Vera C. Rubin Observatory
Systems Engineering

Project-wide Documentation Proposal for Rubin Observatory Operations

(Report from the Rubin Observatory Documentation Working Group)

Matthew Rumore, Chuck Claver, David Cabrera, Rob McKercher, John Andrew, Diane Hascall, Patrick Ingraham, Tony Johnson, Kristen Metzger, Austin Roberts, Jonathan Sick.

SITCOMTN-014

Latest Revision: 2024-04-02

Abstract

This technical note is a report to recommend a future state for Rubin Observatory Operations documentation by the Project-wide Documentation Working Group. This proposal presents a high-level documentation strategy for Rubin Observatory Operations with suggested methodologies to transition Construction Project documents and organizations for the technical documentation package needed to establish construction completeness and operational readiness. It responds to charge items 3 in the *Charge to the Documentation Working Group*, LSE-489. Migration plans, schedule and resources to migrate to the future state for this documentation will be reported in another techincal report, RTN-076.

Change Record

Version	Date	Description	Owner name
0.0	2021-06-26	First draft of preliminary content.	Chuck Claver
0.1	2021-08-30	Input to Documentation Portal.	Jonathan Sick
0.2	2021-09-14	Preliminary content, Portal architecture and effort.	Chuck Claver, Jonathan Sick
0.3	2022-11-18	Incorporate draft from Confluence pages and shared files as pre-draft.	David Cabrera, Matthew Rumore
0.4	2024-03-20	Finalize roadmap; transfer implementation content to RTN-076.	Matthew Rumore, Leanne Guy

Document curator: Matthew Rumore

Document source location: `https://github.com/lsst-sitcom/sitcomtn-014`

Contents

Project-wide Documentation Proposal for Rubin Observatory Operations

1 Overview

This document serves as a roadmap developed by the Vera C. Rubin Observatory Project-wide Documentation Working Group to fulfill one part of its charge, defined in LSE-489. Included are recommendations and suggestions for a systematic structure and methodology within the Rubin Observatory operations to create, update and release different types of technical documentation; collect them from the technical groups across the different systems within Rubin Observatory; classify the information by system, purpose and users' needs; and, create or modify tools to manage and retrieve information with reliability within a flexible and extendable system. The Documentation Working Group considered the wide berth of topics and technical areas, their differing requirements and processes, the variety of produced information, and large, diverse population involved with Rubin Observatory. The strategies in this proposal aim to support Rubin Observatory staff to reliably and confidently demonstrate operations readiness, SITCOMTN-005, and execute operational expectations throughout operations — lasting impact on science, strong work safety and environmentally sustainable culture with continuous improvement, and integrated and diverse community interaction.

To guide the proposal, the Documentation Working Group reported in SITCOMTN-012 on its effort to create an inventory of existing documentation and repositories, predominantly focused on the construction projects: the National Science Foundation (NSF) major research equipment and facilities construction (MREFC) effort managed by the LSST Corporation (LSSTC), and the Department of Energy (DOE) major items of equipment (MIE) effort for the Legacy Survey of Space and Time (LSST) Camera (LSSTCam) managed by SLAC National Accelerator Laboratory. Information is located on different official and non-official repositories, some considered temporary since they will not be continually supported throughout operations. This proposal includes steps to transition documentation from the construction, commissioning and pre-operations efforts into operations within an integrated, project-wide approach.

An evaluation of the level of effort and resources needed to realize this proposal is presented in a separate document, RTN-076. Risks and mitigations will be managed by the Rubin Operations Risk Board and managed using the Rubin Risk Tool, RDO-71.

Section 2 of this report includes four *Documentation Views* developed to create an efficient internal structure that bases its approach on the users, staff and communities interested in accessing the information. Their primary intent is to provide a systematic methodology to record, classify and categorize documentation, catalog their respective repositories, define ownership and target audiences, and organize relationships and dependencies between documents. Crucially, this structure allows individual groups to identify definitive sources of information, i.e., the normative source of information, thereby supporting an orderly way to reference information project-wide throughout operations without depending on an individual's system-specific knowledge. The normative source of information is the primary location where a piece of information is documented, and this location shall be used to reference and source that information to ensure the information is consistently and accurately . The Documentation Working Group intends that the implementation of the *Documentation Views* constitute a deliverable from the construction project to Rubin Operations and form part of construction completeness and operations readiness. It is critical that this effort and responsibility is shared between construction project subsystems, pre-operations and operations technical staff, particularly during knowledge transfer and documentation strategy transition stages. The managing groups will complete the Views such that their documentation are organized in a common way and operations staff are capable of managing the content before taking ownership. It is expected that no content currently used in construction or pre-operations will be lost.

The Documentation Views will power the proposed *Rubin Documentation Portal*, a web application to provide access and discovery to different documents and documentation types and assist users in retrieving information. The proposed architecture includes products already implemented for Rubin Observatory, namely the LSST the Docs (LTD) documentation delivery platform, SITCOMTN-012. The Documentation Working Group expects that the majority of effort for software developers will be to understand the use-cases and how various groups will utilize the *Documentation Views*, and to develop software tools to serve the Documentation Portal via repository metadata. The new Rubin Documentation Portal is detailed in Section 3.

Information from construction should be maintained as-is and available for any future needs; however, it is not necessary to incorporate everything into the future documentation scheme described in this proposal. Project and team managers with product owners should apply a graded approach to this proposal's recommendations and suggestions, preferably tailoring how they create and manage the Views and workflows for their ability to maintain reliable information throughout operations without assuming or requiring a large amount of institutional or system-specific knowledge. A process should be devised to determine if information should be transitioned or archived. A goal of the Documentation Working Group was for a system that does not require teams to recreate, reproduce or replace already useful documentation, but as a team transitions their documentation, they should review when that is advantageous.

This proposal is meant to provide guidance and framework for staff responsible for technical information to improve effective communication to internal and external stakeholders, and the Documentation Views must be created for their specific needs. If people are having difficulty finding information, determining where the source of truth is located, or managers/auditors cannot determine if construction completeness is being achieved, it's appropriate for the technical group to reflect on how they can expand on using the Documentation Views for improvement. While documentation specialist staff will assist in the development and implementation, technical staff that own the information are primarily responsible, and stakeholders and customers should provide feedback. The Documentation Working Group made an extensive effort to use currently available information for direct application across a small set of systems to validate this proposal. Examples of this work are presented herein, and they may be used as an initial starting point or adapted for those served by Rubin Observatory. The Documentation Working Group highly encourages exploiting software to retrieve information from repositories to help or automatically build the Documentation Views.

2 Documentation Views

There are four *Documentation Views* with a set of customized categories and subcategories developed by the Documentation Working Group to functionally describe and organize the technical documentation for the Rubin Observatory. The views and categories will assist those responsible for technical documentation when considering the informational needs and interests of users, communities, operations staff, and other stakeholders. Each view is represented by a *tree structure*, a widely used way of representing hierarchical structures typically in a graphical form. Examples in this proposal include classical node-link diagrams and tree view outlines that are built with *branches* of connected nodes, or *leaf-nodes*, starting from *root-nodes* representing the highest level in the hierarchy. Different tree structure diagrams can be used depending on the particular use case and how to effectively represent the information. (Wikimedia Foundation, 2021)

Each View has its own purpose and they are designed to provide a framework structure to search, reference and retrieve currently available information consistently project-wide. In addition, the Views will provide an orderly way to introduce new documentation and meet an extended goal to facilitate referencing and minimize replication.

The four views are:

- **Product View** — For organizing the ownership of documentation, describing internal systems and providing structure for linking and cross-referencing documentation or informational dependencies. Notably includes the separation of Rubin Observatory Departments.
- **Storage View** — For describing all project documentation storage locations and repositories. In conjunction with the Product View, identifying or determining the normative source of information.
- **Access View** — For describing the user base and documentation applicable to their use cases.
- **Topic View** — For searching and discovering documentation.

The owner or a designated responsible group will use the Documentation Views to develop an effective way to communicate the various pieces of documentation, how they are interconnected, dependencies or connections to another group's documentation, and utilization for various documents. Stakeholders (e.g., internal and external technical groups, Rubin Observatory communities) should be able to retrieve current information with consistency and reliability. It is the responsibility of the owner or designated responsible group to keep this information up-to-date, organized, and readily available for stakeholders. Stakeholders should provide regularly feedback over the course of operations.

This framework will improve the ability of pre-operations and operations staff to clearly establish and review information for construction completeness and operational readiness. For example, all identified normative sources of information should be completed; or a use case where multiple documents referencing one normative source more effectively serves stakeholders instead of including all related information into a single document. The framework also supports managers and auditors in understanding a relevant system. For example, a document that is referenced isn't identified as a normative source of information; a normative source of information is not located in an appropriate repository; or, identifying interfaces between systems to ensure operational requirements are met.

The following subsections propose how each Documentation View tree is constructed, various examples of use, recommendations for technical teams to consider, and capture any assessments of needed resources.

2.1 Product View

The *Product View* is based on a common approach used in product management, a *product tree diagram*. The Documentation Working Group developed a framework customized for Rubin Observatory operations to facilitate the development of documentation and effective communication consistently project-wide. The two primary purposes are to convey ownership and responsibility of technical documentation and more readily describe complicated systems with context and categorization. The Product View allows one to identify specific documents as the normative source of information, then provide a manner which one can associate, link and cross reference documentation and informational dependencies from respective normative sources or references thereafter. These goals support identifying and resolving gaps, miscommunications or error-likely situations. The expected primary users are managers, product owners, engineers, specialists, technicians, and users who want to search downwards in a system's hierarchy.

The Product View will consist of product trees developed by the owning *Rubin Observatory Departments*, their technical teams and product owners. Each department's product tree(s) should be constructed to best organize and compartmentalize the set of *products* that make up the complicated systems under their purview. Products can span all manner of objects, such as department-specific categories, documents, hardware (systems, servers, networking, etc.), data (data sets, validation tests, verification artifacts, etc.), software interface information (alarms and notifications between hardware or software systems, support data storage and metadata schema, etc.), or subject-matter expert support. Each department or group is responsible to manage the product trees in addition to updating technical documentation to support access to retrieve associated information, interfaces and requirements with consistency and reliability. In light of this need, the Documentation Working Group recommends all Product View trees be available via one of the project repositories — it is suggested to create an appropriate lsst.io website so departments and groups can easily make changes.

