernetes, relational databases, messaging systems, and log explorers, in addition to the underlying compute, storage, and networking. Being able to rely on such sophisticated, performant, and reliable tooling eases deployment of our services. Academic and research facilities, including SLAC, are often behind when it comes to providing this level of support.

A major plus for on-prem in this instance is the backing of DOE; this is what they want, and there is commitment to making it work.

4. ARCHITECTURE ON GOOGLE CLOUD

As the RSP (Figure 1) services were built for cloud deployment (see section 6), they are elastically scalable and easy to set up. The Portal and Notebook Aspects each run as an application (in the ArgoCD sense), while each API within the API Aspect is its own application. They rely on a common authentication and authorization (and throttling) infrastructure to ensure that only science users with appropriate data rights have access.

The back-end communication with the USDF services hosting the bulk of the data products is mediated through two primary interfaces: a relational database query interface for catalogs that can talk to the scalable, distributed Qserv service[11,12] and the client/server Butler[3,13] for access to image and other file-based data products.

The client/server Butler manages metadata about data products and user rights to access those products. It can generate signed URLs that provide anonymous, bearer access for a limited time to specific files in the underlying storage at the USDF. These URLs can be consumed by RSP services, or they can be returned directly

*https://incommon.org/

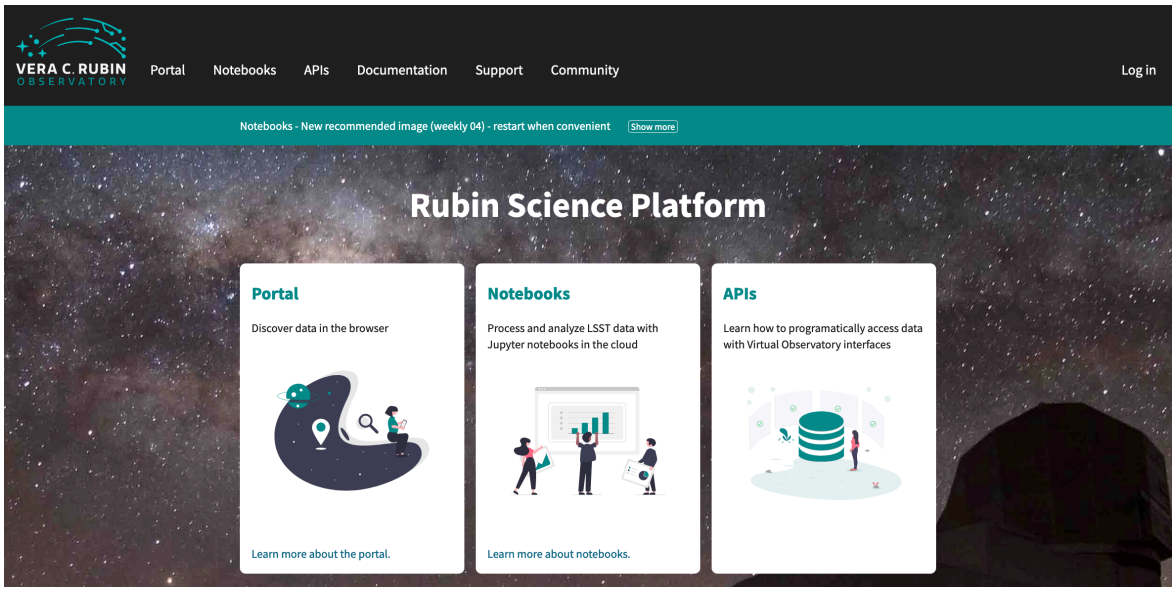

Figure 1. Users hosted on Google will typically use the Rubin Science Platform (RSP) depicted here.

