LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)

LSST Science Platform Vision Document

M. Jurić, D. Ciardi, and G.P. Dubois-Felsmann

LDM-nnn

Latest Revision: 2017-05-22

Abstract

This document defines the high-level vision for the **LSST Science Platform (LSP)**, a set of web applications and services through which the the scientific community will to access, visualize, interact with, and analyze LSST data holdings.

It is meant to inform the development of requirements, product specifications, prioritization, and plans for the Agile development of elements of the DM system that together make up the LSP.

Change Record

Version	Date	Description	Owner name
1	2017-03-15	Initial high-level description of the concept	Mario Juric

Contents

LSST Science Platform Vision Document

1 Preface

The purpose of this document is to lay out the high-level vision for the **LSST Science Platform (LSP)**, a set of web applications and services through which the the scientific community will to access, visualize, interact with, and analyze LSST data holdings. With its companion document – the Data Products Definition Document (DPDD) – it defines the high-level vision for LSST's end-user deliverables.

To a future LSST user, this document should illustrate what will be made available to the science community through the LSST Data Access Centers. To LSST builders, it provides direction on how to flow down the LSST System Requirements Document (LSR) to Data Management requirements (DMSR), DM System design (**?**) and associated documents), sizing (**?** and related documents), budget, and schedule as they pertain to the end-user services provided at the LSST Data Access Centers.

Though under strict change control[1], this is a ***living document***. LSST will undergo a period of construction and commissioning lasting no less than seven years, followed by a decade of survey operations. To ensure their continued scientific adequacy, the high-level vision for LSST Science Platform will be periodically reviewed and updated.

[1]LSST Docushare handle for this document is LSE-XXX.

2 Introduction

2.1 Goals

The LSST is a facility whose primary mission is to acquire, process, and make available the data collected by its telescope and camera, as well as enable "next-to-the-data" creation of added-value (Level 3) data products (SRD).

This document describes the vision for the services to be put into place to fulfill the "*making available*" and "*Level 3* creation" aspects of LSST's mission. Its aim is to present a high-level description of the data access and analysis services provided at the LSST Data Access Centers. It should be read in conjunction with the LSST Data Products Definition Document (DPDD), which provides the high-level description of LSST data products.

2.2 LSST Science Platform Overview

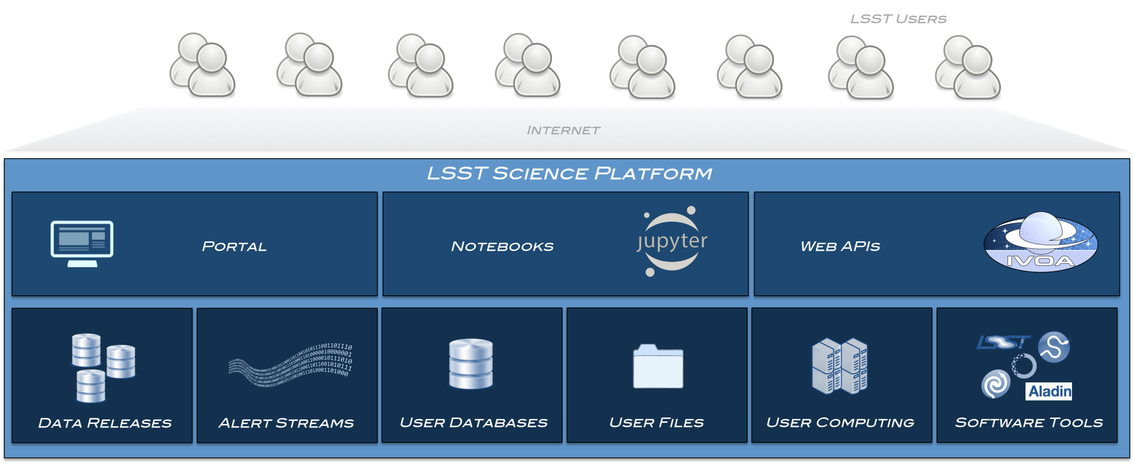

FIGURE 1: A high-level, layered, view of the LSST Science Platform. The LSST data will be exposed to the users through the web Portal, the Jupyter Notebook interface, and machine-accessible Web APIs. The web Portal component will provide the essential data access and visualization services common to present day archives. The Notebook component, based on the Jupyter family of technologies (JupyterHub and JupyterLab) will allow for more sophisticated next-to-the-data analysis. These user-visible services will provide acces to the underlying core LSST data sets – the data releases and alert streams – and be supported by the user Database, File Storage, Computing, and Software Tools components. Together, they will enable the users to access, sub-select, analyze, and perform added-value processing of all flavors of LSST Data Products (see text for detail).

