Data Movement Model for the Vera C. Rubin Observatory

Fabio Hernandez[1], *Mark G.* Beckett[2], *Andrew* Hanushevsky[3], *Tim* Jenness[4], *Kian-Tat* Lim[3], *Peter* Love[5], *Timothy* Noble[6], *Stephen* Pietrowicz[7], and *Wei* Yang[3]

[1]CNRS, CC-IN2P3, 21 avenue Pierre de Coubertin, CS70202, F-69627 Villeurbanne cedex, France
[2]Institute for Astronomy, University of Edinburgh, Royal Observatory, Blackford Hill, Edinburgh EH9 3HJ, UK
[3]SLAC National Accelerator Laboratory, 2575 Sand Hill Rd., Menlo Park, CA 94025, USA
[4]Vera C. Rubin Observatory Project Office, 950 N. Cherry Ave., Tucson, AZ 85719, USA
[5]Lancaster University, Lancaster, UK
[6]Science and Technology Facilities Council, Rutherford Appleton Laboratory, Harwell, UK
[7]NCSA, University of Illinois at Urbana-Champaign, 1205 W. Clark St., Urbana, IL 61801, USA

Abstract. The sky images captured nightly by the camera on the Vera C. Rubin Observatory's telescope will be processed across facilities on three continents. Data acquisition will occur at the observatory's location on Cerro Pachón in the Andes mountains of Chile. A first copy of the raw image data set is stored at the summit and immediately transmitted via dedicated network links to the archive site and the US Data Facility at SLAC National Accelerator Laboratory in California. After a brief embargo period, the full dataset is transferred to the France Data Facility, where a third copy is maintained, and a partial dataset is transferred to the UK Data Facility.

Over its 10-year operational period, beginning in late 2025, annual processing campaigns will be conducted by the three facilities on all images collected to date. Sophisticated algorithms will extract measurements of celestial objects from these images, producing science-ready images and catalogs. Data products resulting from these processing campaigns will be sent to SLAC for integration into a consistent Data Release, which will be made available to the scientific community through Data Access Centers in the US and Chile, as well as Independent Data Access Centers elsewhere.

In this paper we present an overall view of how we leverage the tools selected for managing the movement of data among the Rubin processing and serving facilities, including Rucio and FTS. We will also present the tools we developed to integrate Rucio's data model and Rubin's Data Butler, the software abstraction layer that mediates all access to storage by pipeline tasks that implement science algorithms.

1 Introduction

The Vera C. Rubin Observatory's mission is to explore the universe by conducting the *Legacy Survey of Space and Time* (LSST), the largest-ever sky survey with an unprecedented wide-field imaging system. The observatory aims to capture deep, high-resolution images of the night sky, mapping the cosmos to investigate fundamental questions in astrophysics [1].

The sky images captured nightly by the observatory's 3.2-gigapixel camera covering the wavelength range 320–1050 nm will be processed across facilities on three continents. Data acquisition will occur at the observatory's location on Cerro Pachón in the Andes mountains of Chile. A first copy of the raw image data is stored at the summit and immediately transmitted via dedicated network links to the archive site and the US Data Facility at SLAC National Accelerator Laboratory in California (see Fig. 1). After a brief embargo period, the full dataset is transferred to the France Data Facility, where a third copy is maintained, and a partial dataset is transferred to the UK Data Facility.

Over its 10-year operational period, beginning in late 2025, annual processing campaigns will be conducted by the three facilities on all images collected to date. Sophisticated algorithms will extract measurements of celestial objects from these images, producing science-ready images and catalogs. Data products resulting from these processing campaigns will be sent to SLAC for integration into a consistent Data Release, which will be made available to the scientific community through Data Access Centers in the US and Chile, as well as Independent Data Access Centers elsewhere.

The remainder of this paper is structured as follows. We present in section 2 the main data movement use cases we need to satisfy and in section 3 the tools that have been selected or developed and how they are composed to implement solutions to those use cases.