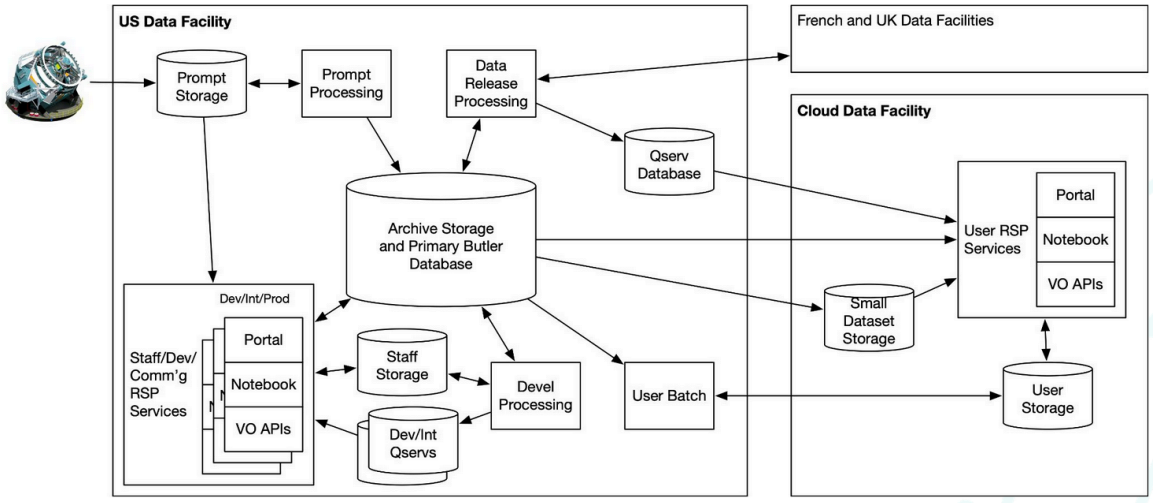

Figure 2. Hybrid model: Data at SLAC but users on the Cloud.

to science users, avoiding mediation of network traffic by Google (which would count as egress). For performance reasons, we may host a subset of the data products, either statically or dynamically chosen (or a combination of both), in the cloud.

We will also provide a service for advanced science users to execute batch jobs. Since these jobs are expected to involve processing a large amount of data, it makes sense to run them at the USDF. We can either have these specialized users obtain SLAC accounts, or we can use batch processing tools designed for the grid to manage the jobs and corresponding resource and access controls.

We reported on the Interim Data Facility (IDF) already at ADASS.[2] Our services have been operating on Google Cloud for over three years now. In Data Preview 0.3 on IDF,[14] the database instance was moved to SLAC, so we have demonstrated this also works.

5. US DATA FACILITY ARCHITECTURE

The scope for the USDF on-prem includes data production services: prompt processing, serving alerts to the community and annual Data Release Processing. The USDF acts as the archive for all data, and provides the Qserv object catalog as well as access to image data, be it cutouts or full images. It will provide batch cycles for cloud-based science users. It will also act as a home for developers and staff (and commissioners) to ensure data quality. This is depicted in Figure 2.

5.1 Hardware

The USDF is hosted by the SLAC Shared Scientific Data Facility (S3DF) which is itself hosted in Stanford Research Computing Facility (SRCF). SRCF accommodates projects from SLAC and Stanford, while the S3DF is the focal system for SLAC projects. The USDF lives in a shared cluster and benefits from economies of scale and standardization across S3DF projects. It is also exposed to potentially disruptive activities by other projects.

In order to support hundreds of PBs of storage, S3DF adopted the Weka filesystem for high throughput. Weka is based on a tiered system with Solid State Disks (SSD) backed by spinning disk. It presents a POSIX interface while the backend is a Ceph object store. This system forms the basis of the data archive. A tape robot provides storage for seldom-read data and acts as a backup tier.

Batch processing is done on a Slurm cluster, currently primarily Advanced Micro Devices (AMD) milan processors with 128 cores and 512 GB RAM per node.

Data is transported to the USDF from the summit over a combined leased-line, ESNet supported network with routing optimized via an overlay. The leased line terminates in Atlanta, where ESNet takes over. Traffic to the two other Data Facilities is also provided by ESNet, connecting to the GEANT* and Renater systems in Europe.

5.2 Batch processing

The USDF supports batch processing for a number of purposes: annual multi-site data releases; pipelines teams testing for algorithms performance; processing by individual developers for their algorithm development; data quality checking and validation.

Multisite processing makes use of the Production ANd Distributed Analysis system (PanDA),[15] developed by ATLAS (A Toroidal LHC Apparatus) for the Large Hadron Colider (LHC). It has a well-defined mechanism for routing work from a central server to multiple remote locations. ATLAS has demonstrated submitting millions of jobs per day to hundreds of sites. A difference between typical astronomy and High Energy Physics (HEP) workflows is the number of and duration of processes: astronomy tends to many more much shorter jobs than HEP.[16] Significant effort was required working with the PanDA team to cluster up short jobs to avoid prohibitive startup costs. PanDA is a heavyweight solution to processing; local processing for the pipelines teams and developers is done using HTCondor.