We define the **LSST Science Platform as a set of web applications and services made available to the scientific community to access, visualize, subset, and perform next-to-the-data analysis of the LSST data set**. It represents the integrated set of services that will be offered to LSST users.

The platform exposes the LSST data and services to the user through three primary user-facing "*aspects*" – the web **Portal**, the **JupyterLab** analysis environment, and a machine-accessible **Web API** interface. These aspects provide three different way to access the data sets and analysis services provided in the LSST Data Access Centers (Figure 1).

The **Portal** aspect is a web portal designed to provide the essential data access and visualization services through a simple-to-use website. It will enable browsing and visualization of the available datasets in ways the users are accustomed to at archives such as IRSA, MAST, or the SDSS archive. We describe it in more detail in Section 3.1.

The **JupyterLab** aspect will provide a Jupiter Notebook-like interface, and is geared towards enabling next-to-the-data analysis. The user experience will be nearly identical to working with Jupyter notebooks locally, except that computation and analysis will occur at resources provided at the LSST Data Access Center. This is an implementation of the "bringing computation to the data" paradigm: rather than imposing the burden of downloading, storing, and processing (potentially large) subsets of LSST data at their home institutions, we will enable our users to bring their codes and perform their analysis at the LSST DAC. This reduces the barrier to entry and shorten the path to science for the LSST science community. We describe it in more detail in Section 3.2.

The third, **Web API**, aspect of the LSST Science Platform will expose the services offered by the LSST Data Access Centers to other software tools and services using commonly accepted protocols. For example, industry-standard protocols such as WebDAV may be used to expose file data, or Virtual Observatory protocols such for catalogs or images (TAP and SIAP, respectively). This interface will open the possibility for remote access and analysis of the LSST data set using applications that the users are already comfortable with (eg., such as TOPCAT or libraries like AstroPy). Furthermore, the offered APIs will allow for federation with other astronomical archives, bringing added value to the LSST dataset. We describe it in more detail in Section 3.3.

Finally, the LSST Science Platform is being envisioned to enable and encourage collaborative work. The capabilities ranging from sharing of derived datasets within smaller groups, collaborations, or with the broader LSST community, to collaborative visualization and editing capabilities expected to become available within the JupyterLab ecosystem.

