Rubin Science Platform remaining work

Gregory Dubois-Felsmann
DM-SST vF2F 23 October 2023

Science Platform status

This presentation covers both:

- Services and Procedures
- User-facing features

Remaining work – data delivery

- We have not practiced the data flow of Prompt Products from AP to release through the RSP, either images or PPDB catalogs
- We have not verified, from a user perspective, that we have a working scheme for allowing users to understand the evolution of DiaObjects and the retrieval of their previous states
- We have not worked out the details of how the many image metadata tables and services will be populated and what their relationships to each other are
- We have not worked out how the “30 day cache” of temporarily-saved products will be implemented, including preserving information in the system that they did previously exist
- We haven’t resolved the problem that the need for compressed *AP* PVIs following the 30-day window was overlooked

Remaining work – databases

- A comment on Qserv/Postgres:
  - From a full-survey-operations perspective, Qserv doesn't come into play until DR1. Most likely, because we used Qserv for the DRP-like data products in DP0.1/DP0.2, we'll use Qserv for the DP1 and DP2 DRP-like catalog data products, but…
  - The majority of users' attention during Year 1 will be on the Postgres-based PPDB
- We don't have user databases
  - So we've had no practice with them in Data Previews yet
- For Qserv:
  - We don't have temporary-table upload (needed for efficient multi-object queries)
  - Beyond basic user databases, we also don't have large-user-table spatial sharding
- We don't have an Alert Archive and haven't determined what the interfaces to it will look like

Remaining work – API Aspect / data services

- We don’t have a query-history service
  - Underlies the model for working smoothly across RSP sessions and between Portal/Notebook
- We don’t have a service-level interface to user databases
- We don’t have the “user file workspace” (a/k/a VOSpace / WebDAV)
  - A basic WebDAV service was demonstrated a couple of years ago and usable from Portal
  - A replacement WebDAV service is well along in SQuaRE work
- We don’t have provenance-query services
  - “Tell me which single-epoch images went into this coadd”
  - “Tell me which calibrations went into this PVI”
- We don’t have the specialized-data-product visualization services
  - Relevant for the parts of our data products that are not in a community-standard form
  - E.g., PSF, background model

Remaining work – API Aspect / data services (2)

- We don’t have a multi-epoch bulk cutout service
  - We haven’t decided on a data format for the output (see also “cell-based coadds”)
- We don’t have a forced-photometry-on-demand service
- We don’t have data-product-recreation services
  - Lossless PVIs
  - Non-persisted coadds
- We don’t have a worked-out design for user access to coadds in the cell-based coadd era
  - How will users see coadds over larger-than-cell sky regions?
    - Science issue for nearby galaxies / extended objects
    - On-the-fly composition? “Concatenated inner region” data stored in addition to the cells?
    - Is a really, really good HiPS solution (for which more work would be needed) an alternative?

Stop, I’m getting depressed – is there any hope?

- We DO have a service architecture that will accommodate a lot of these services once they are implemented
- We DO have at least the start of a Python framework for service implementations

- But we don’t have a full solution for the management of a large workload of long-running service jobs

Remaining work – Notebook Aspect

- Scaling to the expected user load
  - NB: DP1 (late 2024) will be open to all data-rights holders
- Access to user batch computing
- Access to interactive parallel computing resources (“next-to-data analysis”)
- Polish to Python data access interfaces – examples:
  - Multiple Butlers (e.g., Prompt Products vs. DR/DP products)
  - Query history service access
  - Access to additional data services – see below
  - Portal-notebook connections

Remaining work – Portal Aspect

- Basic technical capabilities needed are nearly all in place, shifting focus to UX
- Many of the data services just described will be accessible from the Portal as-is once they are delivered
- A few remaining technical issues:
  - Better support for users having long-running asynchronous jobs (see “data services” above)
    - Exposure of results of async jobs having complex/multiple results (not just a table or an image)
  - Integration of query-history service into the main UX workflow
  - Support for file formats for multi-epoch cutouts and cell-based coadds (the same one?)
  - Support for non-expert users creating JOINs
- Need access to detailed (“deep-linked”) documentation
  - Need to agree on an interface for this that works in the Rubin documentation model

Remaining work – Portal Aspect (UX)

- “Make simple things simple” – stripped-down workflows to do common tasks
  - Simple dialogs for, e.g., “search coadds by position”, “search PVIs by time” (e.g., “last night”)
  - Access to full TAP-query power
- Favor progressive exposure of all the Portal data-analysis capabilities
- Clarification of the organization of the Portal’s top-level UI components
  - Resolve the “blue button problem”
- Highlight access to query history (both per-session and per-user) and status
  - Including long-running jobs beyond just “queries”
- Refresh of the overall visual appearance
  - Use of color
  - Consistency of dialogs
  - Icon style

Remaining work – data model curation

- Many features of the RSP, especially in the API and Portal Aspects, are metadata-driven via the TAP service and/or through related microservices
  - The source information for this metadata lives primarily in these places:
    - The Felis schema for the catalog and image metadata tables in sdm_schemas/yml
    - The DataLink annotations in sdm_schemas/datalink
- There's remaining work to bring the annotations of all tables up to a common standard and fully exploit the system capabilities
  - Jeremy McCormick of SLAC has recently joined the team to work on "Data Engineering" tooling and procedures
  - This should allow greatly accelerating work in this area
- As new services are deployed, hooks to them need to be added to the metadata