Quick SUIT (RSP Portal Aspect) update – November 2020

- 0.3 FTE for maintenance – bug fixes, release builds, deployment support, JupyterLab migrations
- Rubin-funded new development remains frozen, except…

- We are asking Brian Van Klaveren to work on adding image-metadata-specific query-building capabilities to the existing Portal (Firefly) TAP query screens
  - Intended for use on ObsTAP (table "ivoa.ObsCore") services; usable on *any* ObsCore-format tables

  - As-deployed Firefly *already* provides GUI-driven single-table and generic ADQL-entry queries, and
  - Already provides an image-browsing environment based on ObsCore tables

  - Brian will add UI elements for constructing queries against key elements of the ObsCore data model: spatial coverage (e.g., STC polygons), data product types, wavelength ranges, date/time ranges, calibration level, collections, dataset types (will work with Butler types)

  - Status: started in October, startup project (adding SQL syntax highlighting to ADQL-entry screen) completed, work on query components beginning this week

- Replaces the hard-coded non-IVOA-DAX image-search capabilities of the "PDAC"-era Portal
- Not otherwise on the near-term IRSA agenda because IRSA's legacy non-IVOA image metadata services have significant functionality not in the IVOA image metadata service standards

Loose ends

- Unfinished work in the Python API, especially around UI-driven callbacks to user code
- No formal support for completing this
  - gpdf would like to see if it’s possible to do the basic work, with a little maintenance-funded work if needed on the Firefly back end

Firefly development released in 2020

- Some relevant highlights (detailed release notes available on GitHub):
  - Substantial enhancements to interface for browsing complex VOTable and FITS file contents
  - Enhancements to image-metadata viewer for the “CADC model” in which image metadata services return DataLink “links service” URLs instead of direct links to the image data. This is the model we plan to use for the RSP.
  - Complete rethink of the UI for locking multiple image displays to each other
  - Replaced existing UI for interacting with a list of currently-displayed images with one that shares the same UX with other parts of Firefly, i.e., the same capabilities for selection, filtering, etc.
  - Schema browser in ADQL-entry screen improved to facilitate graphical selection of table and column names for use in queries
  - Introduction of Web API for invoking specific screens, either to directly execute them or present the user with a partially-filled-in query
    - Provides the first steps towards Rubin-specific customization of a “front page” for the Portal Aspect pointing people to key datasets and tables
  - Completely replaced online help system implementation to remove GWT code and facilitate layering of deployment-specific help over the help for the core components – much easier to customize help now
    - Not yet actually exploited for any RSP-specific customizations, due to freeze

Current non-Rubin development priorities

- Firefly is now actively shared by IRSA, NED, and the Exoplanet Archive
  - Used in many different ways
  - Development priorities set by a change control board on which Rubin is still represented
  - Rubin’s VO-first posture facilitates arguing for features

- Work in progress now is directly relevant to Rubin:
  - Basic level of use of DataLink service descriptors to find related data (demonstrated at IVOA this week)
    - Enables implementation of *many* remaining RSP requirements, driven by metadata served with TAP queries
  - Migration of image display to use client’s GPU (accelerates zoom, pan, contrast/stretch changes, flux readout)
  - To be released in January

- Key first-half-2020 priority: organization and display of spectral data
  - Not superficially relevant, but depends on robust implementation of DataLink capabilities, which will be immensely useful to the RSP Portal