3 User-facing Services

3.1 Web Portal

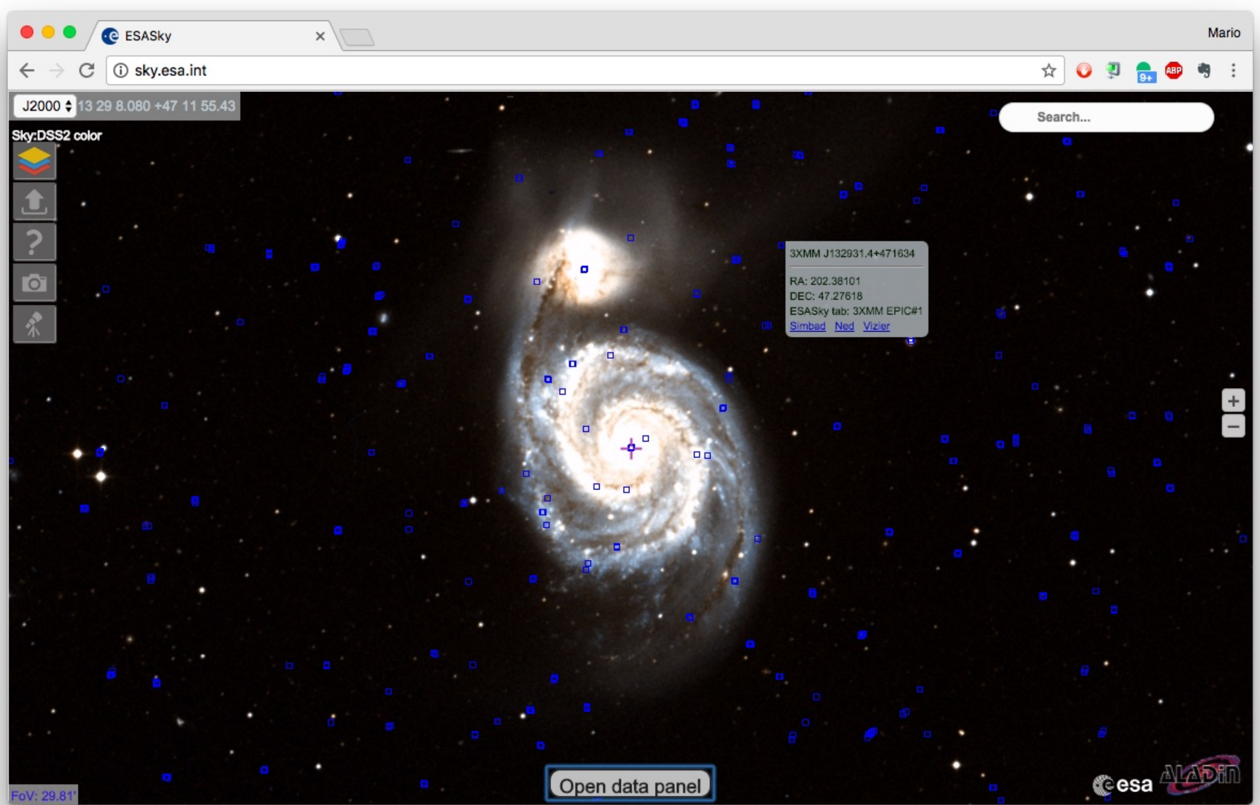

FIGURE 2: The "ESA Sky" web portal interface to ESA Archive holdings. The LSST portal user experience will support similar modern pan/zoom/select metaphor for exploration and visualization of the LSST data set.

The **Portal** aspect is a web portal designed to provide the essential data access and visualization services through a simple-to-use website. It is to enable browsing and visualization of the available datasets in ways the users are accustomed to at archives such as IRSA, MAST, or the SDSS archive, with an enhanced level of interactivity in line with expectations for then-