Data management and movement is also orchestrated by LHC tools: Rucio for data management and FTS3 for movement. These tools also routinely handle large numbers of files and transfers, but the astronomy:HEP difference persists here as well, with astronomy generating many more, much smaller files than HEP. This will make the Rubin Rucio database bigger than ATLAS's and will require some growth planning.

The large number of small files will also be a challenge for network transfer. We are investigating zipping up large numbers of files both for better transfer as well as easier storage on tape.

*pan-European data network for the research and education community

5.3 Non-user-facing services

Currently the primary reasons for putting services on-prem are a low latency requirement for prompt processing, and the still-unfavorable comparison of storage prices between on-prem and cloud. To a lesser degree, those comparisons also apply to CPU.

This means that Prompt Processing and Alerts production, with their 2-minute latency requirement are hosted on-prem. Additionally, there are security requirements on data arriving at the USDF, including physical measures implemented on the racks themselves.[17]

The large data volumes associated with the storage archive and Qserv database hosted at the USDF implies that external access to them must be provided by services.

Kubernetes is used to manage almost all our services, making use of ArgoCD as well as our custom Phalanx system (see §6). Native Kubernetes tools are used to manage standard services, such as Postgres databases, making administration, backups, etc, scalable. Rucio and PanDA are managed by Kubernetes to take advantage of these features.

The Prompt Processing framework executing the Alert Production is implemented using Knative on top of Kubernetes to allow elastic instantiation, configuration, and teardown of pods responding to notifications from the summit in advance of the next visit. Prompt Processing and Alert Distribution both use Kafka installed on Kubernetes. We use the Strimzi Operator to install Kafka and for management. Strimzi has worked well to simplify installation, upgrades, and for maintaining health of the Kafka clusters.

Three large database systems are minimally using Kubernetes, as they are either commercial or custom services with no native Kubernetes support. These are the Engineering and Facilities Database (EFD),[18] Qserv, and Cassandra systems, with Qserv the custom system. The EFD is implemented with an InfluxDB Enterprise High Availability cluster.

For monitoring we use Prometheus. Prometheus has native support for Kubernetes metrics and many of the Kubernetes operators described earlier like Strimzi and the Cloud Native Postgres (CNPG) natively provide metrics in Prometheus. We also wrote some of our own metrics to track Prompt Processing. We use Grafana for creating dashboards to visual metrics and for generating alarms when thresholds are crossed. For logging we use Loki from Grafana Labs to capture logs from Kubernetes and application level logs. Loki supports S3 as a backend so we store logs in an on-premise S3 Ceph Cluster.

6. CONTINUOUS DEPLOYMENT ACROSS THE RUBIN FACILITIES

Being cloud ready is certainly the biggest challenge for most projects/applications which intend to leverage the commercial cloud. Early on we adopted Kubernetes which provided a service architectures that is well isolated from the underlying infrastructure. This approach has already paid off massive dividends. For example when funding lines suddenly shifted we were able to painlessly transition from an on-premises facility to an Interim Data Facility on Google Cloud. Furthermore the Rubin Science Platform (RSP) became a generic data services platform that is currently deployed on eight distinct (and distinctly managed) infrastructures (on-prem and cloud). Finally this has allowed us to leverage the Cloud for services like the RSP which benefit from its advantages such as elasticity, scalability, isolation.

6.1 Division of responsibilities

We have had ways to abstract system services from our developed services for some time, containers and java effectively allow one to run *anywhere*. These work well enough for single user single processor deployments. When you need to scale up another level of orchestration is needed, kubernetes fits neatly between our infrastructure and our developed applications providing a powerful container orchestration and resource management system.

Our agreement with each facility is to provide a Kubernetes deployment platform upon which we may deploy our services. Each thus provides a kubernetes environment with disk, network and compute resources upon which we spin up our services. A vault for secrets is also needed. From the service perspective each facility looks similar. It feels like the first time that this actually works well - we have learned to make our applications portable of course but also the technology is so much better than any precursors.

6.2 Phalanx, Helm and ArgoCD

Helm[*] is a standard approach to tell Kubernetes about software. For our purposes this is most usually a JSON file describing which GitHub repo our code is in where the container is and how to start it.