Products are defined by the owning department in terms of a system or set of systems which can be grouped or decomposed into subsystems or individual components that is appropriate for them and their stakeholders. Referred to as the *generic categories*, the customized categories for the Product View developed by the Documentation Working Group are predefined root-nodes and the first set of leaf-nodes for a product tree. The generic categories were developed to capture critical operational aspects of a product while being extendable to subsystems of a product, where the subsystem may be a product with subsystems in its own right; i.e., a *Level-1 product* can be comprised of multiple Level-2 or lower-level subsystems, and those subsystems which are products are *Level-2+ products*. Note that these terms do not correspond to the construction phase definition of Work Breakdown Structure (WBS) or colloquial ”subsystem” used across the construction project. It is at the department's discretion on how to best organize and characterize these products and their product trees to manage their systems and flow of information; however, it is expected that all generic categories are associated with Level-1 and Level-2+ products either by construction or referenced information. While there will be exceptions for low-level systems, the Product View and its generic categories are designed to robustly capture construction completeness and operations readiness because they should generate sufficient discussion between groups and stakeholders to ensure key information is identified and provided.

2.1.1 Departments for Rubin Observatory operations

The Rubin Observatory departments shall be described by their associated scope, systems and/or products descriptions, and contacts to identify managers, product owners, or other key personnel (e.g., organizational chart, contact list). This section lists each department and respective scope. Contact information should be available to individuals that require it, and it should be clear when it's appropriate to contact an individual or group (e.g., as a product owner). Note that sometime *R* prepends these acronym, e.g., ROO for Rubin Observatory Operations.

Director's Office (DO) — The Vera C. Rubin Observatory Director's Office is responsible for the overall management of the observatory and the LSST survey, as well as fulfilling the mission of the observatory and realizing its vision. The Director's Office includes a Directorate, Administrative Operations, Safety, Communications, and In-Kind Contributions teams.

Observatory Operations (OO) — The Chilean-based Rubin Observatory Operations Department is responsible for operating and maintaining the telescope, camera systems, and summit facilities in order to collect the raw imaging and housekeeping data needed by the LSST. The primary tasks include maintaining the operating facilities, conducting the night-time survey operations, real-time assessment of image quality and observing efficiency, performing the daily calibration, and collecting and analyzing engineering data.

Data Management (DM) — The role of the Rubin Observatory Data Management department is to accept data from the Observatory's telescopes and ancillary systems; to process that data to generate science ready data products; to archive both raw data and derived data products; and, subject to approval from the Science Performance department and the Data Release Board, to make that data available to the scientific community. The Data Management department will develop, maintain and operate the networks, compute and storage hardware, and software that constitutes the Rubin Observatory Data Management System for the duration of the operational period.

System Performance (SP) — Rubin Observatory System Performance department is responsible for ensuring that the LSST as a whole is proceeding with the efficiency and fidelity needed to achieve its science requirements at the end of the 10-year survey. This includes the Wide-Fast-Deep (WFD) survey and all Special Programs (deep drilling fields and mini-surveys). To meet this goal, the System Performance department will track and optimize the integrated performance of the entire system. This includes the performance of the observatory and the progress of the survey with respect to its science objectives, the ability of the community to access and analyze the data and publish results on the four LSST science pillars at an appropriate rate, the evaluation of strategies for improving the survey strategy, and the development of mitigation strategies together with other relevant departments to minimize the impact of changes in the system performance on the overall LSST science.

Education and Public Outreach (EPO or EP) — The mission of the Rubin Observatory Education and Public Outreach program is to offer accessible and engaging online experiences that provide non-specialists access to, and context for, Rubin Observatory data so anyone can explore the Universe and be part of the discovery process. EPO serves as a website that highlights and contextualizes the scientific power of Rubin Observatory for non-specialists and hosts all online resources.

2.1.2 Generic Categories for the Product View

The generic categories provide a basis to define and associate critical systems and objective elements of a department and associated systems; they are not products in themselves. They are designed by the Documentation Working Group to relay information in a complete and concise manner by standardizing the distribution of information and products. Further opportunities arise with a well designed set of product trees: clearly establish relationships and dependencies between systems, serve to introduce the department/system in a digestible manner, and create a more adaptable structure to organize and target information between technical groups internal or external to the department.

The generic categories are separated by five high-level categories that act as root-nodes — Technical Design, Procedures, Safety and Emergency Response, Evaluation and Archival Documents. Each one includes subcategories that are the first- and second-level leaf-nodes. The generic categories are applicable to a variety of systems (e.g., hardware-centric, software-centric, hardware and software distributed, process and protocol dedicated) such that information associated to these categories and subcategories are available for practically all systems or subsystems described via a Product View tree. It may be difficult at first for technical groups to perform a logical and relevant decomposition of the systems such that the generic categories are applicable to all product trees, especially if a system can change in different scenarios, contexts, or states (e.g., the Simonyi Survey Telescope could have a product tree for maintenance and a different product tree for on-sky operations where each capture different components and interdepartmental interfaces separately). However, even in the case where the owner and stakeholders agree a category doesn't apply, the generic categories are a way to define and describe scope, respective responsibilities, inter- and intra-departmental interfaces, requirements, and key expectations.

Teams can and should create product trees that refer to documentation and ensure the product tree describes how these external documents address the generic categories. For example, this can take the form of a diagram coupled with a narrative, potentially with different representations between leaf-nodes. Within the Product View as a whole, consisting of many product trees, it is intended systems and subsystems refer to higher- or lower-level product trees to reduce replication and the risk of confusing users accessing information between the group(s) and stakeholder(s). Furthermore, the referential nature can be applied for interdepartmental information and dependencies so it's clear which department owns the information and any departments which utilize it. For example, an interface control document

(ICD) and an N-squared diagram could be sufficient references to describe the category *Inter-department Interfaces*.

Here are the generic categories:

- Technical Design
  - System Description
    * Description of System(s)
    * Definition of Sub-systems
  - Technical Design Specifics
    * System-level and Intra-department System Interfaces
    * Sub-system Level Information
    * Inter-department Interfaces
  - Access Interfaces (Physical and/or Software)
- Procedures
  - Operational Procedures
  - Maintenance Procedures
    * Preventive Maintenance
    * Reactive Maintenance
    * Turn-over Protocols (e.g., shift change, operational-to-maintenance status change)
    * Sub-system Isolation
  - Software Access and Use Documentation for Users
  - Software Development Documentation for Developers
  - Manuals and Data Sheets
- Safety and Emergency Response (System-level and relevant Sub-systems)
  - Emergency Procedures and Contacts
  - Hazards and Hazard Analysis
  - Mitigations and Verification Artifacts
    * Protocols
    * Energy Isolation
- Evaluation
  - Performance

  - Failure Effects and Failure Mode and Effects Analysis (FMEA)
  - Validation/Verification Test Plans
  - Test Reports
    * Technical Reports (Test and/or Analysis)
    * Verification Reports
- Archival Documents
  - Communications (e.g., Request for Information [RFI])
  - Construction Information, Unrelated to Commissioning/Operations

2.1.3 Examples of Product View trees

Here are two examples of product tree referencing, using Auxiliary Telescope (AuxTel), LSSTCam and Commissioning Camera (ComCam). Note that it would be beneficial to design product trees which take advantage of the similarities between LSSTCam and ComCam, even though there will be major differences, too.

For these three systems, the Access Interfaces category would include a common set of software, e.g., LSST Observing Visualization Environment (LOVE), Nublado (Jupyter) interface, Script Queue, Watcher. This common set of software could be included in a higher-level product tree (potentially a Level-2 product under the Observatory Operations department) or as a referential Level-2+ product trees; then, not only is the information referenced to all three efficiently, but the software product tree(s) can be designed to easily indicate differences between the three systems, say, for the Software Development categories.

The three systems interface with a set of external systems simultaneously connected to them, e.g., Engineering Facility Database (EFD), Environmental Awareness System (EAS), Global Interlock System (GIS). These external systems could also be set up as referential product trees which are referenced by AuxTel, LSSTCam and ComCam, as well as others.

Figure 1 depicts the Rubin Observatory departments and a subset of their systems and products into Level-1 products, with some Level-2+ products included.

Figure 2 is a more comprehensive example of AuxTel, depicting the Technical Design generic category. Each category (yellow) and each subcategory (blue) from the Technical Design is addressed such that the reader has an idea of what the system is, how it is used, and documen-

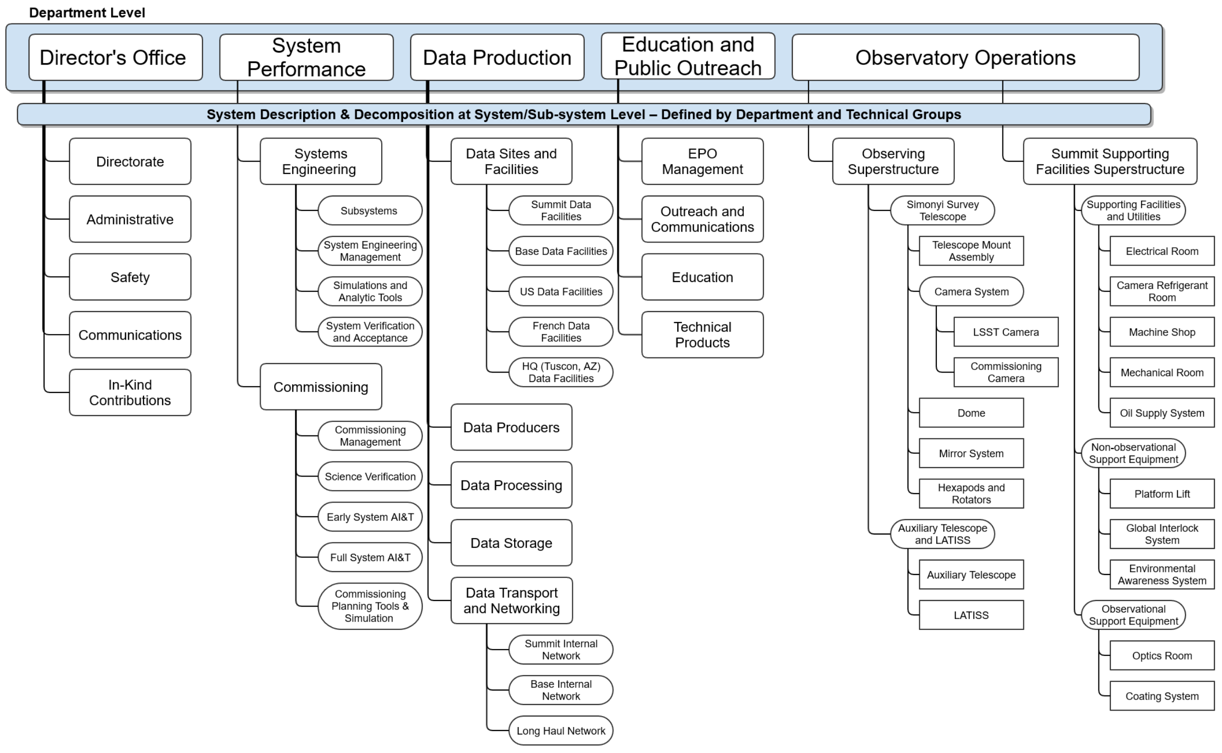


FIGURE 1: Example of Department Decomposition at System and Subsystem Level.