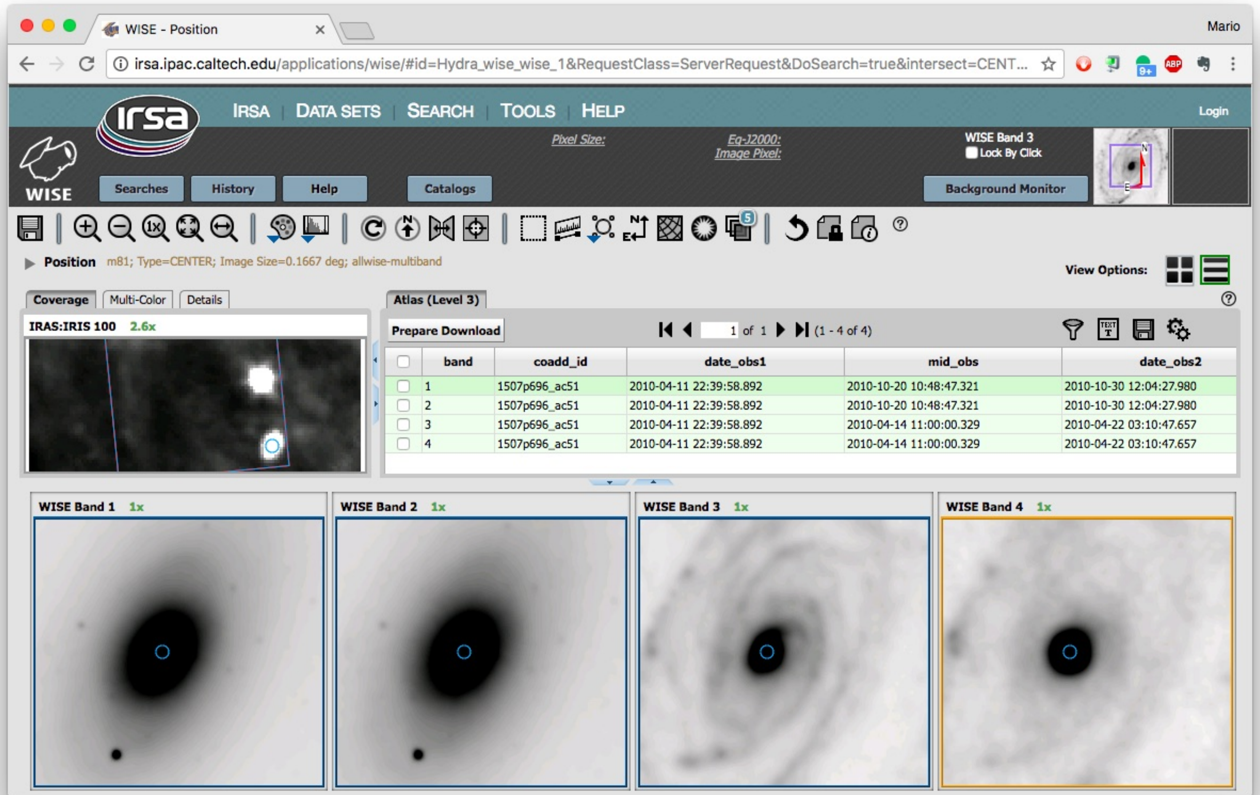

FIGURE 3: The web portal interface to the WISE data set at the Infra-Red Science Archive at IPAC. The LSST portal is being built by extending the Firefly toolkit that powers the IRSA/WISE archive.

contemporary archive portals (similar to that found in ESASky and the DECaLS Viewer). Examples of the types of user experiences to be offered through the LSST portal are shown in Figures 2 and 3.

Through the Portal, the users will be able to view the LSST images, request subsets of data (via simple forms or SQL queries), store the results of such queries to their personal workspaces, as well as download it. The Portal will also make it possible to construct commonly requested plots, and generally explore the LSST dataset in a way that allows them to identify and access (subsets of) data required by their science case.

Virtually all LSST users will use the Portal as their first point of entry to exploring and analyzing the LSST data set. When developing the Portal, we will therefore emphasize the user experience and exploratory capabilities, over analysis features. The latter are expected to be more directly satisfied by the JupyterLab aspect.

3.2 JupyterLab

The **JupyterLab** aspect, based on the Jupyter family of technologies (such as JupyterHub and JupyterLab), will be provided to allow for more sophisticated data selection, analysis, and creation of added value (Level 3) data products. A screen capture of a mature prototype of JupterLab is shown in Figure 4.

The JupyterLab user experience will be nearly identical to working with Jupyter notebooks locally, except that computation and analysis will occur at resources provided at the LSST Data Access Center. This is an implementation of the “bringing analysis to the data” paradigm: rather than imposing the burden of downloading, storing, and processing (large) subsets of LSST data at their home institutions, we will enable our users to bring their codes and perform their analysis at the LSST DAC. We expect this will reduce the barrier to entry and shorten the path to science for the LSST science community.

We will provide JupyterLab instances to LSST users in an environment carrying a library of preinstalled commonly used and useful software tools: AstroPy, the LSST science pipelines, Anaconda Scientific Python Distribution, and others. The users will be able to upload and install their own tools as well.

The JupyterLab aspect of the science platform will play a key role in commissioning, quality

