



LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST) Science Platform Design

Gregory Dubois-Felsmann and Kian-Tat Lim

LDM-542

Latest Revision: 2017-06-26

Draft Revision NOT YET Approved – This LSST document has been approved as a Content-Controlled Document by the LSST DM Technical Control Team. If this document is changed or superseded, the new document will retain the Handle designation shown above. The control is on the most recent digital document with this Handle in the LSST digital archive and not printed versions. Additional information may be found in the corresponding DM RFC. – **Draft Revision NOT YET Approved**

Abstract

This document describes the design of the LSST Science Platform, the primary user-facing interface of the LSST Data Management System.

Change Record

Version	Date	Description	Owner name
1	YYY-MM-DD	Unreleased.	Gregory Dubois- Felsmann, K-T Lim

Draft

Contents

1	Introduction	1
1.1	The Portal Aspect	1
1.2	The Notebook Aspect	1
1.3	The API Aspect	1
2	The Architecture of the Science Platform	1
2.1	Design Overview	1
2.1.1	Functional Architecture	1
2.1.2	Deployment Architecture	1
2.2	Data Access and Storage	1
2.2.1	Databases	1
2.2.2	Images	2
2.2.3	LSST-specific Objects	2
2.2.4	Data Access Services	2
2.2.5	User Catalog Data Support	2
2.2.6	User Workspace Storage	2
2.2.7	Data Access Permissions and Quotas	2
2.2.8	Support of Previous Releases	2
2.3	Computing Resources	2
2.3.1	Basic user compute services	3
2.3.2	Large-scale batch and parallel computing	3

2.3.3	Resource management	3
2.4	Authentication and Authorization	3
2.5	Cybersecurity Considerations	3
2.6	Additional Support Services	3
3	The Portal Aspect	3
3.1	Generic Data Browsing	3
3.2	Documentation Delivery	3
3.3	Semantics-aware Workflows	3
3.4	Use of the LSST Stack	3
3.5	Extensibility of Visualizations	3
3.6	Extensibility of the UI	3
4	The Notebook Aspect	3
4.1	JupyterHub / JupyterLab service	3
4.2	Pre-installed LSST software	3
4.3	Pre-configured access to LSST data	4
5	The API Aspect	4
5.1	Overview - VO services and custom LSST services	4
5.2	Catalog and other tabular data access	4
5.2.1	TAP and other VO-compliant services	4
5.2.2	Non-VO catalog interfaces	5
5.3	Image data access	5

5.3.1	Image finding	5
5.3.2	Image retrieval	5
5.4	Metadata access	5
5.4.1	Data Releases	5
5.4.2	Tables	6
5.4.3	Table Structure	6
5.5	User Workspace access	6
6	The Interconnectedness of the Science Platform	6
6.1	Sharing Data	6
6.2	Sharing Queries	6
6.3	Sharing State	6
7	Application of the Science Platform inside the Project	7
7.1	Developer Support	7
7.2	Commissioning	7
7.3	Observatory Operations	7

Science Platform Design

1 Introduction

1.1 The Portal Aspect

1.2 The Notebook Aspect

1.3 The API Aspect

2 The Architecture of the Science Platform

2.1 Design Overview

2.1.1 Functional Architecture

2.1.2 Deployment Architecture

2.2 Data Access and Storage

2.2.1 Databases

2.2.1.1 Database content overview

- Image and Visit metadata
- Catalogs
- Composite data
- Observatory metadata (including EFD)
- Reference catalogs

2.2.1.2 Database technologies

- Conventional
- Qserv

2.2.2 Images

2.2.3 LSST-specific Objects

("ORM-ish" behavior, composite objects, etc.)