	Rubin Observatory Operations	**System Performance**	**Data Management**
Technical Design	ICDs/IDDs Requirements	Validation of Science Platform	Science Platform Design and Operations
Procedures	Day/Night Shift Turn-over, Energy Isolation Protocols, Calibration Procedures	Predictive Analysis, Hazard Validation	Data Release and Prompt Alerts Processing
Evaluation	Verification Test Plans, Failure Effects, Test Reports	Failure Mode and Effect Analysis	Assertion Testing

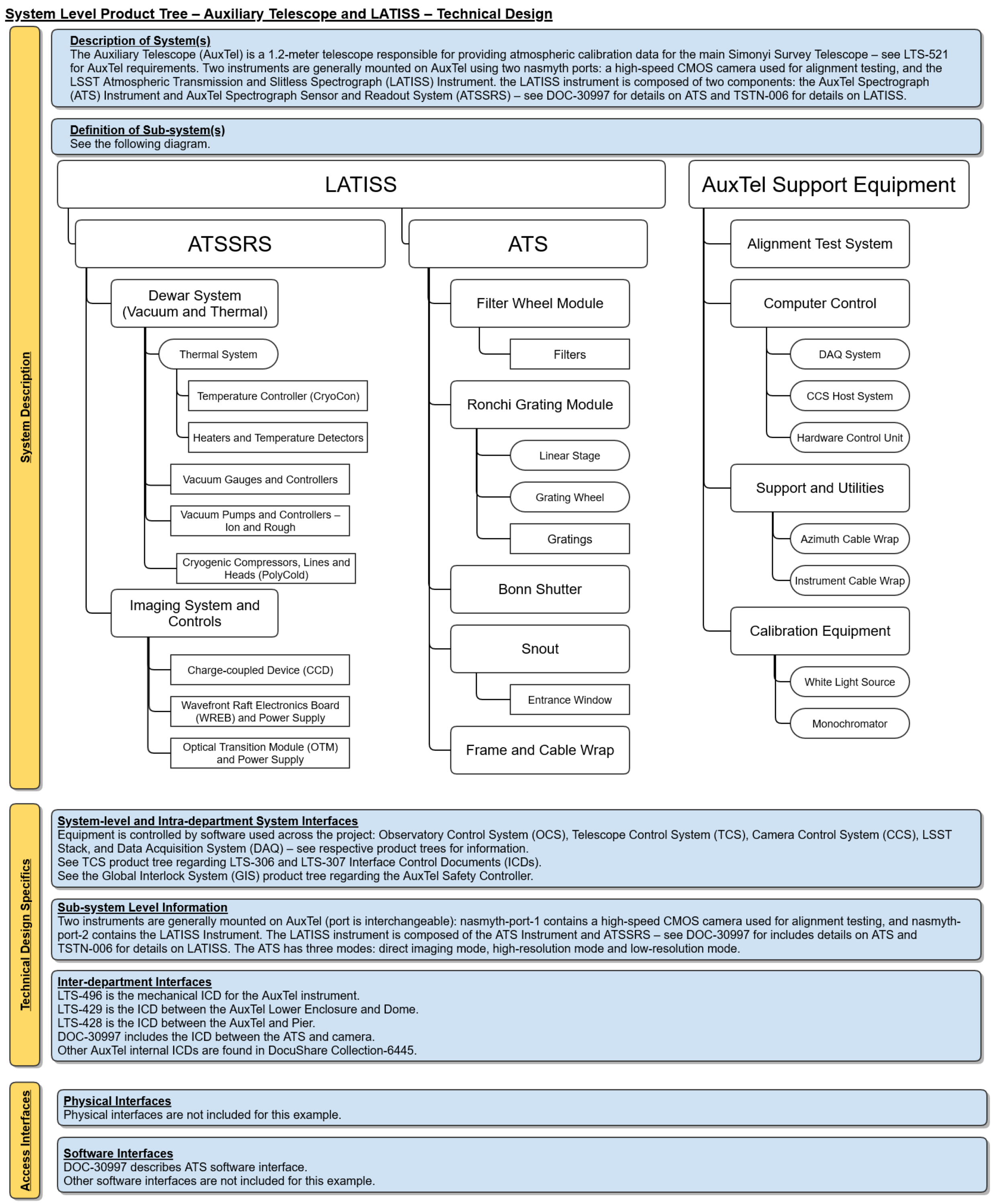
tation with references therein for retrieving critical information. Note that physical interfaces are not address in the example.

While it may seem Figure 1 and Figure 2 are part of one example, that is not the case. As previously described, it would be beneficial to organize the Product View to leverage the replicated information such as common software. This was not considered for the two figures.

As another example of the Product View, Table 2.1.3 shows how multiple departments can separate their responsibilities between a few high-level requirements and operational needs. With well-defined responsibilities, teams can communicate more effectively and the normative source of information should be identifiable.

2.2 Storage View

The purpose of the *Storage View* is to capture the official repositories retaining, archiving, organizing, and accessing Rubin Observatory technical documentation in a reliable, consistent manner for users, developers and management. The Storage View provides a method for departments and technical groups to define locations for respective normative sources of information. With canonical information storage, others can develop associations, links and references to impart information to inter-departmental or external documents or groups, allowing a reduction in replication, reliable information flow, and preservation of informational dependencies. Unless mandated by the project, the department or responsible group, in conjunction with the stakeholder(s), can choose the manner and method of storing information and transposing information to other locations. The primary users are managers, engineers, specialists, technicians, web development staff, system administrators, and documentation specialist.



System Level Product Tree – Auxiliary Telescope and LATISS – Technical Design

Description of System(s)
The Auxiliary Telescope (AuxTel) is a 1.2-meter telescope responsible for providing atmospheric calibration data for the main Simonyi Survey Telescope – see LTS-521 for AuxTel requirements. Two instruments are generally mounted on AuxTel using two nasmyth ports: a high-speed CMOS camera used for alignment testing, and the LSST Atmospheric Transmission and Slitless Spectrograph (LATISS) Instrument. the LATISS instrument is composed of two components: the AuxTel Spectrograph (ATS) Instrument and AuxTel Spectrograph Sensor and Readout System (ATSSRS) – see DOC-30997 for details on ATS and TSTN-006 for details on LATISS.

Definition of Sub-system(s)
See the following diagram.

System-level and Intra-department System Interfaces
Equipment is controlled by software used across the project: Observatory Control System (OCS), Telescope Control System (TCS), Camera Control System (CCS), LSST Stack, and Data Acquisition System (DAQ) – see respective product trees for information.
See TCS product tree regarding LTS-306 and LTS-307 Interface Control Documents (ICDs).
See the Global Interlock System (GIS) product tree regarding the AuxTel Safety Controller.

Sub-system Level Information
Two instruments are generally mounted on AuxTel (port is interchangeable): nasmyth-port-1 contains a high-speed CMOS camera used for alignment testing, and nasmyth-port-2 contains the LATISS Instrument. The LATISS instrument is composed of the ATS Instrument and ATSSRS – see DOC-30997 for includes details on ATS and TSTN-006 for details on LATISS. The ATS has three modes: direct imaging mode, high-resolution mode and low-resolution mode.

Inter-department Interfaces
LTS-496 is the mechanical ICD for the AuxTel instrument.
LTS-429 is the ICD between the AuxTel Lower Enclosure and Dome.
LTS-428 is the ICD between the AuxTel and Pier.
DOC-30997 includes the ICD between the ATS and camera.
Other AuxTel internal ICDs are found in DocuShare Collection-6445.

Physical Interfaces
Physical interfaces are not included for this example.

Software Interfaces
DOC-30997 describes ATS software interface.
Other software interfaces are not included for this example.

FIGURE 2: Example of Technical Design Generic Category from Product View, for an Auxiliary Telescope Subsystem.

To ensure a high degree of reliability, it is crucial staff store all operational and historical information within officially designated locations. Specifying official repositories will limit the level of effort in the utilization and maintenance of multiple documentation systems, prevent or highly discourage using non-official storage repositories, and reduce risk to storing and accessing operational and historical information. By limiting the available repositories, effort and resources can focus on best using the official storage locations or modify them in an orderly, consistent manner.

As a shared responsibility between system administrators and departments, all official repositories should be well maintained. System administrators must ensure access throughout operations (e.g., readily available and backed-up) and a level of expertise must be on-hand for maintenance, issues and general assistance to staff. Departments and staff must maintain their information and its organization within the repositories. They should recognize and act when information or utilization becomes outdated or unused as to prevent large amounts of depreciated content, stress on storage and bit rot. This is especially important to the development, deployment and maintenance of the Rubin Documentation Portal architecture (Section 3), as leveraging specific use cases and repository's native or project-developed metadata fields are key components. Additionally, this effort will better suit the project and departments by formalizing a long-term organizational structure of each repository, create customized workflows, transition to the Documentation Views, and implement new or updated documentation.

Non-official storage locations pose unique operational and managerial risks which can lead to information and data loss, increased information security risks, and impact to the project's schedule and budget. This includes information control (e.g., access, enforced version control, archiving), reliable data recovery, unknown or unplanned risks, and lack of available support (e.g., available labor, expertise). Platforms may become unsupported or unavailable and the information therein becomes lost or otherwise inaccessible. This risk has already been realized within the construction project, although of minor impact, as evident from a small number databases (e.g., personal drives, old network drives) that are difficult to access or have lost or inaccessible information. [SITCOMTN-012] Additionally compounding the risk to information access, it becomes increasingly difficult to share information, especially when the repository is ill-defined, uncertainty of what replicated information is the most current, and uncertainty of which repository is the normative source of information. Without a discrete set of storage locations, it is impossible to guarantee reliable and navigable information or data across multiple platforms throughout operations.

A Storage View tree is constructed a root-node for a repository and the first set of leaf-nodes capturing the repository's organizational structure and metadata structure. Critically, each Storage View tree should readily indicate what are the normative sources of information so it's accurately referable within other Documentation Views. It will be the responsibility of the department or technical group to define the organizational structure. Software developers require the metadata and repository's organizational structure to design and implement the Rubin Documentation Portal, including the development of additional metadata fields with assistance from system administrators.

This section summarizes some important details from SITCOMTN-012; however, that document includes detailed information, such as the inventory for each repository. Recommendations from the Documentation Working Group for the repositories are provided, including transitional plans, future in operations, and some suggested repository's organizational structure.

2.2.1 DocuShare

Xerox® DocuShare® content management system (Xerox, 2021) is the Construction Project's official document repository. It was selected during the design and development phase to meet the NSF requirement for a document management system. The Construction Project Office expects DocuShare to be the repository for official versions of management policies, plans and procedures, design documents, safety documentation, hazard analyses, released requirements and interface control documents generated from the SysML model, and project standards, guidelines and templates. This list is not intended to be exhaustive.