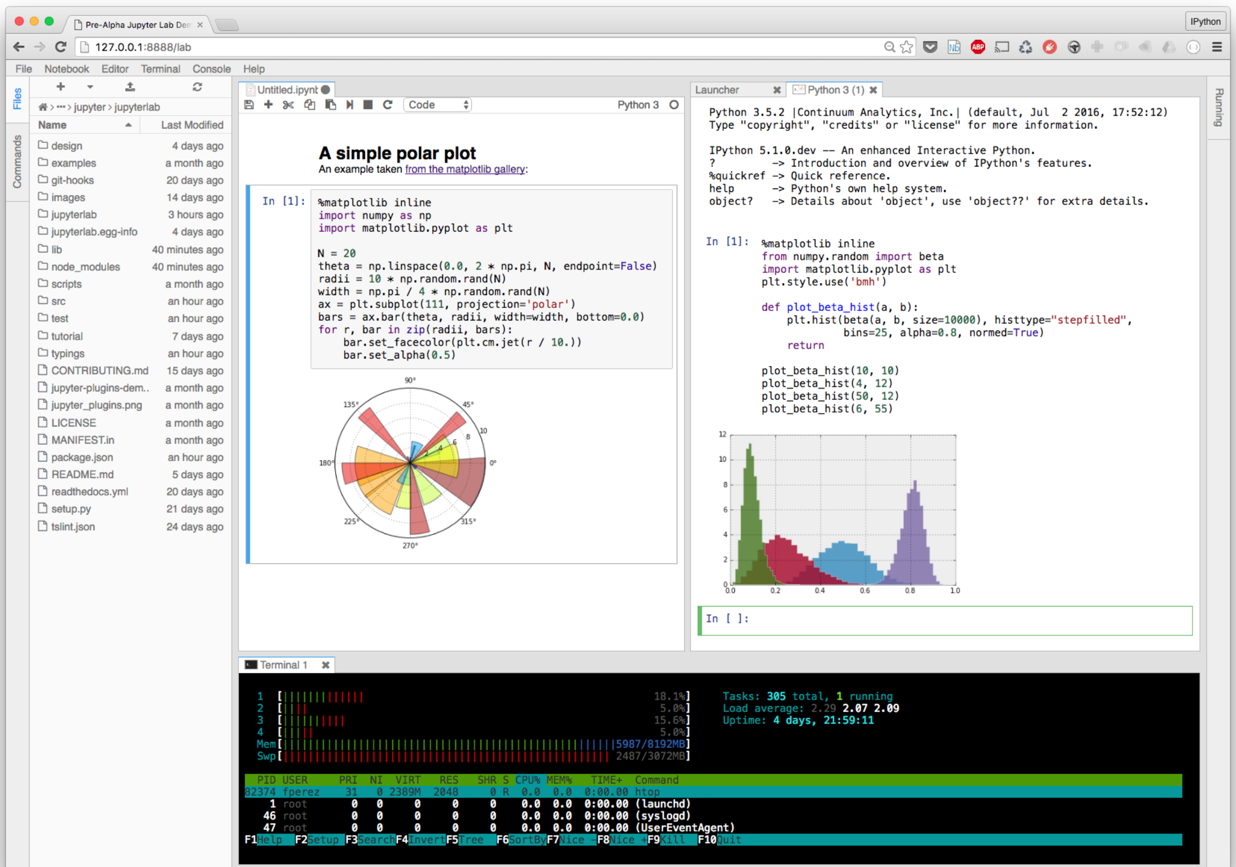

FIGURE 4: A screen capture of the JupyterLab interface (credit: JupyterLab team, `http://blog.jupyter.org/2016/07/14/jupyter-lab-alpha/`)

assessment, and science validation of the as-built system. It will be the primary method of performing interactive analysis of acquired data (e.g. adjusting and executing prepared notebooks driving commissioning tasks), as well as commanding the batch resources to execute larger processing tasks. Due to this, we expect the JupyterLab aspect to reach maturity earlier than the others, and certainly in time for commissioning.

3.3 Web APIs

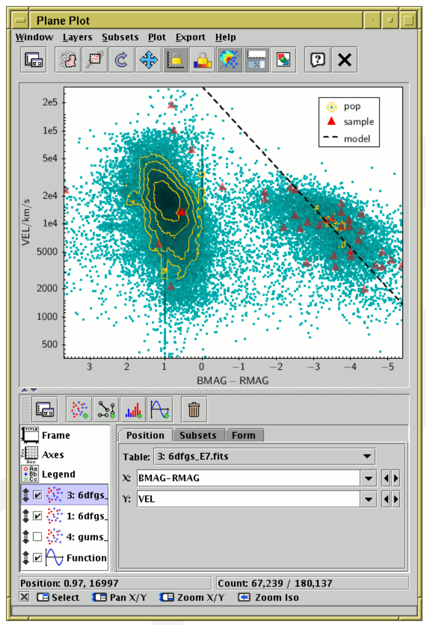

FIGURE 5: A screen capture of Tool for OPerations on Catalogues And Tables (TOPCAT), that is capable of remotely accessing catalogs and images using VO protocols. Tools such as these will be able to directly access the data sets served by the LSST DACs (figure credit: Mark Taylor, `http://www.star.bris.ac.uk/~mbt/topcat/sun253/sun253.html`).