Of course there are parts of this configuration which are site specific and parts which are generic. For example the database URL for an application will be different in USDF and Summit, the *value* of the database URL is different are each site, the URL variable is the same so the application does not change only the configuration. Phalanx[†] is our repository of configuration files for all of our deployments. Within Phalanx each application has a directory with a helm configuration - it also has a values file for each environment to specify the specific values for that environment. Each environment also has a Vault configuration so that secrets such as database password may also be specified by an environment specific identifier. Finally Phalanx has a list of deployed applications on each site.

Actual deployment is manged by ArgoCD[‡]. As the name implies ArgoCD provides continuous delivery for Kubernetes. It interprets the configuration files stored in Phalanx to deploy containers on the kubernetes system associated with each environment. Some of the current environments are listed in Table 1, typically each environment also has a *dev* and *integration* version. This is a powerful system which can track the main or any given branch or tag of a github repository to keep the application in sync.

Table 1: Phalanx environments - typically we have dev, int and production for each.

Name	**Environment endpoint**
base	base-lsp.lsst.codes (La Serena)
ccin2p3	data-dev.lsst.eu (French Data Facility)
idfprod	data.lsst.cloud (Production RSP in GCP)
minikube	minikube.lsst.codes (GitHub Actions CI)
roe	rsp.lsst.ac.uk (UK Data Facility)
roundtable-prod	roundtable.lsst.cloud (SQuaRE services)
summit	summit-lsp.lsst.codes (Rubin Summit)
tucson-teststand	tucson-teststand.lsst.codes (T&S/SITCom)
usdfprod	usdf-rsp.slac.stanford.edu (Production RSP at USDF)
usdfprod-prompt-processing	Prod for USDF Prompt Processing

6.3 Continuous Integration

We rely on Github actions to build containers for each branch or tag of a given application. In Phalanx a specific branch or tag can be used to test a specific new release of a given application. We normally also enable pre-commit to run linting and formatting (black[§]). Most Phalanx applications use tox[¶] for Python automation (testing, documentation building etc).

The science pipelines are built by Jenkins nightly.[19] For specific changes developers may invoke the *stack-os-matrix*[‖] to check changes will not break the nightly build. Weekly and stable releases are built from sources and packaged as a conda[**] environment as well as an Apptainer[††] container image specifically to be distributed via a software distribution network based on CERN's CernVM-FS[‡‡]. SLAC subscribes to this distribution channel

[*] https://helm.sh/
[†] https://phalanx.lsst.io
[‡] https://argo-cd.readthedocs.io/en/stable/
[§] https://black.readthedocs.io/en/stable/
[¶] https://tox.wiki/en/latest/index.html
[‖] https://developer.lsst.io/stack/jenkins-stack-os-matrix.html
[**] https://docs.conda.io
[††] https://apptainer.org
[‡‡] https://sw.lsst.eu

and makes available the Rubin software stack to pipeline developers, to individual users, as well as to tasks executing in its batch farm. Hosted by the French data facility, this distribution mechanism ensures all Rubin data facilities use a bit-by-bit identical copy of the software for producing the data releases and allows data centers where science collaborations will consume Rubin data products to get an always up-to-date release of the software (e.g., NERSC[*]).

7. OPEN ISSUES

We have not yet tested the client server butler between SLAC and Google. We have checked the bandwidth and latency is acceptable but it remains to be seen how it will work under the load of 10,000 users on day one of a data release.

Butler implements a client cache so a scientists notebook or script repeatedly accessing the same image will be well handled. But for the cloud connections we consider a general cache for images, this would only work if many scientists use the same image. We do not know the image access patterns that will occur in operations and hence are not sure if this will work or perhaps the client cache is adequate. We are considering permanently caching the deep coadds in the cloud since they will be accessed most frequently. Individual processed visit images may well never be amenable to caching.

We need to investigate how to get a consolidated view of key metrics and monitoring across the cloud and on-premise. Such a view would help immensely in tracking down issues which may appear in one part while the cause is elsewhere.

For user batch we have specified users would have to have SLAC accounts and log in to SLAC to submit jobs.[9] It would be user friendly to allow submission of batch jobs directly from the cloud hosted science platform which execute on SLAC. In principle we can do this but there are issues ranging from time allocation at SLAC to accessing of results which are not yet resolved.