As a tool heavily used by the project, it is recommended to continue the use of DocuShare into operations. Operations should use the DocuShare instance currently in use by the construction project with the Archive Server add-on enabled, as the continuity afforded by doing so is important and useful. Under this model, the construction project content would be moved to the archive server, and the active server would contain only operations and operations-relevant construction content following an agreed-upon directory structure. Until this transition is completed project-wide, the current location for operational documentation is Collection-602.

The suggested DocuShare branch of the Storage View for operations starts with the a collection for each department. Additional collections can be created based on the project's needs as a whole (e.g., interdepartmental items such as widely used software tools). The next level of nodes is defined by the system's and/or product's department or owner. Further lower-level nodes would depend on how the department and owners want to parse the information, e.g., separation by subsystem, separation by topics such as reports, or a combination of the two. As for construction documentation, it is recommended to have a consistent organizational structure when transferring over to the archive server. The archiving process should include a determination by the department if the information is useful in operations (determining if it should be archived) and if the information should be queryable in the archive (e.g., raw data). This is especially topical for vendor documentation and deliverables.

A more detailed analysis of arguments and considerations for this proposal's recommendation is available in DocuShare Options Trade Study for the Documentation Working Group. [Document-36788]

2.2.2 LSST the Docs (www.lsst.io)

LSST the Docs (LTD), also known by its URL ”lsst.io”, is a documentation hosting platform built and operated by the SQuaRE team within the Data Management group. LTD hosts *versioned static websites*, meaning any website built from HTML, CSS and JavaScript that doesn't need an active server to render content (as opposed to say Confluence, DocuShare, or Drupal websites). Static websites are a natural fit for documentation projects that originate from repositories hosted by GitHub (GitHub, Inc.), so LTD is uniquely developed to be built around versioned documentation in GitHub. *lsst.io* is one such deployment used as a hosting domain for Rubin Observatory, where all subdomains of lsst.io are independent documentation projects. The technical motivation and design of LTD are documented in SQR-006: The LSST the Docs Platform for Continuous Documentation Delivery. SQR-006 The key technical features of LTD are:

- high reliability, scaling, and security: documentation is hosted in Amazon S3 and served through the Fastly content distribution network. We don't operate any servers that receive traffic from users;
- versioned documentation; and
- flexibility to host any type of static website.

Using LTD documents provides a simple use case with additional features developed by the SQuaRE team. The root URL for a documentation project hosts the ”default” version, which has a configurable meaning for each project (such as software release versions, temporary collaborative drafts, or an active version in DocuShare). Users can also browse other versions of the documentation through the ”/v/” dashboard pages (for example `https://www.lsst.io/v/`).

LTD hosts two types of documentation projects: *guides* and *documents*. Guides are multi-page websites convenient for user interaction and navigation (e.g., Data Management Developer Guide, T&S software guides at `https://obs-controls.lsst.io`). Documents are ”single-page” artifacts, analogous to documents that might be found in DocuShare, and they are sometimes referred to as *technical notes* (shortened to ”technote” or an appended ”TN”) — see SQR-000: The LSST DM Technical Note Publishing Platform for the motivation to create technical notes. [SQR-000]

A unique example is the homepage for the LTD documentation platform, `https://www.lsst.io`, which serves as a portal for LTD indexed documentation for searches and faceted browsing capabilities. (SQuaRE Team, 2021a) Users can search across metadata and full text (this feature is powered by the commercial service Algolia (Algolia, 2021a) in conjunction with a scraper bot built by SQuaRE) or browse through curated collections. The site itself is built with React/Gatsby.js (`https://github.com/lsst-sqre/www_lsst_io`), the search database is SaaS (Algolia, 2021b)), and the bot that indexes content into the search database is called Ook (`https://github.com/lsst-sqre/ook`). It is still in development and the current status is documented at `https://www.lsst.io/about/`.

As a customized set of tools that is heavily used by the project, it is recommended to continue the use of LTD and its software system into operations. The project should use documents and technotes to capture information that is static, rarely changed or serving a temporary need (e.g., proposals, documenting proof of principles, status updates); whereas, guides should be used for information that is actively updated with current information (e.g., troubleshooting guides, procedures not under change control). Neither LTD documentation type should be used for documents under change control albeit one can link to the change-controlled document within a document or guide. Further, as the LTD software system, it’s additional utilities and `https://www.lsst.io` are designed and built by the SQuaRE team, there is considerable opportunity to leverage the expertise and experience already developed to implement a documentation portal for other platforms (see Section 3).

The LTD branch high-level nodes of the Storage View could be by document type (i.e., guides or documents). Following the department/owner or specific technote series (e.g., RTN, SITCOMTN) could be the next level of nodes.

2.2.3 Confluence and Jira

Confluence (Atlassian Corporation, 2021) and Jira (Atlassian, 2021) are part of the Atlassian Corporation suite of tools used by the construction and operations projects for many purposes. They are collaborative tools where teams/groups can document, share and develop information. Confluence is organized into *spaces*, each with a varying number of pages and sub-pages. Jira is organized into *projects*, each of which tracks a list of enumerated tickets or issues. Both include a large variety of features, tools and extendable add-ons to manipulate or organize the information. Many of these spaces and projects are very specialized, with some set up for personal use.

On the construction project, there are two instances of each software tool, requiring different credentials — `https://confluence.lsstcorp.org` and `https://jira.lsstcorp.org` is for the NSF MREFC effort hosted by LSSTC out of Tucson, Arizona; and, `https://confluence.slac.stanford.edu/` and `https://jira.slac.stanford.edu/` is for the DOE MIE effort for LSSTCam hosted by SLAC National Accelerator Laboratory out of Stanford, California. Rubin Observatory operations and pre-operations staff currently use the LSSTC tools.

Currently, the LSSTC Jira instance includes the following normative source of information: (1) verification elements, plans, cycles, cases, results, etc., (2) construction-related risks, opportunities and mitigations, (3) the Failure Reporting Analysis and Corrective Action System (FRACAS) for failures and corrective actions, and (4) hazard mitigation verification.

As tools that are heavily used by the project, it is recommended to continue the use of Confluence and Jira into operations. With a wiki like Confluence which that integrates well with Jira, the two are especially convenient for rapidly developing new ideas, taking and storing meeting minutes, collecting information in interactive tables, and drafting outlines for future documentation, all while tracking the tasks and being able to actively report on status. Other powerful features include simultaneous editing, native sharing options internal to Confluence/Jira or external such as email, etc. As operational-based information takes form, departments and technical groups should reconsider the spaces and projects to prevent having a large number of unused or old areas, as seen with the current Confluence and Jira instances. In the case of

Confluence, project staff must be fastidious in moving content needed on a long-term basis into other official storage locations at the earliest appropriate stage (e.g., DocuShare, lsst.io). Users must understand the limitations of information and provided guidance as to how and when information should be moved from Confluence or Jira to another storage location.

The Confluence branch of the Storage View would begin with the spaces. Most spaces have a relatively small number of top-level pages which are tailored to a specific need (e.g., separation of subsystems), and these top-level pages can naturally be the next level of nodes. The Jira branch would begin with the projects, but the next level nodes are not as apparent and should be determined by the owner or responsible group. There are many ways to create the next level nodes for Jira; for example, most projects have a natural breakdown of structure used to organize relationships between tickets (e.g., epics). Lower-level nodes will depend on use; they could include items such as meeting notes or customized dashboards.

2.2.4 Engineering models in Solidworks Product Data Management (PDM)

Solidworks Product Data Management (PDM) Professional (Solidworks, 2021) is the official computer-aided design (CAD) model repository for the Telescope and Site construction group, including vendor documentation and deliverables. It uses a check-out / check-in system to allow configuration management of the design where earlier versions can be accessed if needed to compare designs or revert to an earlier design. A workflow feature allows the designs to go through a review process until the design is approved and locked from further changes. A revision process is also included in the workflow to allow for changes to the designs if needed after final approval. The current configuration of the PDM vault contains top-level access to baseline design data and interface control documents (ICDs) (drawings) along with as-designed vendor subsystems. Solidworks PDM is the normative source of information for system decomposition.

It is recommended to continue use of PDM vault into operations. In operations, as-built design information will be needed to help with logistics planning for maintenance and future design upgrades. No information has been deleted or archived at this time. If continued into operations, it is recommended the vault is reorganized so that legacy data is archived for access but is not easily mistaken for as-built design data. The specific structure for the branch of the Storage View is highly dependent on this reorganization. For example, the high-level nodes could be by subsystem or physical location, lifecycle of particular drawings (e.g., as-designed, as-built), or use within specific states of the telescope (e.g., on-sky imaging, preventive maintenance shutdown).

2.2.5 MagicDraw

MagicDraw (No Magic, Inc.) is a tool used to maintain a model of the Rubin Observatory system by creating a relational database between system elements. There are a number of elements captured in the MagicDraw database defined as the normative source of information, with the information synced and/or exported to other repositories. As the normative source of information, it will be up-to-date when given over from construction and will continue active use within operations. Details on how Rubin Observatory uses MagicDraw are included in the user guides collected on the following Confluence page: `https://confluence.lsstcorp.org/display/SYSENG/MagicDraw+LSST+Users+Guide`.

- **Hazard Analysis** — normative source — synced to Jira for verification tasks.
- **Failure Mode and Effects Analysis (FMEA)** — normative source — known to be incomplete.
- **Requirements** — normative source — exports to DocuShare.
- **Verification Elements** — normative source — synced to/from Jira.
- **Verification Plans/Cycles/Cases** — synced to/from Jira via Syndeia™ (Intercax, LLC). See Section 2.2.3 for normative source).
- **SAL Commands, Events, Telemetry** — imported from CSC XML.
- **Operations Concepts** — source of truth.
- **System-level State Machine** — source of truth.
- **Interlocks** — modeled from source material.
- **Structural Decomposition** — will be synced from Solidworks in the future.

It is recommended to continue the use of MagicDraw into operations. The user guides should be relocated into a more appropriate storage location, such as lsst.io. The content in MagicDraw should be reviewed to create the tree for the Storage View that can readily reflect normative sources of information and how/what references or depends on this information. It is crucial that the normative source of information is clearly defined between MagicDraw and other repositories which are synced or used for archival purposes, as well as the workflow and method to do so. The effort required to complete the FMEA information will depend on the level of completeness when handed over to operations. Effort to sync the structural decomposition with Solidworks can be evaluated when the system information is more complete.

2.2.6 Euporie

Euporie is a network drive on a server managed by the construction project; and, it is accessible with Rubin Observatory credentials through VPN at `smb://euporie`. The drive contains a number of personal directories and a directory named ”TS-Deliverables” managed by the Telescope and Site group. Stored in subdirectories of TS-Deliverables are vendor-supplied documentation as contract deliverables (design documentation, construction drawings, manuals and other miscellaneous information), with each subdirectory.