Backend Platform services – such as access to databases, images, and other files – will be exposed through machine-accessible web APIs. These will serve the data using community-accepted formats and protocols, making it easy to remotely access the LSST data and DAC services. Furthermore, to ensure maximal exposure of the DAC services through the Web APIs, the other two aspects of the Platform – Portal and JupyterLab – will internally access the LSST datasets using the same Web APIs.

Exposing the LSST data through Virtual Observatory interfaces plays a particularly important role. It will allow federation of the LSST data set to other archives, and enable the use of standard tools such as TOPCAT or DS9 by the end-users. The latter will further lower the barrier to access to LSST data, shortening the path to science.

While this document does not proscribe a full set of protocols and formats to expose the LSST data, VO Simple Cone Search and TAP (for catalogs) and SIAP (for images) must be supported.

3.4 Integrated environment

All aspects of the Platform are intended to be *well integrated*, enabling a seamless workflow so the users to be able to move back and forth between them as needs dictate. The aim is to enable a user to find or create data in one aspect, and view or analyze that data in another.

As an example of how these connections can aid a user in exploring the LSST data, data queries will be shareable across the Portal and JupyterLab. This will allow a user to build a query using the Portal user interface, view the (possibly preliminary) results by browsing it in the Portal, and then access the final results from a JupyterLab notebook or a remotely connected client (e.g. TOPCAT) for further analysis. The reverse flow will also be enabled; a user can code and submit a complex SQL query in the Notebook and then browse, and visualize the results in the Portal.

By making the environments integrated, we allow for a shallower learning curve and a gradual transition to more complex environments once needed. For example, a user may begin interacting with the LSST dataset using the Portal but may ultimately reach the limitations of selection tools exposed through that aspect. The integrated nature of the platform allows such user to switch to JupyterLab, and access and continue working on the data hitherto analyzed in the Portal.

3.5 Supporting Collaborative Work

The LSST Science Platform will allow for collaborative work at two levels:

- **Workspaces**: Creation and sharing of data sets – catalogs, images, queries, and other data products – within a pre-defined group (eg., a research group at a University, or a large science collaboration). Such groups would have access to a shared virtual “workspace” within the LSST DAC. This workspace will include shared files, shared catalogs (stored in user databases) as well computing cycles allocated to the group as a whole. This shared workspace will be equally "visible" from all three aspects of the platform – e.g., uploads to the workspace will be possible either through a form in the Portal, from JupyterLab, or using a file transfer client.

- **Shared editing**: We look to offer a "Google Docs"-like collaborative editing, visualization, and data analysis capabilities. These capabilities are expected to become available within the Jupyter ecosystem, and will be included in the JupyterLab aspect as they do.

The levels of support for collaboration described above are responsive to the large majority of user needs identified by end-user focus groups in R&D. At the same time, they minimizing the technical risks by leveraging widely used and well understood technologies (SDSS-like MyDB user databases, backend A&A mechanisms, VO protocols, Jupyter).

4 Backend Services

** UNFINISHED **

While

4.1 Serving Level 1 and 2 Data Products

** UNFINISHED **

All data products will be made available using user-friendly databases and web services. More

In this section, we describe the high-level concepts for services used to serve the LSST catalogs and the event alert stream. Further details of the LSST data products may be found in the LSST Data Products Definition Document (DPDD).

5 Enabling Level 3: User Databases, File storage, and Computing

** UNFINISHED **

–Need to say something about a batch system and MyDB type functionality being available to the user. May want to move some of the below discussion to JupyterLab section? Is this the place to introduce the Workspace concept?–

Queries, visualizations, and analysis performed through the Portal and Notebooks will be served by a shared computing cluster, file storage, and database resource (bottom row of Figure 1). At start of operations, this computing cluster will number 2,400 cores (approximately 18 TFLOPs), with 4 PB of file and 3 PB of database storage (numbers for the U.S. DAC). These will be shared by all users, the number of whom we're estimating in the low thousands.

