LSST Data Exploitation

Gregory Dubois-Felsmann & Tim Jenness

Large Synoptic Survey Telescope

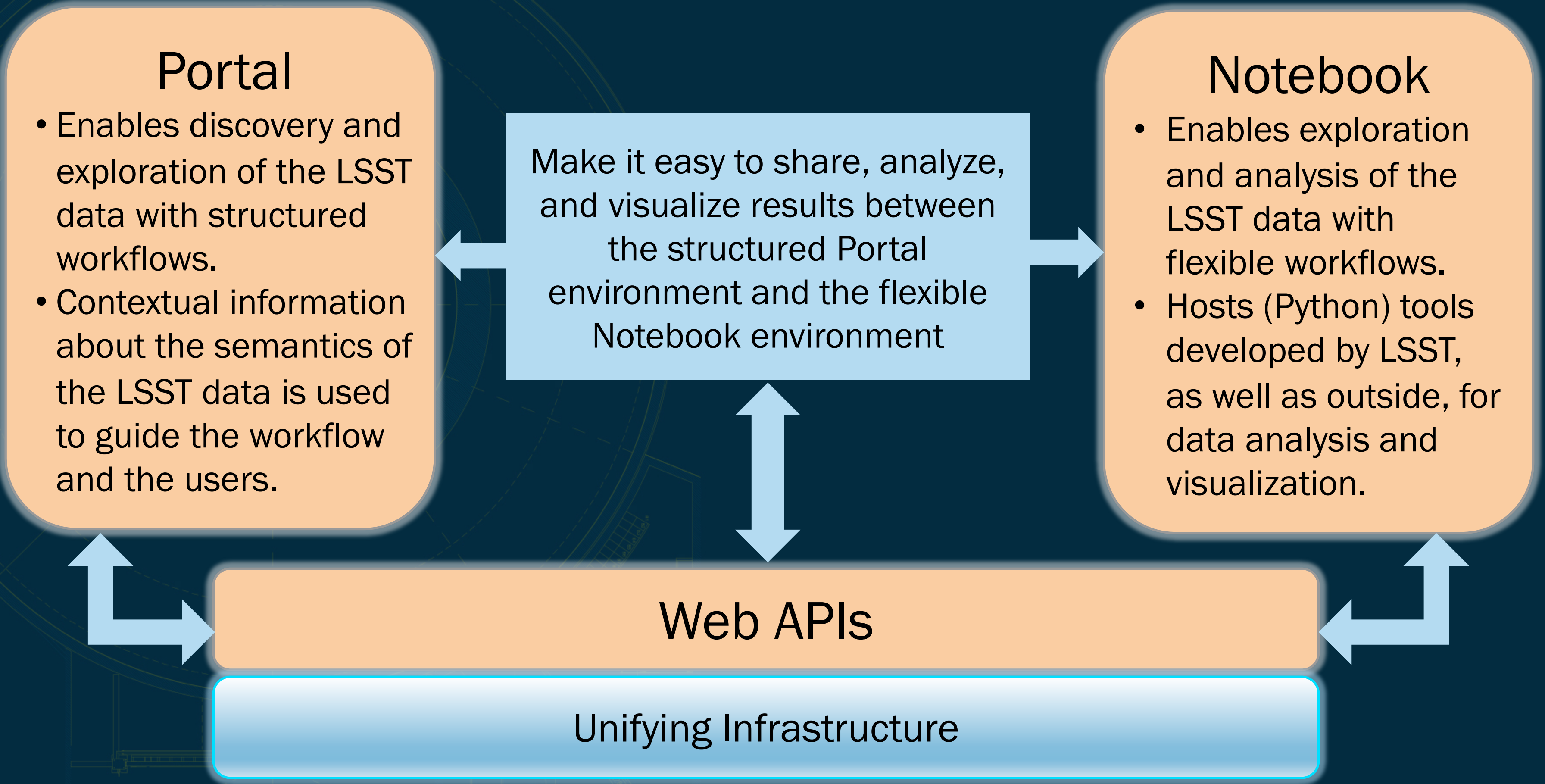
LSST Data Types

- Raw and calibrated single-epoch single-filter images
- Coadded maps of the full survey footprint, O(20k sq. deg.), in six bands, plus a multi-band optimal detection coadd, available as cutouts for any location, as well as HiPS maps
- Time-independent object characterization
- Per-epoch independent source detection, in addition to forced photometry

LSST Data Volumes

- 10-year survey of the southern sky in 6 filters: 20,000 sq. deg.
- 2.75 million “visits”.
- For DR11:
  - 37 billion objects
  - 7 trillion sources
  - 30 trillion forced sources
  - 15 PB catalog

Conceptual Architecture of the Science Platform



Portal

- Enables discovery and exploration of the LSST data with structured workflows.
- Contextual information about the semantics of the LSST data is used to guide the workflow and the users.