It is recommended to discontinue use of Euporie for operations. A review of the content of personal and shared directories is recommended to determine what information should be relocated on a case-by-case basis. Additionally, a determination should be made if information should be archived as construction-related or moved to a more active storage location for operations staff access. After this is complete, and if access will continue for purpose unrelated to operations, users must understand the limitations of information (namely for long-term operational activities) and provided guidance as to how and when information should be moved from Euporie to another storage location.

2.2.7 GitHub

GitHub™ (GitHub, Inc.) is used by the project for software and documentation collaboration, storage, version control via git™ (Software Freedom Conservancy, Inc.) and deployment. GitHub is primarily utilized via *repositories*, or repos. Repositories are owned by individual users or an *organization*, and organizations can include *teams* for additional granularity.

The Rubin Observatory construction and operations projects use a large number of GitHub repositories, organizations and teams, primarily to ease access control. The main project GitHub organization is `https://github.com/lsst`; however, not all operations software is found in this organization. GitHub organizations with operational software and documentation will likely transition smoothly to Operations teams. Particular attention will need to be paid to maintaining organizations that no longer have an active team associated with them.

As a heavily used tool by the project, it is recommended to continue the use of GitHub via git into operations. Effort by construction, pre-operations and operations teams already implemented infrastructure for utilization and the GitHub collaboration structure, including basic structure for operations. There are a large number of repositories containing the normative source of information, and the project should ensure these repositories are clearly indicated as such and what respective information is the normative source.

2.2.8 Drupal

Drupal (Drupal Association) is an open source web content management framework used for project websites' back-end, including `https://www.lsst.org`, `https://project.lsst.org` and sites created to facilitate project meetings and reviews. To avoid the complication of having to make multiple site updates when content changes, the documents on each of these sites are served by hyperlinks that pull files from whatever repositories contain their normative sources of information. While it is possible to upload discrete files to a specific site's server location, by policy and standard the project eschews doing so.

The major exception is project-level meetings and reviews sites, such as the Project and Community Workshop (PCW) or agency reviews. Their site directories contain document and presentation files in order to preserve the content presented at the time. At the conclusion of the event, these files are uploaded to DocuShare in a collection specific to the event. Documents such as policies, requirements, and design documents are uploaded as a ZIP file to preserve a snap-shot while preventing replication and multiple handles that may cause confusion with the normative source of information.

It is recommended for the project to evaluate the need for web content management in operations. If needed, the project should consider if other tools planned for operations would fulfill the same need, or if Drupal should be continued into operations. The level of change control appropriate for websites must be determined, as well as files in site directories or hyperlinked files without change control. Since the method described for project-level meetings and reviews may not be implemented by all subsystems on the construction project, it is recommended the project review the content to ensure historical information is preserved appropriately. If Drupal is discontinued, it is recommended the project review if all required archival information is transferred to the new web content management system and/or an actively supported repository (e.g., DocuShare). If web content management is controlled to the same degree as construction, it is recommended that the project provide a clear workflow of how and where normative sources of information is used and stored.

2.2.9 LSSTCam information

During construction, the Camera used a set of web-based tools used across multiple sites during the testing and construction of the Camera and its subcomponents — eTraveler, Data Portal and Data Catalog. Databases and servers managed by SLAC used these tools to capture procedure, testing data, track component/assembly status and information, and to track the acceptance and non-conformance reports. Both the ”prod” and ”dev” instances include information on production parts. Additionally, there is construction information stored in a SLAC-hosted instance of Confluence and Jira.

It is recommended the project review the information to determine the type of information and method by which it is archived to an actively supported repository (e.g., DocuShare). Data within repositories that are not be actively supported or available may be lost. For repositories that may not be available throughout operations, the project should consider which information should be transferred for future use in operations or archived for future retrieval.

2.2.10 Primavera P6

Primavera P6® is a project portfolio management tool by Oracle® Corporation. (Oracle Corporation) It was used for planning, managing, and executing the Construction project. It is recommended that the P6 record of the Construction project be exported and saved at the end of the Project; its use will not continue into Operations.

2.2.11 Verification reports

The Rubin Observatory Verification Architecture used for construction and commissioning activities is composed of our MagicDraw model, Jira with Test Manager application, and Syndeia to synchronize data between MagicDraw and Jira. For all verification events performed by the SIT-COM team, a SIT-COM Test Plan/Report (SCTR) is created setting the standard for verification execution and results. The SCTR/DMTR is generated using the Jira REST API pulling the information from the verification objects. Formal detailed records of each verification done by Rubin Observatory is in the LSST Verification and Validation Jira project, with reports published to lsst.io as a SCTR or DMTR in addition to archiving it into DocuShare.

Similar to MagicDraw, it is crucial that the normative source of information is clearly defined between the verification reports and other repositories which are synced or used for archival purposes, as well as the workflow and method to do so.

2.2.12 Computerized Maintenance Management System, CMMS

In operations, the project will use openMAINT® as a customizable Computerized Maintenance Management System (CMMS), with the assistance from the managing company, Tecnoteca srl©. (Tecnoteca srl) The CMMS is expected to execute and coordinate maintenance activities, maintain the history of these activities, and provide reporting tools. It is expected that not all procedures will be stored in CMMS, e.g., a hyperlink to a procedure in DocuShare may be used instead of writing the procedure into CMMS. As the work release workflow will be in Jira, the CMMS will have an interface (e.g., REST API) and update Jira as needed. With a well-defined workflow, the CMMS and Jira interactions can be streamlined to minimize confusion of scope and use.

It is recommended the project review the expected capabilities of CMMS by the start of operations so it can be determined what type of information will be in CMMS as the normative source of information. All other information should be stored in another actively supported repository. It is crucial that the normative source of information is clearly defined between the CMMS and other repositories which are synced or used for archival purposes, as well as the workflow and method to do so.

2.2.13 NOIRLab Alcea Risk Tool

The Rubin Observatory is required to include risks, opportunities and mitigations in the NOIRLab managed risk management software tool, Alcea Tracking Solutions software `https://noirlab.alceatech.com/`. The Rubin Observatory information is exported and archived in GitHub by operations staff, and a user guide for the Rubin Observatory is available at . Access will be limited to those using the tool directly and designees. Given the limited access, the project expects to use Jira for detailed planning and analysis.

It is recommended the project should confirm that the normative source of information for risk, opportunities and mitigations will be Jira. Assuming Jira will be used, the type or criteria of risks, opportunities and mitigations required to be included in the Alcea Risk Tool must be defined and agreed upon between the project and NOIRLab. Additionally, it is recommended that a workflow, method and/or software is used to ensure information is consistently synced at an agreed upon frequency.

2.3 Access View

The *Access View* is a method of categorizing information into trees serving role-based needs to stakeholders. Its purpose is to assist users in retrieving and discovering documentation and information applicable to their use cases. As such, the access trees will be tailored to use case and role-based needs. The Access View is to be agnostic to the storage location because access is provided through the Documentation Portal so users do not need to know the repository for information.

The following subsections include four recommended trees for the Access View developed by the Documentation Working Group: Operational Access View, Safety Access View, Scientific Access View, and Engineering Access View. It was natural to develop an Access View tree from a system decomposition, but it is useful for the project to determine if there are more efficient or natural trees, such as activity-based trees. For each Access View tree, there are many ways and reasons to break the tree into the root-nodes. With such sprawling associated subjects, it's more crucial to develop a strategy to break these root-nodes and leaf-nodes by considering categories, impacts to users, and how users would want to find associated information.

2.3.1 Operational Access View

The Operational Access View tree is intended to help with interfaces of real-time data and visualization, observations and its programming, and system state. Figure 3 is an example of the Operational Access View. It may be natural to have separate leaf-nodes for the Main Telescope and AuxTel, capturing specific differences such as control interfaces. Another natural break down could be day- and night-time operations, as shown in Figure 4. Responsible groups should take advantage of commonalities and referential nature of the Documentation Views. Users include use of equipment control devices, procedures and protocols for operation and maintenance, such as control room staff, viewing assistants.

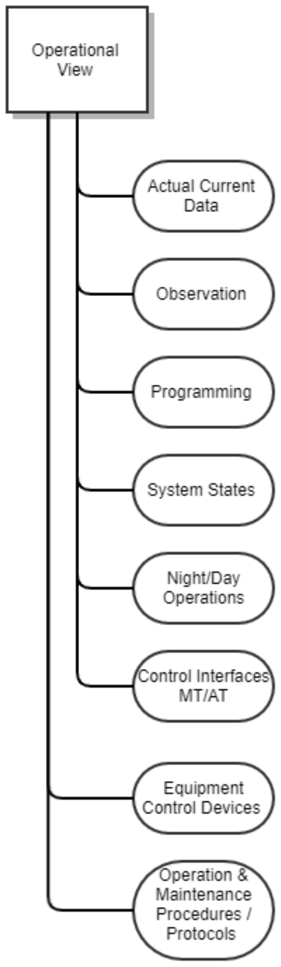


FIGURE 3: Example of Operational Access View Tree with Root-nodes.

2.3.2 Safety Access View

The Safety Access View tree focuses on accessing information related directly to safety procedures and protocols that apply to the different systems. This can help add consistency and verify requirements. This tree would lead users to general safety procedures, specific safety procedures, emergency procedures of telescope and summit facilities (excludes emergency response). The example provided by Figure 5 includes safety and security aspects. Users include safety, engineering, security, maintenance staffs and management.