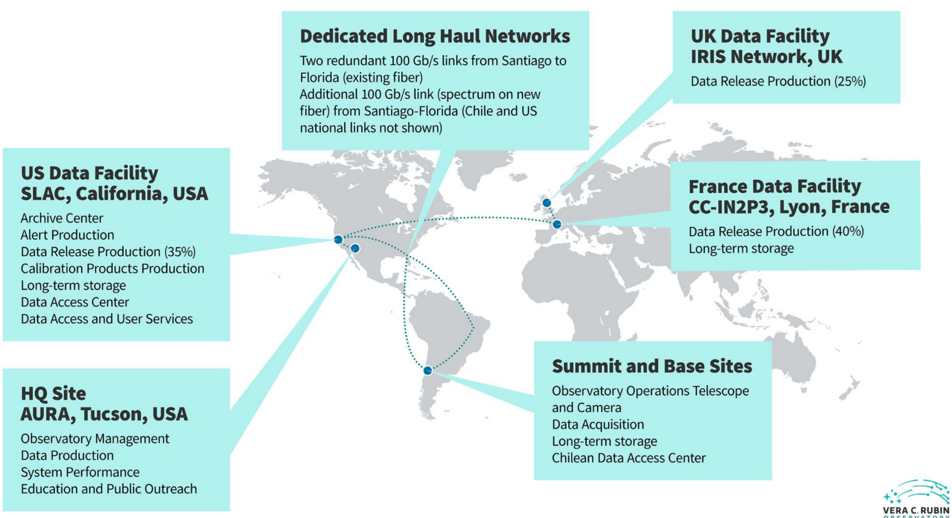

Figure 1. Raw images flow from the Summit Site, where the telescope is located in Chile, to the Base Site and then to the Archive Center at SLAC through long haul network links specifically deployed for the needs of the Observatory. Data is transferred from the Archive Center to the European Data Facilities for archival and processing. The US, UK and France Data Facilities collectively provide the computational capacity for processing the images taken by the Observatory for the duration of the survey. The Observatory headquarters are located in Tucson, USA.

2 Data movement use cases

A dataset of about 5 PB of new image data will be recorded by the instrument every year, for a total of 50 PB of raw data accumulated over the duration of the survey. Processing the input dataset for the purpose of producing a data release generates approximately ten times the size of the input dataset, including intermediate datasets not part of the the published release.

This section presents three distinct use cases for moving data among the data facilities used by the Rubin Observatory.

2.1 From summit to archive

The data acquisition system stores each exposure as a set of approximately 200 files, one per sensor on the camera focal plane. Once an exposure is recorded at the summit site, its constituent files are transferred in parallel to an object store at the archive center via the S3 protocol [2]. To optimize these transfers over the international network linking the summit to the SLAC archive site, we employ specialized network connection pooling, keep-alive mechanisms, and TCP tuning.

Given that raw images undergo prompt processing for transient object detection and alert generation, the target end-to-end latency for transferring a single exposure—including data compression and other overheads—is set to seven seconds for four gigabytes of compressed data.

Ancillary data (e.g., telemetry, specialized databases) are replicated to the archive center using native protocols to avoid translation steps that can add latency and complexity. Additionally, a small number of certified calibration files are transferred infrequently from the archive to the summit and other locations.

2.2 From archive to processing facilities and back

Annual processing of the entire image dataset recorded since the beginning of the survey is carried out across three facilities: the US Data Facility, hosted at SLAC National Accelerator Laboratory in California, USA[1], the France Data Facility, hosted by the IN2P3 computing center (CC-IN2P3) in Lyon, France[2], and the UK Data Facility, operated by the LSST:UK consortium[3].

Raw image data is replicated from the US to the European facilities. Both the US and France data facilities store a complete copy of the raw image dataset. The UK facility receives the raw images corresponding to the spatial region assigned to it for processing. Data movement between these sites is facilitated by ESnet[4], which handles transatlantic data transport; GEANT[5], which connects European sites; and the national research and education networks, JANET[6] (UK) and RENATER[7] (France).

The entire set of final data products, along with selected intermediate products from each campaign, is replicated from the facility where they are generated to the archive center. There, they are consolidated and incorporated into a new data release, which is delivered annually to the science community for analysis [3].

The LSST Science Pipelines is the software developed by Rubin Observatory to process the survey data [4]. It includes advanced image processing algorithms and supporting middleware. A central component of this middleware is the Rubin Data Butler, an abstraction layer that mediates access to the data required by, or generated through, the pipelines

[1] https://www.slac.stanford.edu
[2] https://cc.in2p3.fr
[3] https://www.lsst.ac.uk
[4] https://es.net
[5] https://geant.org
[6] https://www.jisc.ac.uk/janet
[7] https://renater.fr

[5]. The Data Butler retrieves data from persistent storage (using appropriate protocols and data formats) based on queries specified by scientifically relevant identifiers (rather than file paths), and delivers the data as in-memory Python objects to the pipelines. It also persists the in-memory objects generated by the science algorithms. Crucially, the Butler manages the location of all files within the data store, recording their locations and relationships in a relational database. Together, the file registry and the storage system where files are located constitute a *repository*.