Make it easy to share, analyze, and visualize results between the structured Portal environment and the flexible Notebook environment

Notebook

- Enables exploration and analysis of the LSST data with flexible workflows.
- Hosts (Python) tools developed by LSST, as well as outside, for data analysis and visualization.

The Science Platform Vision

The Science Platform concept rests on the provision of three "*Aspects*" of access to the LSST data, storage, and computing: Portal, Notebook, and Web APIs

- A Portal with access to all LSST data products, visualization and exploration tools, and structured workflows that guide users to discover and understand the data and their semantic connections.
- A Notebook-style flexible, interactive Python computational environment (JupyterHub/Lab) with access to the LSST Data Products, computing, and storage, and with pre-provisioned access to installed versions of the LSST Python/C++ software stack.
- A set of Web APIs underpinning these user environments, both internally and externally available, providing access to the Data Products, primarily based on IVOA standards.

All these rest on a computational infrastructure; a flexible LSST and user storage infrastructure; and a database system supporting both the Project data products and users' own databases. This enables the users to create workflows that cross between the Portal and Notebook environments and the use of the APIs as their needs dictate.

Technology Overview

Portal Aspect – a Web application

- Based on an evolution of the open-source Firefly toolkit that is used in the IRSA archives
  - Native astronomical image visualization with data overlays
  - Tabular data viewing, 2D & 3D plotting, filtering, sorting, synthetic columns, with brushing and linking
  - All-sky visualization based on HiPS and MOC
- Data exploration workflows illuminating the connections among the LSST data products

Notebook – a Web application

- Based on JupyterLab and JupyterHub (for login, session control)
- Container-based deployment of the LSST software stack
- Python processes execute close to the data at the LSST Data Access Centers

Data Access APIs

- Preferring IVOA protocols (e.g., TAP, SODA, VOSpace) wherever feasible
- Metadata services for data discovery (principally TAP_SCHEMA and ObsTAP/SIAv2)
- May need some extensions for the richness of the data model or performance, still evaluating

LSST Data Dissemination

Data advertised through the IVOA Registry solutions of the day (noting that we have not yet turned to analyzing how adequate existing standards are for our needs)

- the LSP Portal Aspect providing access to the data via a Web GUI
- interactive access from Python in the Notebook Aspect via a combination of

LSST-specific APIs (i.e., the Butler, mainly) and community APIs for access to the IVOA data services (i.e., Astroquery and PyVO or what they’ve evolved into by then)

Investigating Dask for code-to-the-data processing.

Catalog Cross Correlations

We will provide selected key datasets on the LSP in addition to the LSST data; Gaia DRn is the only confirmed one, but we will evaluate others for their utility to commissioning and calibration.

Beyond that no decisions have been made.

We are aware that there is great interest in the community in combining data from LSST, Euclid, and WFIRST, and there is substantial overlap with the teams working on preparing for that.

The Web API Aspect of the Science Platform

Encompasses all externally visible APIs provided for data access

- "Zero-install" access to LSST data for remote users

Services provided

- IVOA standard data retrieval protocols (TAP+ADQL, SIAv2, SODA, etc.)
- Structural metadata provided by standard means (e.g., TAP_SCHEMA)
    - Exploring extensions to support contextual documentation provision
    - Metadata provision integrated into the LSST data production design (see "Science Data Model")
- Still evaluating whether any custom APIs or data formats will be needed
    - Custom APIs will be designed with usability in a DataLink context in mind
- "Workspace" APIs for user data space access and management (VOSpace and WebDAV)
- IVOA Registry standards will be followed to support data discoverability

Community tool support

- We support the use of community client applications (e.g., TOPCAT) and libraries (e.g., PyVO)
- Working with the community to adopt standards for authorization
- Evaluating providing an "astroquery.lsst" for user convenience

Current status and plans for VO Usage

Current development activity

- TAP 1.1 for catalog and native image metadata access (working with CADC code base)
  - Implementing ADQL spatial functions over Qserv shared-nothing, spatially sharded, parallel database
- VOTable 1.3 support in TAP server, in the LSST Portal (Firefly), and via Python (pyvo, astroquery)
- ObsCore (and CAOM2) mapping of image metadata; ObsCore support in Firefly
- SODA version of existing image cutout service

Planned activity

- SIA and ObsTAP services on top of ObsCore mapping
- VOSpace access to user file workspace, in addition to WebDAV
- HiPS map generation from LSST coadds; MOC generation for additional data subsets
  - Exploitation of HiPS Progenitor data, subject to further standardization, in Firefly
- Mapping between native (Apache Avro-based) transient alert model and VOEvent

Still in exploration

- Applications of DataLink
- Applications of VO-DML

Areas where we are interested in collaborating on standards updates, extensions, etc.

- Authenticated access to IVOA services such as TAP: developing standards to the point where client tools (e.g., TOPCAT, PyVO) work cross-project
- Improvements to transport of large tabular data results
- Improvements in the sophistication of client libraries (e.g., pyvo) that exploit data model metadata