Science User Interface

David Ciardi & Xiuqin Wu

The SUI Team

13 November 2014

SUI Team Members

Xiuqin Wu	Technical Lead
David Ciardi	Science Lead
John Rector	System Engineer
Trey Roby	System Architect
Tatiana Goldina	Senior Developer
Lijun Zhang	Senior Developer
Loi Ly	Senior Developer
Steve Groom	System Engineer
Jason Surace	Scientist
Gregory Dubois-Felsman	System Engineer

Overall Philosophy

- The SUI is the entry point of the community to the LSST data – for expert and novice users
- The SUI needs to be simple enough to engage the novice and flexible enough to meet the needs of the power-user
- The SUI is more than just the list of requirements – it is a system and a library to mate the data with the user

Overall Philosophy

- The SUI is a toolbox for us and the community
- Data access and manipulation functions all accessible via an API – the GUI will utilize the APIs
- GUI-specific components will be a library of components
- Enable creativity and flexibility

Overall Philosophy

- No way we can anticipate all the needs/wants of the community
- IPAC will build a portal that fulfills the needs of the general user (e.g., searching, image visualization, table manipulation, plotting, workspace etc.)
- Components will be usable by others to use and to build tools that meet their own special needs

What we are doing

- Understanding the requirements and the data products
- Building an early prototype based upon existing IPAC interface technology
  - Exercise APIs and Qserve to get to the simulation data
  - Identify areas which are unclear or undefined
  - Build a strong connection with the SUI and the DB (and eventually the file system)
  - Gets the engineers quickly involved and up to speed

Where we are going

- Assess the requirements and connect them to each design element
  - Identify what is missing, incomplete, or unclear
- Develop the more detailed SUI long-term schedule
  - Start with the LDM240
  - Create a set of completion milestones leading to the final product
  - This needs to blend with the overall DM schedule
- Define a clear path to a “preliminary” design review in the next year

SUI: Entry Point for LSST Data

- Provide users easy access to LSST data
- Enable users to do as much data discovery as possible
  - Searches
- Facilitate data analysis by providing tools needed
  - Visualizations
- Manage work flow and data collections for users
  - Workspace
- Supply software building blocks so others can build their own UI

Working On …

- Open source the IPAC Firefly package
- Configure local development environment
- Setup Qserv and testing database
- Talk with database team, design the APIs
- Understand the need for visualization in other parts of LSST, inside or outside DM
- Get up to speed on LSST system and lingos
- Get up to speed with Firefly for non-firefly developers
- Prototype application to access LSST data
- Study Python's role in SUI
- Study future web technologies to enhance and extend Firefly
  - WEB GL, D3 plotting, Angular JS, React, Backbone JS, Ember

SUI Key Technologies

- Tomcat server
- GWT
- Server side Java
- Java Script
- Python (in the study process)

Some Detailed Discussion Points

- SUI in Github
- Key technologies for SUI
  - GWT, server side Java,
- User identification, authentication, privilege
  - relationship with the security policy in development, led by NCSA
  - SSO system?
  - The same SSO system to support internal data access for QA?
  - Support anonymous user access?
- User workspace/environment
  - supply VM or a container
  - a separate place for SUI related data storage
  - a separate place for Qserv related data storage
  - a separate place for user's own data storage and/or software
  - Public vs private – related to L3 data
- L3 data
  - Released for public or collaborator access
- Image data model
  - meta data collection optimized for search/query
  - Image data optimized for data transfer
- Manage resource requests
  - VM/container, space/CPU

Qserv APIs

- User login through SSO?
- Meta data information about the table, including UID for VO table
    - data definition table, for each column in each data table
- Support for ADQL?
- Search types
    - by position
    - by general SQL
- Results returned and displayed
    - Default columns to be returned/displayed
- Web based API
- Search progress status report
    - each search should have a unique identifier
- Partial results return
- Cancel a search