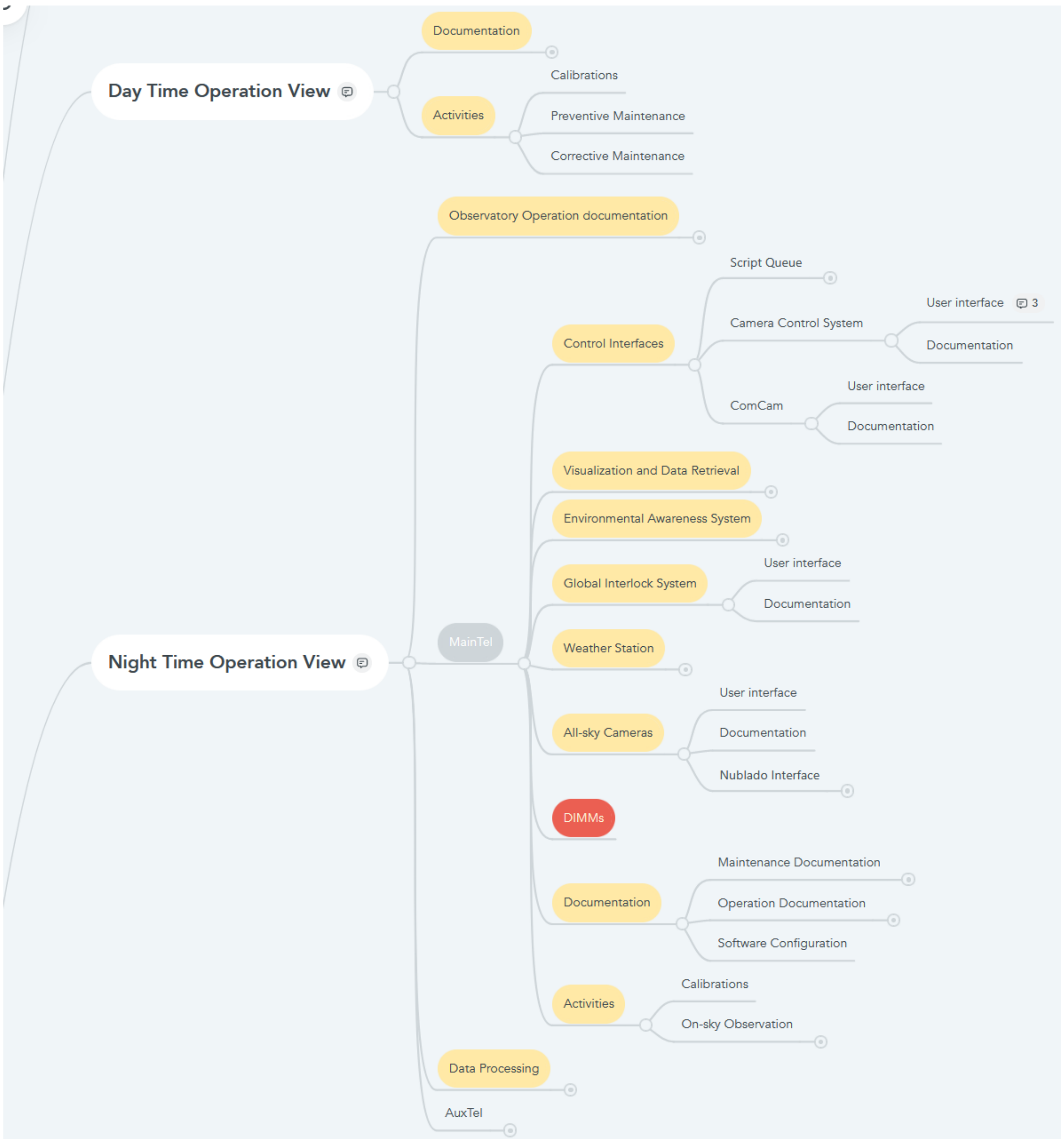


FIGURE 4: Example of Operational Access View Tree from Daytime and Nighttime Operations.

2.3.3 Scientific Access View

The Scientific Access View tree is for scientific platforms and interfaces that correspond or impact Rubin Observatory science objectives. It is intended to ease access to the information

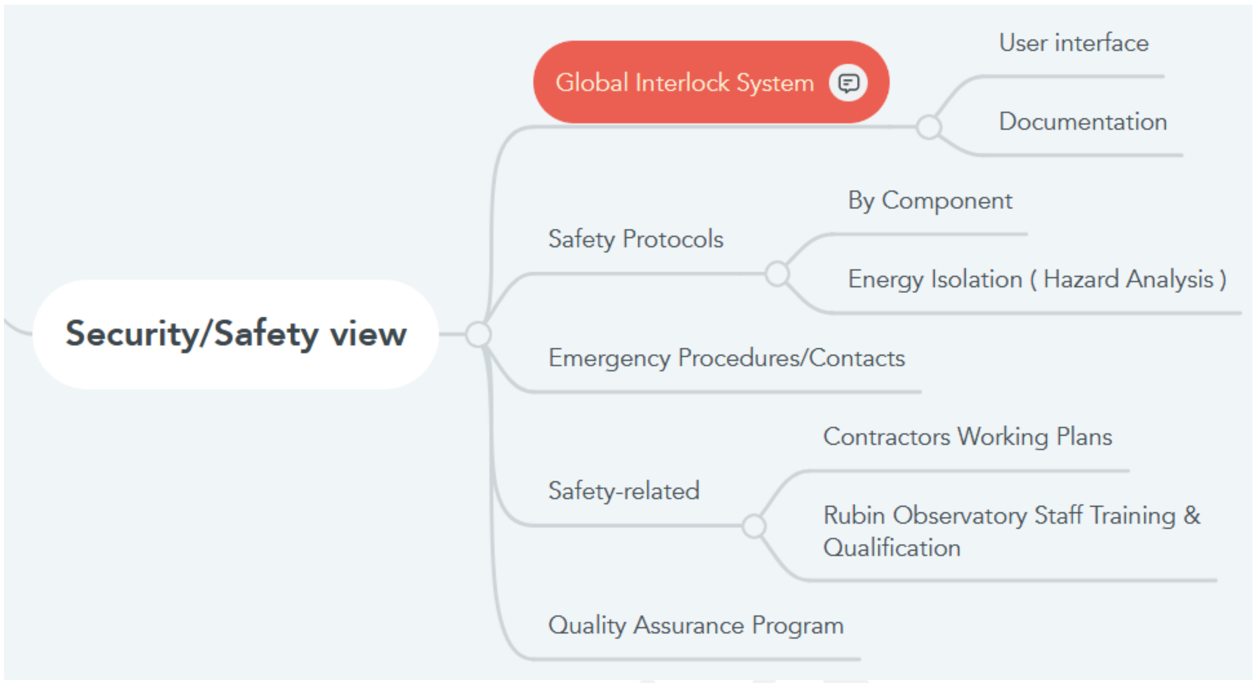

Security/Safety view
Global Interlock System
User interface
Documentation
Safety Protocols
By Component
Energy Isolation (Hazard Analysis)
Emergency Procedures/Contacts
Safety-related
Contractors Working Plans
Rubin Observatory Staff Training & Qualification
Quality Assurance Program

FIGURE 5: Example of Safety Access View Tree.

by providing a clear list of the software and various databases. An example is provided by Figure 6. Users include scientific, engineering, and observing staff and management.

2.3.4 Engineering Access View

The Engineering Access View tree is for accessing Rubin Observatory technical information. In addition to the technical documentation, access interfaces and facility-related data would be needed. An example is provided by Figure 7. Users include staff and management involved in system performance and system engineering activities.

2.4 Topic View

The *Topic View* is an approach to categorize information into document types. Its purpose is to allow users the ability to find information of a specific form within a documentation system. It first requires a defined decomposition of systems, subsystems and components. Topics could range to any categorization, such as physical location or department ownership. The Topic View must be agnostic to the storage location because it requires an access portal to utilize it, such as the Rubin Documentation Portal (Section 3). The construction of a Topic View tree

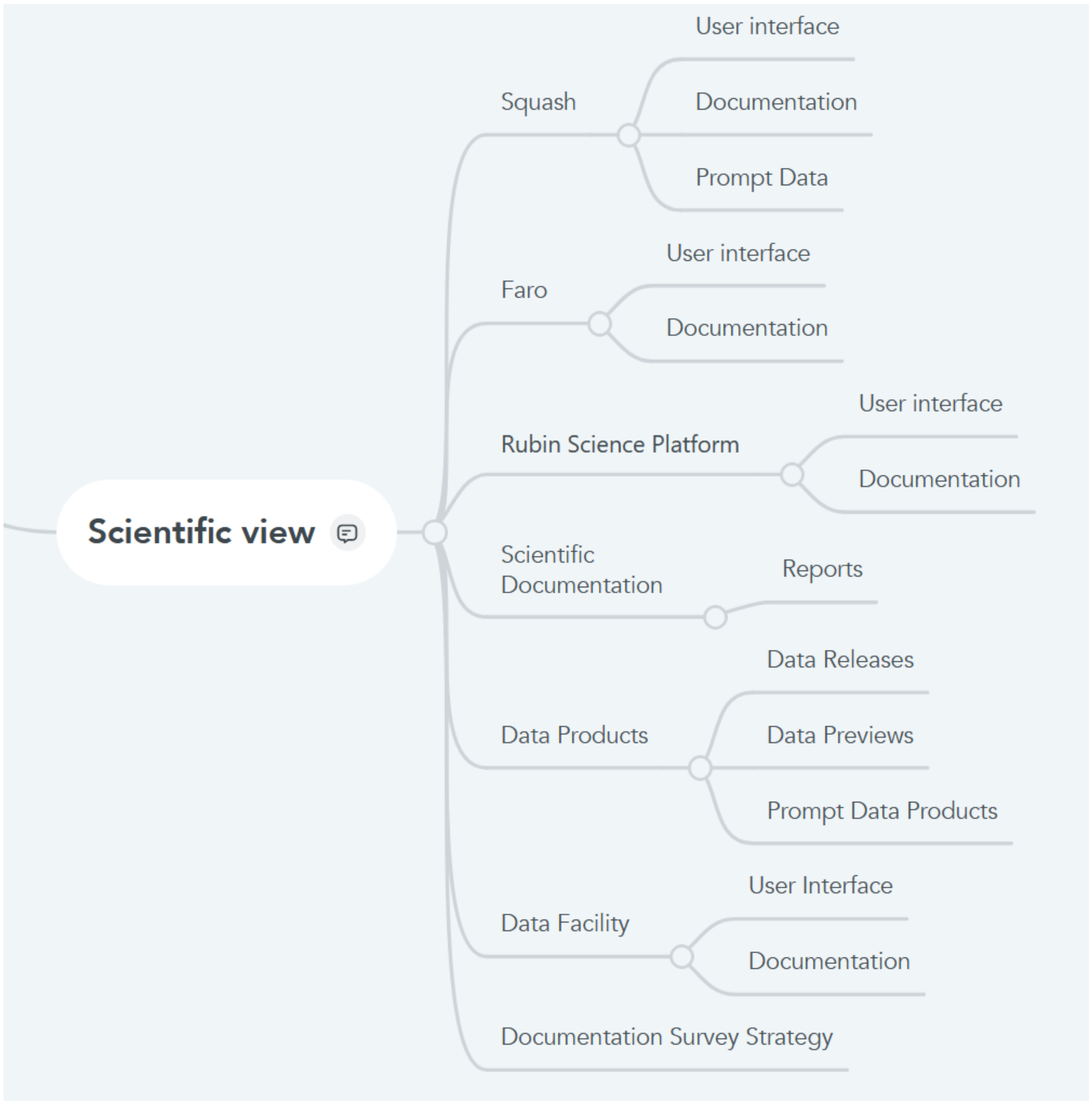


FIGURE 6: Example of Scientific Access View Tree.

depends on the implementation of the portal, the purpose of the topic tree, and the targeted user base.