8. CONCLUSION

The Vera C. Rubin Observatory will come into operations in 2025. The Rubin US data facility, hosted at SLAC, stores the Rubin data and processes it in conjunction with our French and UK partner facilities.

We have already run most of our scientist facing services on Google for three years. We currently support one thousand users on the science platform hosted on Google and are preparing for the many thousands of more users Rubin expects.

Transition to a hybrid model with the bulk of the data stored on premises at SLAC but maintaining the scientist facing services on Google is underway.

ACKNOWLEDGMENTS

This material is based upon work supported in part by the National Science Foundation through Cooperative Agreement AST-1258333 and Cooperative Support Agreement AST-1202910 managed by the Association of Universities for Research in Astronomy (AURA), and the Department of Energy under Contract No. DE-AC02-76SF00515 with the SLAC National Accelerator Laboratory managed by Stanford University. Additional Rubin Observatory funding comes from private donations, grants to universities, and in-kind support from LSSTC Institutional Members.

[*] https://www.nersc.gov

REFERENCES

[1] Ivezić, Ž., Kahn, S. M., Tyson, J. A., Abel, B., Acosta, E., Allsman, R., Alonso, D., AlSayyad, Y., Anderson, S. F., Andrew, J., Angel, J. R. P., Angeli, G. Z., Ansari, R., Antilogus, P., Araujo, C., Armstrong, R., Arndt, K. T., Astier, P., Aubourg, É., Auza, N., Axelrod, T. S., Bard, D. J., Barr, J. D., Barrau, A., Bartlett, J. G., Bauer, A. E., Bauman, B. J., Baumont, S., Bechtol, E., Bechtol, K., Becker, A. C., Becla, J., Beldica, C., Bellavia, S., Bianco, F. B., Biswas, R., Blanc, G., Blazek, J., Blandford, R. D., Bloom, J. S., Bogart, J., Bond, T. W., Booth, M. T., Borgland, A. W., Borne, K., Bosch, J. F., Boutigny, D., Brackett, C. A., Bradshaw, A., Brandt, W. N., Brown, M. E., Bullock, J. S., Burchat, P., Burke, D. L., Cagnoli, G., Calabrese, D., Callahan, S., Callen, A. L., Carlin, J. L., Carlson, E. L., Chandrasekharan, S., Charles-Emerson, G., Chesley, S., Cheu, E. C., Chiang, H.-F., Chiang, J., Chirino, C., Chow, D., Ciardi, D. R., Claver, C. F., Cohen-Tanugi, J., Cockrum, J. J., Coles, R., Connolly, A. J., Cook, K. H., Cooray, A., Covey, K. R., Cribbs, C., Cui, W., Cutri, R., Daly, P. N., Daniel, S. F., Daruich, F., Daubard, G., Daues, G., Dawson, W., Delgado, F., Dellapenna, A., de Peyster, R., de Val-Borro, M., Digel, S. W., Doherty, P., Dubois, R., Dubois-Felsmann, G. P., Durech, J., Economou, F., Eifler, T., Eracleous, M., Emmons, B. L., Fausti Neto, A., Ferguson, H., Figueroa, E., Fisher-Levine, M., Focke, W., Foss, M. D., Frank, J., Freemon, M. D., Gangler, E., Gawiser, E., Geary, J. C., Gee, P., Geha, M., Gessner, C. J. B., Gibson, R. R., Gilmore, D. K., Glanzman, T., Glick, W., Goldina, T., Goldstein, D. A., Goodenow, I., Graham, M. L., Gressler, W. J., Gris, P., Guy, L. P., Guyonnet, A., Haller, G., Harris, R., Hascall, P. A., Haupt, J., Hernandez, F., Herrmann, S., Hileman, E., Hoblitt, J., Hodgson, J. A., Hogan, C., Howard, J. D., Huang, D., Huffer, M. E., Ingraham, P., Innes, W. R., Jacoby, S. H., Jain, B., Jammes, F., Jee, M. J., Jenness, T., Jernigan, G., Jevremović, D., Johns, K., Johnson, A. S., Johnson, M. W. G., Jones, R. L., Juramy-Gilles, C., Jurić, M., Kalirai, J. S., Kallivayalil, N. J., Kalmbach, B., Kantor, J. P., Karst, P., Kasliwal, M. M., Kelly, H., Kessler, R., Kinnison, V., Kirkby, D., Knox, L., Kotov, I. V., Krabbendam, V. L., Krughoff, K. S., Kubánek, P., Kuczewski, J., Kulkarni, S., Ku, J., Kurita, N. R., Lage, C. S., Lambert, R., Lange, T., Langton, J. B., Le Guillou, L., Levine, D., Liang, M., Lim, K.-T., Lintott, C. J., Long, K. E., Lopez, M., Lotz, P. J., Lupton, R. H., Lust, N. B., MacArthur, L. A., Mahabal, A., Mandelbaum, R., Markiewicz, T. W., Marsh, D. S., Marshall, P. J., Marshall, S., May, M., McKercher, R., McQueen, M., Meyers, J., Migliore, M., Miller, M., Mills, D. J., Miraval, C., Moeyens, J., Moolekamp, F. E., Monet, D. G., Moniez, M., Monkewitz, S., Montgomery, C., Morrison, C. B., Mueller, F., Muller, G. P., Muñoz Arancibia, F., Neill, D. R., Newbry, S. P., Nief, J.-Y., Nomerotski, A., Nordby, M., O'Connor, P., Oliver, J., Olivier, S. S., Olsen, K., O'Mullane, W., Ortiz, S., Osier, S., Owen, R. E., Pain, R., Palecek, P. E., Parejko, J. K., Parsons, J. B., Pease, N. M., Peterson, J. M., Peterson, J. R., Petravick, D. L., Libby Petrick, M. E., Petry, C. E., Pierfederici, F., Pietrowicz, S., Pike, R., Pinto, P. A., Plante, R., Plate, S., Plutchak, J. P., Price, P. A., Prouza, M., Radeka, V., Rajagopal, J., Rasmussen, A. P., Regnault, N., Reil, K. A., Reiss, D. J., Reuter, M. A., Ridgway, S. T., Riot, V. J., Ritz, S., Robinson, S., Roby, W., Roodman, A., Rosing, W., Roucelle, C., Rumore, M. R., Russo, S., Saha, A., Sassolas, B., Schalk, T. L., Schellart, P., Schindler, R. H., Schmidt, S., Schneider, D. P., Schneider, M. D., Schoening, W., Schumacher, G., Schwamb, M. E., Sebag, J., Selvy, B., Sembroski, G. H., Seppala, L. G., Serio, A., Serrano, E., Shaw, R. A., Shipsey, I., Sick, J., Silvestri, N., Slater, C. T., Smith, J. A., Smith, R. C., Sobhani, S., Soldahl, C., Storrie-Lombardi, L., Stover, E., Strauss, M. A., Street, R. A., Stubbs, C. W., Sullivan, I. S., Sweeney, D., Swinbank, J. D., Szalay, A., Takacs, P., Tether, S. A., Thaler, J. J., Thayer, J. G., Thomas, S., Thornton, A. J., Thukral, V., Tice, J., Trilling, D. E., Turri, M., Van Berg, R., Vanden Berk, D., Vetter, K., Virieux, F., Vucina, T., Wahl, W., Walkowicz, L., Walsh, B., Walter, C. W., Wang, D. L., Wang, S.-Y., Warner, M., Wiecha, O., Willman, B., Winters, S. E., Wittman, D., Wolff, S. C., Wood-Vasey, W. M., Wu, X., Xin, B., Yoachim, P., and Zhan, H., "LSST: From Science Drivers to Reference Design and Anticipated Data Products," *ApJ* **873**, 111 (Mar 2019). DOI: https://doi.org/10.3847/1538-4357/ab042c.

[2] O'Mullane, W., Economou, F., Huang, F., Speck, D., Chiang, H.-F., Graham, M. L., Allbery, R., Banek, C., Sick, J., Thornton, A. J., Masciarelli, J., Lim, K.-T., Mueller, F., Padolski, S., Jenness, T., Krughoff, K. S., Gower, M., Guy, L. P., and Dubois-Felsmann, G. P., "Rubin Science Platform on Google: the story so far," *arXiv e-prints* , arXiv:2111.15030 (Nov. 2021). DOI: https://doi.org/10.48550/arXiv.2111.15030.