Since a given Butler repository is aware only of the files present at a single facility, files replicated between facilities need to be placed in the repository's data store at the location expected by the Butler. Upon reception, replicated files are ingested into the receiving facility's local Butler repository, making them available for the processing pipelines.

2.3 From archive to data access centers

Annually released data products must be distributed to approximately 15 to 20 data access centers across the Americas, Europe, and Asia-Pacific regions, where scientific analysis will be conducted. These distribution campaigns will be centrally coordinated by Rubin to ensure timely delivery of data releases to all analysis centers. The goal is to distribute the multi-petabyte datasets to the data access centers in a tiered manner, with some centers receiving data directly from the archive center and then sending the data on to other data access centers, thereby reducing the load on the archive center [6].

3 Data movement tools

Several software tools are employed to implement the use cases outlined in the previous section. CERN's Rucio [7] and its companion FTS [8] manage the movement of files between the archive site and data facilities, as well as from the archive to the data access centers. In addition, Rubin-specific tools have been developed to register files and automate actions when replicated files arrive at their destination. These tools and their usage are described in the following subsections.

Rubin Observatory operates a dedicated instance of Rucio, configured to transfer files between the Rucio Storage Elements (RSEs) at each facility. These storage endpoints support a data movement protocol that Rucio utilizes to transport data across them. The US and UK data facilities use XrootD [9], while the France data facility uses dCache [10]. All of these systems expose the webDAV protocol [11], an extension of HTTP [12]. Data is transferred securely across sites using confidential channels built on top of secure HTTP.

Each processing facility exposes at least two RSEs: one for storing input data required for processing (e.g., raw images, calibration data, reference catalogs, etc.) and another endpoint for storing the products generated by the image processing pipelines [13]. Data stored by the input data RSE is protected against modification and removal and is also archived to tape. Data products stored in the products RSE are less sensitive as they can be regenerated and even some of them may be deleted after a processing campaign is complete. All RSEs are configured to use the identity logical-to-physical filename mapping. This configuration ensures that the file pathnames are preserved relative to the Butler repository's datastore location, which is critical for proper file replication to the destination where the Butler expects to find them.

3.1 Registration of files to replicate

To perform replication, we create Rucio Datasets, each composed of a set of files that are already in their appropriate locations at the source RSE. Upon registration, preconfigured Rucio subscriptions trigger the actual file movement to the destination facility, in accordance with the defined replication rules. Rucio delegates the execution of file transfers to FTS, which then instructs the storage endpoints at the facilities to move the data, typically by requesting the destination facility to pull the data from the source facility. The use of Datasets allows grouping of related files, use of subscription patterns applied to spatially-defined Dataset names to associate spatial regions with RSEs, and a clear way to know when all related files have been generated (via Dataset closure) and replicated.

Rubin has developed the tool `rucio_register`[8], which allows for the selection of existing files from a Butler repository based on specified criteria. The tool attaches Rubin-specific metadata to these files and registers them into one or more Rucio Datasets. The metadata, encoded as a JSON record, contains a minimal set of information extracted from the origin Butler repository. This ensures that replicated files can be ingested properly into the local Butler repository at the destination facility.

Only files that require replication to another facility are registered with Rucio. As a result, Rubin's instance of Rucio is aware only of files replicated across processing facilities. Files that are local to each facility and not subject to replication remain known only to that facility's Butler and are not registered in Rucio. Since the US Data Facility gets a complete copy of all final data products, by definition files that are not replicated are intermediates in the calculations that are not required to be persisted.

The pipeline processing generates many ancillary files in addition to pixel data. A data preview processing run [14] demonstrated that the number of JSON and YAML files is approximately of the same scale as the number of FITS and Parquet data files (see Fig. 2). Given that the ancillary files are significantly smaller (sometimes a few kB per file) this can lead to very large file transfer overheads. To mitigate this problem we have modified the Butler infrastructure to allow the small files from a single processing run to be combined into one or more Zip files. These Zip files contain the Butler metadata necessary to allow the Butler to retrieve individual files whilst making a single file available to Rucio.