2.2.4 Data Access Services

(This is a description of the functional architecture of the services; a detailed description of the APIs offered by the services is in section 5 below)

2.2.5 User Catalog Data Support

("Level 3 catalogs")

2.2.6 User Workspace Storage

(Generic file-oriented storage, assumed to also support "Level 3 image data", e.g., custom coadds)

2.2.7 Data Access Permissions and Quotas

2.2.8 Support of Previous Releases

(This could go elsewhere in the outline hierarchy)

2.3 Computing Resources

2.3.1 Basic user compute services

(including the services used for the JupyterLab Python kernels)

2.3.2 Large-scale batch and parallel computing

2.3.3 Resource management

2.4 Authentication and Authorization

2.5 Cybersecurity Considerations

2.6 Additional Support Services

3 The Portal Aspect

3.1 Generic Data Browsing

3.2 Documentation Delivery

3.3 Semantics-aware Workflows

3.4 Use of the LSST Stack

3.5 Extensibility of Visualizations

3.6 Extensibility of the UI

4 The Notebook Aspect

4.1 JupyterHub / JupyterLab service

4.2 Pre-installed LSST software

4.3 Pre-configured access to LSST data

5 The API Aspect

The Web APIs provide language-agnostic, location-agnostic authenticated access to LSST data. They are intended to be used by client tools and automated processes. They are not optimized for direct human use via a browser or command line URL transfer tool, although they can of course be used that way. The available endpoints and the URL structures they accept will be documented.

The Web APIs are undergoing detailed design. Initial prototypes exist that provide non-VO-compliant access to catalogs, images, image mosaics, and image cutouts. This section describes the ultimate goals for the LSP Web APIs and documents design features where they are already known.

5.1 Overview - VO services and custom LSST services

The Web APIs support standard VO protocols where possible. They will also support extensions to the VO protocols and additional custom LSST services that are useful for the Portal Aspect or JupyterLab Aspect. To the extent that it makes sense, such extensions will be proposed to the IVOA for incorporation into the standard protocols.

5.2 Catalog and other tabular data access

Catalogs, metadata, provenance, and other tabular data are accessed through the `dbserve` endpoint.

5.2.1 TAP and other VO-compliant services

SCS and TAP will be supported.

5.2.1.1 ADQL implementation The ADQL implementation will support basic SQL92 (without complex subqueries) and region specification extensions. Slight variations in the dialect

may be visible between the "conventional" RDBMS back-end and the Qserv distributed back-end.

5.2.1.2 Return data formats Returned data formats include VOTable, compressed JSON, and SQLite.

5.2.2 Non-VO catalog interfaces

A custom interface will provide metadata about table columns. See section 5.4.3.

5.3 Image data access

Images and other files are accessed through the `imgserv` endpoint.

5.3.1 Image finding

Images may be discovered via SIA data discovery or TAP query on the image metadata tables.

5.3.2 Image retrieval

Images may be retrieved via the URL returned in an SIA request or via SODA. For convenience, key/value pairs as used by the Data Butler client library may be provided to a custom URL interface.

5.4 Metadata access

Catalog and image metadata can be accessed via TAP query to provided metadata tables.

5.4.1 Data Releases

Each Data Release has its own set of tables. These are expected to differ in schema between Data Releases. In addition, the image and catalog metadata available for each Data Release

will vary. Finally, identifiers within catalogs and images (except raw image identifiers) will vary between Data Releases.

Accordingly, the Data Release number is a key parameter that must be presented to the Web APIs. Collections of datasets that are not part of a Data Release (e.g. in-process, unreleased data or intermediate data products) are assigned labels that are used in place of the Data Release number.

The number of Data Releases available through the LSP and therefore the Web APIs aspect is discussed in section 2.2.8,

5.4.2 Tables

Available tables include those in ? .

5.4.3 Table Structure

Each table will have associated metadata that records, for each column, a description, UCD when appropriate, unit when appropriate, and any relationship with other columns (e.g. so several columns making up a covariance matrix can be identified as such without resorting to pattern-matching of column names).

5.5 User Workspace access

WebDAV and/or VOSpace will be provided for user workspace access.

6 The Interconnectedness of the Science Platform

6.1 Sharing Data

6.2 Sharing Queries

6.3 Sharing State

7 Application of the Science Platform inside the Project

7.1 Developer Support

7.2 Commissioning

7.3 Observatory Operations

References

Draft