[3] Jenness, T., Irving, D. H., Bosch, J., Salnikov, A., and Lust, N. B., "Converting Rubin Observatory's Data Butler to a client/server architecture," in [*Software and Cyberinfrastructure for Astronomy VIII*], Ibsen, J.

and Chiozzi, G., eds., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **13101**, 13101–129 in press (2024). https://dx.doi.org/doi#here.

[4] Selvy, B. M., Claver, C., Willman, B., Petravick, D., Johnson, M., Reil, K., Marshall, S., Thomas, S., Lotz, P., Schumacher, G., Lim, K.-T., Jenness, T., Jacoby, S., Emmons, B., and Axelrod, T., "Using model based systems engineering for the development of the Large Synoptic Survey Telescope's operational plan," in [*Modeling, Systems Engineering, and Project Management for Astronomy VI*], *Proc. SPIE* **9911**, 99110D (Aug. 2016). DOI: https://doi.org/10.1117/12.2233904.

[5] Dubois-Felsmann, G. and Jenness, T., "Data Management System (DMS) Requirements," (Dec. 2019). Vera C. Rubin Observatory LSE-61, https://lse-61.lsst.io/.

[6] O'Mullane, W. and Blum, R., "Statement of Work for the Rubin Observatory US Data Facility," (Apr. 2024). Vera C. Rubin Observatory Technical Note RTN-080, https://rtn-080.lsst.io/.

[7] O'Mullane, W., Willman, B., Graham, M., Guy, L., Blum, R., and Marshall, P., "Guidelines for Rubin Independent Data Access Centers," (Aug. 2021). Vera C. Rubin Observatory Technical Note RTN-003, https://rtn-003.lsst.io/.

[8] Bosch, J., AlSayyad, Y., Armstrong, R., Bellm, E., Chiang, H.-F., Eggl, S., Findeisen, K., Fisher-Levine, M., Guy, L. P., Guyonnet, A., Ivezić, Ž., Jenness, T., Kovács, G., Krughoff, K. S., Lupton, R. H., Lust, N. B., MacArthur, L. A., Meyers, J., Moolekamp, F., Morrison, C. B., Morton, T. D., O'Mullane, W., Parejko, J. K., Plazas, A. A., Price, P. A., Rawls, M. L., Reed, S. L., Schellart, P., Slater, C. T., Sullivan, I., Swinbank, J. D., Taranu, D., Waters, C. Z., and Wood-Vasey, W. M., "An Overview of the LSST Image Processing Pipelines," in [*Astronomical Data Analysis Software and Systems XXVII*], Teuben, P. J., Pound, M. W., Thomas, B. A., and Warner, E. M., eds., *Astronomical Society of the Pacific Conference Series* **523**, 521 (2019). DOI: https://doi.org/10.48550/arXiv.1812.03248.

[9] O'Mullane, W., "User batch - possibilities and plans.," (July 2023). Vera C. Rubin Observatory Data Management Technical Note DMTN-223, https://dmtn-223.lsst.io/.

[10] Dubois-Felsmann, G., Economou, F., Lim, K.-T., Mueller, F., Pietrowicz, S. R., and Wu, X., "Science Platform Design," (Jan. 2019). Vera C. Rubin Observatory Data Management Controlled Document LDM-542, https://ldm-542.lsst.io/.

[11] Wang, D. L., Monkewitz, S. M., Lim, K.-T., and Becla, J., "Qserv: a distributed shared-nothing database for the lsst catalog," in [*State of the Practice Reports*], *SC '11*, 12:1–12:11, ACM, New York, NY, USA (2011). http://doi.acm.org/10.1145/2063348.2063364.

[12] Mueller, F., Gaponenko, I., Gates, J., Hanushevsky, A., Jammes, F., Lim, K.-T., Salnikov, A., , and Slater, C. T., "Qserv: A Distributed Petascale Database for the LSST Catalogs," (Nov. 2022). Vera C. Rubin Observatory Data Management Technical Note DMTN-243, https://dmtn-243.lsst.io/.

[13] Jenness, T., Bosch, J. F., Salnikov, A., Lust, N. B., Pease, N. M., Gower, M., Kowalik, M., Dubois-Felsmann, G. P., Mueller, F., and Schellart, P., "The Vera C. Rubin Observatory Data Butler and pipeline execution system," in [*Software and Cyberinfrastructure for Astronomy VII*], *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **12189**, 1218911 (Aug. 2022). DOI: https://doi.org/10.1117/12.2629569.