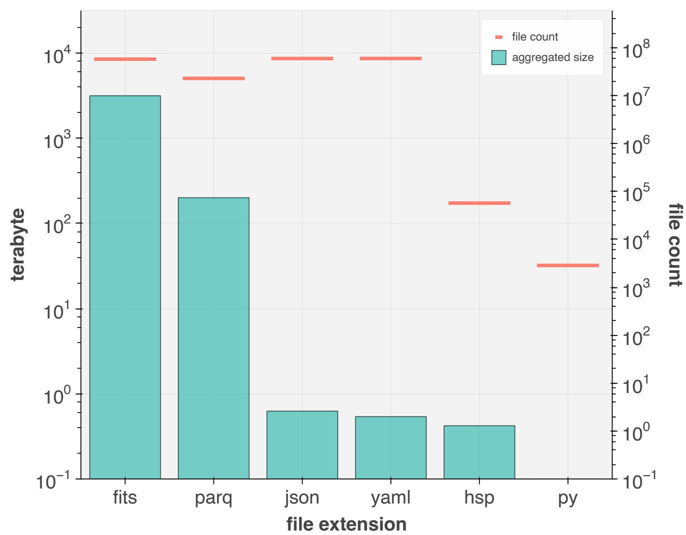

Figure 2. Number of files and total file sizes from a data preview processing run.

[8] https://github.com/lsst/rucio_register

3.2 Ingestion at reception

FTS notifies Rucio about the completion of individual file transfers. Rubin's `HermesK`[9], which is a modification of Rucio's Hermes daemon, filters messages and uses Kafka as a mechanism to signal the destination Rubin facility that a new file was replicated and to take appropriate actions. Kafka was selected as a reliable message bus used for other purposes within the Rubin project [see e.g., 15, 16]. Its ordering guarantees are not strictly necessary in this application, but the ability to scale to multiple consumers may be needed as the number of files increases.

Messages distributed through Rubin's Kafka control-plane include Rubin-specific metadata. Those messages are received by Rubin's `ingestd`[10], a daemon running at each destination facility responsible for ingesting newly replicated files into the local Butler repository.

Each facility only receives notifications about files successfully replicated to the storage endpoints it operates. This is achieved by following a simple convention: the name of Kafka topic the notification is sent to is identical to the name of the Rucio storage element. Each facility's `ingestd` is configured to only monitor Kafka messages specifically targeted to the facility's RSEs (see Fig. 3).

In a complex distributed system such as this, having stateless daemons such as `ingestd`, idempotent transactions such as ingestion of file batches, and triggering off known synchronization points such as replication acknowledgement helps ensure a consistent, if conservative, view of the available data across multiple sites.

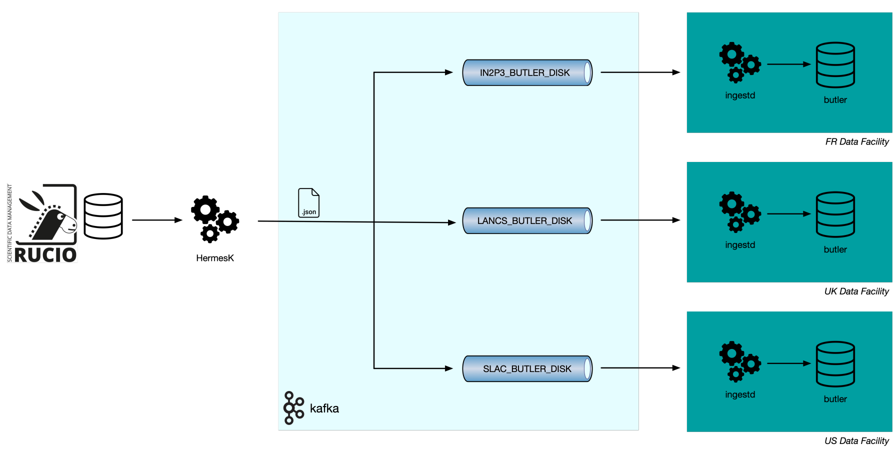

Figure 3. `HermesK` emits notifications about successful file transfers via Kafka topics named after the destination RSE. At the receiving facility `ingestd` monitors those notifications and ingests the newly received file into the local Butler repository. The JSON-encoded, Rubin-specific metadata associated to the file when it was first registered into Rucio contains the details needed for ingestion.

4 Summary

We presented several use cases for the movement of data among the facilities participating to processing of Rubin Observatory data, the tools used to implement solutions to satisfy those

[9] https://github.com/lsst-dm/ctrl_rucio_ingest
[10] https://github.com/lsst-dm/ctrl_ingestd

uses cases as well as the tools Rubin has developed for integrating Rubin-specific software components to more generic software systems for large scale inter-site data transfer.