The Documentation Working Group suggests the following categories to define a system for the Topic View:

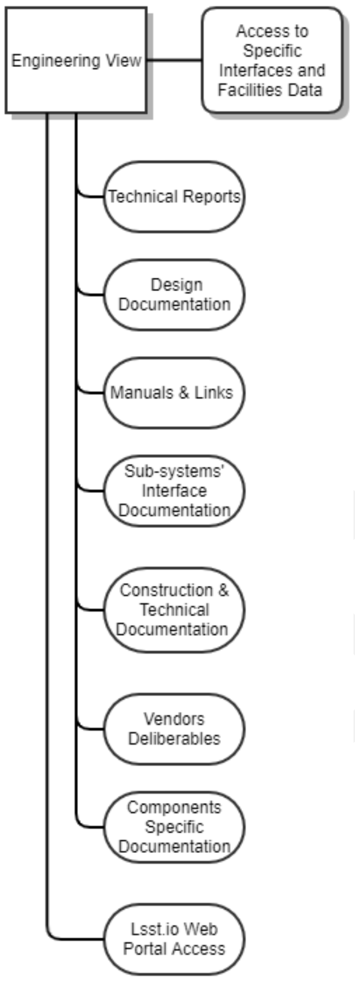

Engineering View
Access to Specific Interfaces and Facilities Data
Technical Reports
Design Documentation
Manuals & Links
Sub-systems' Interface Documentation
Construction & Technical Documentation
Vendors Deliberables
Components Specific Documentation
Lsst.io Web Portal Access

FIGURE 7: Example of Engineering Access View Tree.

- design documents
- requirement documents
- technical manuals (e.g., operation or maintenance)
- operational documents (e.g., procedures)
- data performance
- evaluation processes
- maintenance reports
- safety and hazard mitigations

- access and software

3 The Rubin Documentation Portal

The Documentation Working Group recommends the creation of a *Rubin Documentation Portal* web application as a means of making documentation content discoverable and accessible to Rubin Observatory staff. This portal will provide interfaces for both searching (based on content and metadata) and browsing (based on hierarchical categorization) of documentation resources, and the Documentation Views (Section 2) would provide different methods of doing so. Documentation does not reside within the portal itself. Rather, the portal's objective is to efficiently link the user to the document, where it is stored in any of the observatory's adopted storage platforms (Section 2.2). The new Rubin Documentation Portal discussed herein is based upon the website ()lsst.io-cite which provides a search and browsing interface for the Rubin Observatory's public-facing technical documentation operated by the SQuaRE (Data Management) team. (A detailed description of , LSST the Docs (LTD) and applicable software is available in [SITCOMTN-012].) The new portal will be accessible only those with Rubin Observatory staff credentials, and will be purpose-built for observatory and survey operations. This section describes the design principles, technical architecture, security model, and cost estimate of the Rubin Observatory Documentation Portal.

3.1 Design drivers

The following are high-level design drivers for the future-state Rubin Documentation Portal. They reflect the recommendations made by the Documentation Working Group presented in this report.

The role of the Rubin Documentation Portal is to link to documentation resources. The portal itself does not host the content itself, nor provide user interfaces for creating and maintaining new versions of documentation content. This requirement reflects the recommendation from the Documentation Working Group that documentation content should be hosted on a select set of platforms that are idiomatic for the content and the teams that work with that content.

The Rubin Documentation Portal must provide equal support for content stored in any of the storage platforms (Section 2.2).

The Rubin Documentation Portal must be capable of supporting several hierarchical browsing schemes for accessing content based on different organizational views of the documentation (Section 2).

The Rubin Documentation Portal should automatically update and sort documentation content, to the greatest extent possible. In other words, curators should only need to maintain documentation in the document's storage platform, without any administrative action through the portal's user interface or infrastructure. A consequence of this requirement is that the Rubin Documentation Portal should not persist information about a document that is not available from the document's own storage platform.

The Rubin Documentation Portal should be secured so that it is only accessible to users with Rubin Observatory staff credentials.

The Rubin Documentation Portal should not maintain fine-grained access control for specific documents or categories of documents. For secured documents, the portal relies upon the security mechanisms of the document's own storage platform. The portal should also reduce its metadata storage of confidential documents to ensure that content cannot be inferred from a search, for example.

3.2 Technical architecture

The architecture described here is based upon that which is already put into production with the portal for public-facing documentation. Starting from this working archetype relieves a great deal of technical risk and development from the new portal's implementation. Both portals would share the use of Algolia (Algolia, 2021a) as a search back-end and Ook (SQuaRE Team, 2021b) as a content indexing service. The Rubin Documentation Portal will have an independent front-end to support the specific Access Views recommended by the Documentation Working Group (Section 2.3). It will also use a separate instance of the Algolia database to eliminate any risks associated with leaking internal documentation to the public-facing portal.

Figure 8 depicts the components of the Rubin Documentation Portal.

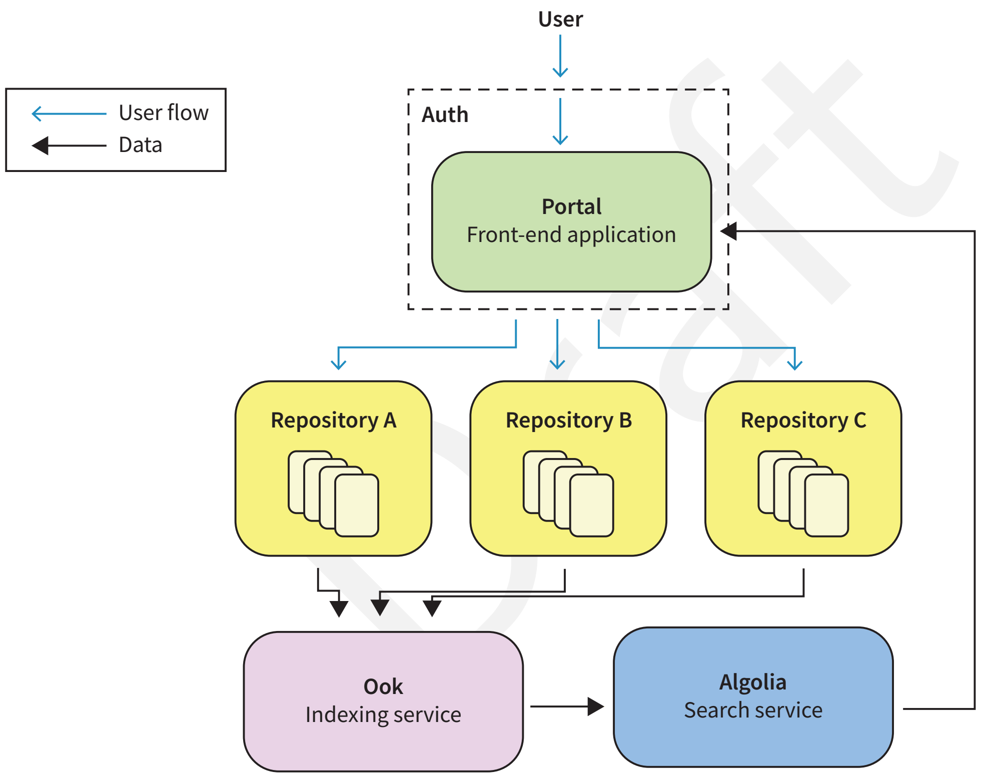


FIGURE 8: Architecture of the Proposed Rubin Documentation Portal. Users find documents on the web portal application, which in turn provides links into the original documentation repositories. Data in the portal application is supplied by the Algolia search service (Algolia, 2021a), which in turn gets its metadata from the original documents in their repositories via the Ook indexing service (SQuaRE Team, 2021b).

3.2.1 Algolia

The core function of the Rubin Documentation Portal is to enable access to documentation through browsing and search. To implement this, it needs a back-end service that contains metadata about Rubin Observatory's documentation holdings and provides interfaces to access and query that metadata from the front-end (i.e., the website). This search database and interface could be made for ”free” with entirely open-source components such as Elasticsearch (Elasticsearch B.V., 2022) and in-house web service. However, a search database and service are sufficiently generic that we cannot add value by making it in-house, and in fact developing, tuning, and operating this service, would be costly in terms of labor. For the public documentation portal, the developers opted to use Algolia and recommend the same choice for the new Rubin Documentation Portal.

Although is currently operating on a free open-source license of Algolia, the new internal Rubin Documentation Portal would be an entirely paid license. Algolia prices based on record counts and request rates. In operating , we found record count to be the limiting factor. At the moment, $1,000$ records costs $1 per month. $1,000,000$ records would cost $850 per month with volume discounts. For reference, the service currently uses $110,000$ records to host all technical notes and change-controlled documents with drafts hosted on LSST the Docs (LTD), also known as lsst.io.

Note that a single document is composed of potentially as many records in Algolia. To optimize full-text search, we break a document into smaller records, generally across section boundaries. Our current algorithms generally produce small record, so it is possible to reduce costs by tuning how we segment content into Algolia records. Furthermore, each sorting option requires a separate pre-sort index. Sorting documents by date, and document number, in addition to relevance, consumes three times as many records as only sorting by relevance.

3.2.2 Ook

The Ook service is responsible for continuously indexing content into Algolia. (SQuaRE Team, 2021b) Whereas we chose Algolia to provide a turn-key search database service, for we chose to build the indexing service in-house to have complete control over how documents are indexed, and what metadata is associated with each document. For example, LaTeX-based documents are indexed based on metadata exported from the Lander PDF landing page generator (also developed in-house), which in turn parses LaTeX syntax in the document source to access metadata such as titles, authors, and so on. The configurability of Ook is beneficial to indexing other types of highly specialized documentation.

The key design principle of Ook is that metadata is extracted from the document as it appears in its repository, rather than requiring direct human interaction with Ook or Algolia to curate the data. This allows Ook to scale well across an organization as large and varied as Rubin Observatory because individual teams manage documents as they already do in the repositories they are already familiar with.

Ook is built such that new content types can be added by writing additional Python-based workflows for each content type. Ook itself provides utilities for queuing ingests, converting content and formatting data for Algolia, and working with the Algolia service itself.

Ook indexing operations can be triggered several different ways. For example, the lsst.io service publishes messages to a Kafka cluster whenever documentation is published on that platform; Ook subscribes to those messages and queues indexing workflows. Ook indexing operations can also be scheduled through an HTTP API. Generally, the goal is to trigger indexing operations automatically whenever the source material changes.

The existing Ook indexing workflows work by downloading content from websites and web services (HTTP APIs). If content is not easily accessible, it would be possible to develop an alternative workflow, such as submitting copies of the document directly to Ook for indexing. Some document repositories may offer online access but have hard-to-use APIs, DocuShare being a prime example. These difficulties can be worked around, for example by emulating a web browser to download content and metadata, but at the cost of more fragile indexing workflows.

Ook is currently operated as a Kubernetes application in the Google Cloud. This arrangement is ideal for minimizing operations cost, and providing convenient scaling.