Not all users will be accessing the computing cluster concurrently; though difficult to predict with accuracy because of a lack of direct comparables, an estimate on order of a 100 concurrent users is likely reasonable. This would translate to typical allocations of 20 cores per user, sufficient to enable preliminary end-user science analyses (working on catalogs, smaller number of images) and creation of some added-value (Level 3) data products. A good analogy is one of being given a server with a few TB of disk, few TB of database storage, that is co-located next to the LSST data, and with a chance to use tens to hundreds of cores for analysis (depending on system load).

For larger endeavors (e.g., pixel-level reprocessing of the entire LSST dataset), the users will be steered towards resources beyond the LSST DACs (e.g., national supercomputing centers, university computing centers, or the public cloud).

6 Development Methodology and Prioritization Guidance

6.1 Iterative Development Leveraging Existing Technologies

The services constructed for the LSST Science Platform will be developed following the iterative, Agile methodology. This is especially desirable for user-facing services, where iterative development and nearly continuous stakeholder feedback provide guidance as to the details of features to be implemented, and the delivery of intermediate milestones.

The development of the LSP Portal. JupyterLab, as well as Web API aspects will start from significant existing code bases, standards, and prior art. This is a deliberate approach designed to minimize risk, and leverage end-user familiarity with these interfaces, thus reducing the barrier to adoption of the interface eventually deployed for LSST.

The **Portal** is based on existing, production quality, archive portal interface developed at IRSA/IPAC – the *Firefly* toolkit. The primary challenge is integrating the existing Firefly code, and updating the user experience to conform to anticipated user expectations (e.g., supporting all-sky maps and pan/zoom/click-type exploration). Consistent with the general philosophy, DM should look at achieving the necessary upgrades by re-using existing well-known tools and libraries (e.g. Aladin Lite).

Similarly, the **JupyterLab** environment will be based on the open-source JupyterLab product delivered and maintained by the Juptyter team. The development of the JupyterLab aspect of the LSST Science Platform should focus on deployment and integration with the backend services and other aspects of the platform, rather than developing new or radically different features within the JupyterLab product.

Finally, the **Web API** aspect is envisioned as implementing existing, widely-adopted, community protocols (e.g. such as those from Virtual Observatory suite of protocols and standards), down to leveraging existing libraries wherever appropriate.

6.2 Prioritization Guidance

Here we give some overall feature prioritization guidance, to enable the construction of initial (mostly functional) requirements and intermediate development milestones.

Portal aspect:

1. Deployment of the initial Firefly back-end within the (prototype) LSST Data Access Center at NCSA.
2. Integration of the initial Firefly front- and back-ends with the LSP backend services. For example, this includes the authentication and authorization mechanisms, relational databases, file stores, etc.
3. User experience improvements, such as addition all-sky maps with pan/zoom/select navigation metaphors, modernization of the look-and-feel, streamlining of the UI and deprecation of rarely used widgets. **Once this level of functionality is met (at scale), the Portal aspect will have achieved the minimum level of viability for deployment to operations**.

4. Improved user workflow integration with other aspects of the LSST Science Platform. For example, it should be possible to begin data exploration in the Portal (e.g., by interactively selecting data sets) and seamlessly transfer the sub-selected catalogs and images to the JupyterLab environment for further, more complex, analysis using provided Python libraries.

5. Addition of new new widgets and abilities to the Portal, that address most requested and broadly useful end-user needs.

6. Widget-level integration with JupyterLab.

JupyterLab aspect:

1. Deployment of the initial JupyterLab product within the (prototype) LSST Data Access Center at NCSA.

2. Integration of the JupyterLab product with LSP backend services, most notably authentication and authorization, user management, databases, and file stores. **Once this level of functionality is met (at scale), the JupyterLab aspect will have achieved the minimum level of viability for deployment to operations**.

3. Development of libraries and utilities to ease the submission of user-written code from Jupyter notebooks to the batch system.

4. Creation and curation of a library of 3rd party code that will be made available to LSP end-users.

Web APIs:

1. Development and deployment of initial data access APIs needed to satisfy the backend needs of the Portal and JupyterLab aspects. These may not yet "speak" the final, standards-compliant, protocols.

2. Integration of the Web API aspect with LSP backend services, most notably authentication and authorization, user management, databases, and file stores.

3. Deployment of standards-compliant protocols throughout the Web API aspect, and integration with all other elements of the Platform. **Once this level of functionality is met**

(at scale), the Web API aspect will have achieved the minimum level of viability for deployment to operations.