[14] Dubois-Felsmann, G., "Technical definition of DP0.3 (Solar System data)," (Jan. 2023). Vera C. Rubin Observatory Technical Note RTN-050, https://rtn-050.lsst.io/.

[15] Maeno, T., Alekseev, A., Barreiro Megino, F. H., De, K., Guan, W., Karavakis, E., Klimentov, A., Korchuganova, T., Lin, F., Nilsson, P., Wenaus, T., Yang, Z., and Zhao, X., "PanDA: Production and Distributed Analysis System," *Computing and Software for Big Science* **8**, 4 (Jan. 2024). DOI: https://doi.org/10.1007/s41781-024-00114-3.

[16] Karavakis, E., Guan, W., Yang, Z., Maeno, T., Wenaus, T., Adelman-McCarthy, J., Barreiro Megino, F., De, K., Dubois, R., Gower, M., Jenness, T., Klimentov, A., Korchuganova, T., Kowalik, M., Lin, F.-H., Nilsson, P., Padolski, S., Yang, W., and Ye, S., "Integrating the PanDA Workload Management System with the Vera C. Rubin Observatory," *arXiv e-prints* , arXiv:2312.04921 (Dec. 2023). DOI: https://doi.org/10.48550/arXiv.2312.04921.

[17] O'Mullane, W., Allbery, R., AlSayyad, Y., Bellm, E., Clements, A., Dubois, R., Hoblitt, J., Silva, C., Sullivan, I., Lim, K.-T., oversight: Ashley Zauderer-Vanderley (NSF), P. M. A., and (DOE), K. T., "Rubin Observatory Data Security Standards Implementation," (June 2023). Vera C. Rubin Observatory Data Management Technical Note DMTN-199, https://dmtn-199.lsst.io/.

[18] Fausti Neto, A., Thornton, A., Reuter, M., and Economou, F., "Sasquatch: Rubin Observatory metrics and telemetry service," in [*Software and Cyberinfrastructure for Astronomy VIII*], Ibsen, J. and Chiozzi, G., eds., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **13101**, 13101–59, in press (2024). https://dx.doi.org/doi#here.

[19] Jenness, T., Economou, F., Findeisen, K., Hernandez, F., Hoblitt, J., et al., "LSST data management software development practices and tools," in [*Software and Cyberinfrastructure for Astronomy V*], *Proc. SPIE* **10707**, 1070709 (July 2018). DOI: https://doi.org/10.1117/12.2312157.

APPENDIX A. ACRONYMS

Acronym	Description
ADASS	Astronomical Data Analysis Software and Systems
AMD	Advanced Micro Devices
API	Application Programming Interface
AST	NSF Division of Astronomical Sciences
ATLAS	A Toroidal LHC Apparatus
AURA	Association of Universities for Research in Astronomy
CERN	European Organization for Nuclear Research
CI	Continuous Integration
CNPG	Cloud Native Postgres
CPU	Central Processing Unit
DAC	Data Access Center
DE	dark energy
DMSR	DM System Requirements; LSE-61
DMTN	DM Technical Note
DOE	Department of Energy
EFD	Engineering and Facility Database
ESNet	Energy Sciences Network
FS	File System
FTS3	File Transfer Service 3
GB	Gigabyte
GCP	Google Cloud Platform
HEP	High Energy Physics
IDF	Interim Data Facility
IVOA	International Virtual-Observatory Alliance
JSON	JavaScript Object Notation
LDM	LSST Data Management (Document Handle)
LHC	Large Hadron Collider (at CERN)
LSE	LSST Systems Engineering (Document Handle)
LSST	Legacy Survey of Space and Time (formerly Large Synoptic Survey Telescope)
LSSTC	LSST Corporation
NERSC	National Energy Research Scientific Computing Center
POSIX	Portable Operating System Interface
PanDA	Production ANd Distributed Analysis system
RAM	Random Access Memory
RSP	Rubin Science Platform
RTN	Rubin Technical Note

S3	(Amazon) Simple Storage Service
S3DF	SLAC Shared Scientific Data Facility
SLAC	SLAC National Accelerator Laboratory
SQuaRE	Science Quality and Reliability Engineering
SRCF	Stanford Research Computing Facility
SSD	Solid-State Disk
UK	United Kingdom
URL	Universal Resource Locator
US	United States
USDF	United States Data Facility