3.2.3 Front-end application

The front-end web application is how users (Rubin Observatory staff) find documentation. The web application does not provide the document itself—instead, the web application provides a search result card that the user can click on to access the document in its original repository. The Algolia services provides all browsing and search functionality; the front-end application provides the user interface over top of Algolia.

In addition to providing a link to the original document, the portal can also provide an immediate view of a document's metadata. Although the front-end application can show a basic view of a document's metadata based on metadata common to all records, the website can be developed to show additional metadata for different types of documents.

For , we built the site as a React JavaScript application. This allowed us to use and customize the pre-made widgets provided by Algolia for building the user interface.

The front-end application will be accessible only to users who log in. The simplest way to approach this is by putting the application behind a VPN so that the application is completely separate from security concerns. Another approach would be to place the application behind an OAuth proxy to provide a slightly better user experience.

3.3 Support for multiple views

In Section 2, the Documentation Working Group outlines several views for hierarchically arranging documents trees. These views correspond to navigational structures in the front-end application. Although the front-end application code is generally "aware" of the different trees, individual documents are placed in the tree on the basis of metadata in their Algolia records, so that the Algolia service can pre-sort and filter documents into the trees. Since Ook supplies metadata to Algolia, and Ook in turn leverages metadata native to the document and the document's repository, the responsibility for curating documents into different views is the responsibility of individuals managing documents in each repository.

3.4 Information security

Documentation has multiple types of security concerns, such as control over who and how documents are updated, control over who can access documents, and ensuring the long term integrity and preservation of information. Since the Rubin Documentation Portal is not the canonical repository for any documentation, it is not involved in controlling document updates and preservation. The portal's key security concern is access control.

The portal is designed to only provide authentication-based access control. Any Rubin Observatory staff member with credentials can access any metadata records contained within the Rubin Documentation Portal. Once a user selects a document to view, they are forwarded to that documentation repository and must authenticate with that repository and be subject to its access control rules.

However, the metadata contained in the Rubin Documentation Portal can be potentially rich, even including the full-text content of a document to enable search functionality. It is conceivable that some of this metadata may not be appropriate for observatory-wide access (such as information with export controls). In these cases, the most realistic approach to preserving strict access controls in these situations is to limit what metadata is available. In order of strictness, the following approaches can be used:

1. Omit full-text content of a document from Algolia records.
2. Limit or obfuscate other metadata (such as titles) in the Algolia records.
3. Omit the individual documents altogether from Algolia and instead link to a documentation landing page hosted by the secure document repository itself.

In all cases, controlling how documents are indexed is done by configuring the indexing service, Ook.

4 Lingual translations

Due to the nature of the Rubin Observatory, it is inevitable that some documents will need to be translated into other languages. The major need will be English and Spanish translations for the American and Chilean operational locations. It is a significant yet inevitable undertaking for bilingual translations throughout the observatory lifetime. English/Spanish bilingual documents will be needed, and they must be available with up-to-date information to prevent issues during construction and maintenance. The accuracy of language is crucial (e.g., context, terminology), from work instructions to inclusivity (e.g., engagement in continuous improvement). The risk of improper or unavailable translations can generally impact the schedule (e.g., delay in work) and results in increased risk to personnel, equipment, the environment or security when procedures or protocols cannot be accurately followed.

The Documentation Working Group recommends that, at a minimum, the following translated in both, English and Spanish:

- procedure documents and manuals for activities conducted on a routine basis,
- any and all safety documents, and
- websites that facilitate the organization and location of translated documentation (e.g. DocuShare).

Any other documents, drawings, schematics, parts lists, etc. should be translated on an as-needed basis.

Workflows for documentation should include steps to determine if translation is needed and its implement, as defined by the project of technical group. The determination and reason for translation should be recorded; i.e., required, preferred, suggested or unneeded. When releasing documents, at least one approver should be designated specifically to confirm that the translated version matches the original. This is especially important when revisions are made, to ensure any changes are fully and accurately captured. It should be clear within the document or applicable sections which have translation and the respective location(s).

When producing translations, the Documentation Working Group recommends that they are reviewed by a technically-competent bilingual person. For version-controlled documents, it is recommended that original and translated versions be kept in a single file. The location for translated documents can be done in whatever format is most appropriate for the document; e.g., a schematic might have both English and Spanish labels next to each other, or a procedure may be written entirely in English and then repeated in Spanish.

The construction project should be involved as much as practical, but the effort will continue into operations. Additionally, the Documentation Working Group believes it is a requirement for the project to allocate resources for ongoing support throughout operations. While estimating the resources needed for this, it is recommended to consider additional translation needs for global associations, including user-facing scientific documentation.

5 References

Algolia, 2021a, The flexible ai-powered search & discovery platform, URL https://www.algolia.com/,
Accessed: 2021-06-06

Algolia, 2021b, SaaS: Secure & reliable search that scales with your business, URL https://www.algolia.com/solutions/saas/,
Accessed: 2021-06-06

Atlassian, 2021, Jira: Issue & project tracking software, URL https://www.atlassian.com/software/jira,
Accessed: 2021-06-06

Atlassian Corporation, 2021, Confluence: Your remote-friendly team workspace, URL https://www.atlassian.com/software/confluence,
Accessed: 2021-06-06

[LSE-489], Blum, B., Ivezić, Ž., Kahn, S., Krabbendam, V., 2020, Charge to the Project-Wide Documentation Working Group, URL https://lse-489.lsst.io/,
Vera C. Rubin Observatory LSE-489

[SITCOMTN-005], Claver, C., Bauer, A., Bechtol, K., et al., 2021, Construction Completeness and Operations Readiness Criteria, URL https://sitcomtn-005.lsst.io/,
Vera C. Rubin Observatory Commissioning Technical Note SITCOMTN-005

[SITCOMTN-012], Claver, C., Cabrera, D., McKercher, R., et al., 2021, Rubin Observatory Construction Documentation Inventory, URL https://sitcomtn-012.lsst.io/,
Vera C. Rubin Observatory Commissioning Technical Note SITCOMTN-012

Drupal Association, Open source content management tools, URL https://www.drupal.org/,
Accessed: 2022-11-10

Elasticsearch B.V., 2022, Free and open search: The creators of elasticsearch, elk, kibana, elastic, URL https://www.elastic.co/,
Accessed: 2022-10-10

GitHub, Inc., Where the world builds software, URL https://www.github.org/,
Accessed: 2021-06-06

Intercax, LLC, The digital thread platform for model-based engineering, URL https://intercax.com/products/syndeia/,
Accessed: 2022-11-10

[Document-36788], McKercher, R., 2020, *DocuShare Options Trade Study*, Document-36788, URL https://docushare.lsst.org/docushare/dsweb/Get/Document-36788

No Magic, Inc., Magicdraw, URL https://www.3ds.com/products-services/catia/products/no-magic/magicdraw/,
Accessed: 2021-06-06

Oracle Corporation, Primavera p6 enterprise project portfolio management, URL https://www.oracle.com/construction-engineering/primavera-p6/,
Accessed: 2024-03-20

[RDO-71], Roberts, A., Blum, R., Claver, C., et al., 2023, Rubin Observatory Risk and Opportunity Management Plan, URL https://rdo-71.lsst.io/,
Vera C. Rubin Observatory RDO-71

[RTN-076], Rumore, M., Guy, L., 2024, Migration Plan for Construction Project Documentation to Operations, URL https://rtn-076.lsst.io/,
Vera C. Rubin Observatory Technical Note RTN-076

[SQR-000], Sick, J., 2015, The LSST DM Technical Note Publishing Platform, URL https://sqr-000.lsst.io/,
Vera C. Rubin Observatory SQuaRE Technical Note SQR-000

[SQR-006], Sick, J., 2016, The LSST the Docs Platform for Continuous Documentation Delivery, URL https://sqr-006.lsst.io/,
Vera C. Rubin Observatory SQuaRE Technical Note SQR-006

Software Freedom Conservancy, Inc., Open source distributed version control system, URL https://git-scm.com/,
Accessed: 2022-11-10

Solidworks, 2021, Solidworks PDM, URL https://www.solidworks.com/product/solidworks-pdm,
Accessed: 2021-06-06

SQuaRE Team, 2021a, *Vera Rubin Observatory Technical Documentation Portal*, URL https://github.com/lsst-sqre/www_lsst_io,
Accessed: 2022-10-10

SQuaRE Team, 2021b, Ook, librarian indexing bot, URL `https://github.com/lsst-sqre/ook`, Accessed: 2022-10-10

Tecnoteca srl, openmaint property & facility management, URL `https://www.openmaint.org/`, Accessed: 2024-03-20

Wikimedia Foundation, 2021, Tree structure, URL `https://en.wikipedia.org/wiki/Tree_structure`,
Accessed: 2022-05-23

Xerox, 2021, Docushare enterprise content management platform, URL `https://www.xerox.com/en-us/services/enterprise-content-management/docushare`,
Accessed: 2022-10-10

6 Acronyms

Acronym	**Description**
API	Application Programming Interface
CMMS	Computerized Maintenance Management System
CSC	Commandable SAL Component
ComCam	The commissioning camera is a single-raft, 9-CCD camera that will be installed in LSST during commissioning, before the final camera is ready.
DM	Data Management
DMTR	DM Test Report
DOE	Department of Energy
EAS	Environmental Awareness System
EFD	Engineering and Facility Database
EPO	Education and Public Outreach
FMEA	failure modes and effect analysis
FRACAS	Failure Reporting Analysis and Corrective Action System
GIS	Global Interlock System
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ICD	Interface Control Document
LOVE	LSST Operators Visualization Environment
LSE	LSST Systems Engineering (Document Handle)

LSST	Legacy Survey of Space and Time (formerly Large Synoptic Survey Telescope)
LSSTC	LSST Corporation
LaTeX	(Leslie) Lamport TeX (document markup language and document preparation system)
MIE	Major Item of Equipment
MREFC	Major Research Equipment and Facility Construction
NOIRLab	NSF's National Optical-Infrared Astronomy Research Laboratory; `https://noirlab.edu`
NSF	National Science Foundation
P6	Primavera, a comprehensive project management tool
PCW	Project Community Workshop
PDF	Portable Document Format
PDM	Phase Dispersion Minimization
RDO	Rubin Directors Office
REST	REpresentational State Transfer
ROO	Rubin Observatory Operations
RTN	Rubin Technical Note
S3	(Amazon) Simple Storage Service
SAL	Service Abstraction Layer
SE	System Engineering
SIT	System Integration, Test
SLAC	SLAC National Accelerator Laboratory
SP	Story Point
SQR	SQuARE document handle
SQuaRE	Science Quality and Reliability Engineering
SaaS	Software as a Service
T&S	Telescope and Site
TS	Test Specification
URL	Universal Resource Locator
VPN	virtual private network
WBS	Work Breakdown Structure
WFD	Wide Fast Deep
XML	eXtensible Markup Language