5 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.

This work has been supported by the UK Science and Technology Facilities Council (STFC) funding for UK participation in LSST, through grants ST/X001334/1 and ST/Y003004/1.

References

[1] Ž. Ivezić et al., LSST: From Science Drivers to Reference Design and Anticipated Data Products, ApJ **873**, 111 (2019), `arXiv:0805.2366`. 10.3847/1538-4357/ab042c

[2] Amazon S3 REST API introduction, https://docs.aws.amazon.com/AmazonS3/latest/API/Welcome.html

[3] F. Hernandez, G. Beckett, P. Clark, M. Doidge, T. Jenness, E. Karavakis, Q. Le Boulc'h, P. Love, G. Mainetti, T. Noble et al., Overview of the distributed image processing infrastructure to produce the Legacy Survey of Space and Time, EPJ Web of Conf. **295**, 01042 (2024). 10.1051/epjconf/202429501042

[4] J. Bosch et al., An Overview of the LSST Image Processing Pipelines, in *Astronomical Data Analysis Software and Systems XXVII*, edited by P.J. Teuben, M.W. Pound, B.A. Thomas, E.M. Warner (2019), Vol. 523 of *ASP Conf. Ser.*, p. 521, `arXiv:1812.03248`. 10.48550/arXiv.1812.03248

[5] T. Jenness, J.F. Bosch, A. Salnikov, N.B. Lust, N.M. Pease, M. Gower, M. Kowalik, G.P. Dubois-Felsmann, F. Mueller, P. Schellart, The Vera C. Rubin Observatory Data Butler and pipeline execution system, in *"Software and Cyberinfrastructure for Astronomy VII"* (2022), Vol. 12189 of *Proc. SPIE*, p. 1218911, `arXiv:2206.14941`. 10.1117/12.2629569

[6] A. Bolton, RTN-086: Bulk Data Transfer Policies and Procedures. (2024), Vera C. Rubin Observatory Technical Note, `https://rtn-086.lsst.io`

[7] M. Barisits, T. Beermann, F. Berghaus, B. Bockelman, J. Bogado, D. Cameron, D. Christidis, D. Ciangottini, G. Dimitrov, M. Elsing et al., Rucio: Scientific Data Management, Computing and Software for Big Science **3**, 11 (2019). 10.1007/s41781-019-0026-3

[8] File Transfer Service, `https://fts.web.cern.ch/fts`

[9] XRootD, `https://xrootd.github.io`

[10] T. Mkrtchyan, K. Chitrapu, V. Garonne, D. Litvintsev, S. Meyer, P. Millar, L. Morschel, A. Rossi, M. Sahakyan, dCache: Inter-disciplinary storage system, EPJ Web Conf. **251**, 02010 (2021). 10.1051/epjconf/202125102010

[11] L. Dusseault, HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV), RFC-4918, `https://www.ietf.org/rfc/rfc4918.txt`

[12] R. Fielding, J. Mogule, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, Hypertext Transfer Protocol – HTTP/1.1, RFC-2616, https://datatracker.ietf.org/doc/html/rfc2616

[13] K.T. Lim, Multi-Site Data Release Processing Using PanDA and Rucio. (2022), DMTN-213: Vera C. Rubin Observatory Data Management Technical Note, https://dmtn-213.lsst.io

[14] Q. Le Boulc'h, F. Hernandez, G. Mainetti, The Rubin Observatory's Legacy Survey of Space and Time DP0.2 processing campaign at CC-IN2P3, EPJ Web of Conf. **295**, 04049 (2024). 10.1051/epjconf/202429504049

[15] A. Fausti Neto, F. Economou, M.A. Reuter, J. Sick, R. Allbery, A.J. Thornton, Sasquatch: Rubin Observatory metrics and telemetry service, in *Software and Cyberinfrastructure for Astronomy VIII*, edited by J. Ibsen, G. Chiozzi (2024), Vol. 13101 of *Proc. SPIE*, p. 131011M. 10.1117/12.3019081

[16] T. Ribeiro, R.E. Owen, D.J. Mills, M.A. Reuter, A.W. Clements, W. O'Mullane, Replacing DDS with Apache Kafka as middleware technology for the Rubin Observatory control system, in *Software and Cyberinfrastructure for Astronomy VIII*, edited by J. Ibsen, G. Chiozzi (2024), Vol. 13101 of *Proc. SPIE*, p. 1310118. 10.1117/12.3020002