Implementing SIAv2 Over Rubin Observatory's Data Butler

Tim Jenness,[1] Stelios Voutsinas,[1] Gregory P. Dubois-Felsmann,[2] and Andrei Salnikov[3]

[1]*Vera C. Rubin Observatory Project Office, 950 N. Cherry Ave., Tucson, AZ 85719, USA*

[2]*Caltech/IPAC, California Institute of Technology, MS 100-22, Pasadena, CA 91125-2200, USA*

[3]*SLAC National Accelerator Laboratory, 2575 Sand Hill Rd., Menlo Park, CA 94025, USA*

Abstract. The IVOA Simple Image Access version 2 protocol defines an easy way to provide community access to a collection of data. At the Vera C. Rubin Observatory we currently enable ObsTAP access to our data holdings via an ObsCore export or view of our Data Butler repositories. This approach does come with some deployment constraints, such as requiring pgsphere and compatibility with our CADC TAP implementation, so recently we decided to see whether we could instead provide an SIAv2 service that talks directly to our Data Butler. Here we describe our motivation, implementation strategies, and current deployment status, as well as discussing some metadata mismatches between the Butler data models and SIAv2.

1. Motivation

The Rubin Data Butler (Jenness et al. 2022) consists of a metadata registry and a file datastore. The registry contains sufficient information to construct ObsCore records. Previously it was possible to provide an ObsCore table using two methods: export the records as CSV or Parquet files and load them into a static database, or provide live syncing to an ObsCore table using hooks in the registry backend (Salnikov 2022).

Both these approaches have been implemented by us and are the only way we had to provide an ObsTAP service (Louys et al. 2017). However, there are downsides. Whilst static export works fine for formal data releases where it can be integrated into our high performance Qserv database (Mueller et al. 2023), it is not suitable for evolving datasets such as those from the nightly prompt products. "Live" ObsCore does work but requires that the deployment has pgsphere available. Also, if the configuration changes (something that can happen a lot during early operations) the entire table needs to be rebuilt.

These two issues suggested that there would be a benefit from providing a simpler, yet standardized, query layer directly on top of the Butler. The IVOA's Simple Image Access version 2 (SIAv2; Dowler et al. 2015) was the obvious choice for this. Interfacing directly to the Butler provides much more flexibility since configuration

changes can be picked up with a simple redeployment of the service and it can work immediately with any Butler repository.

2. Implementation

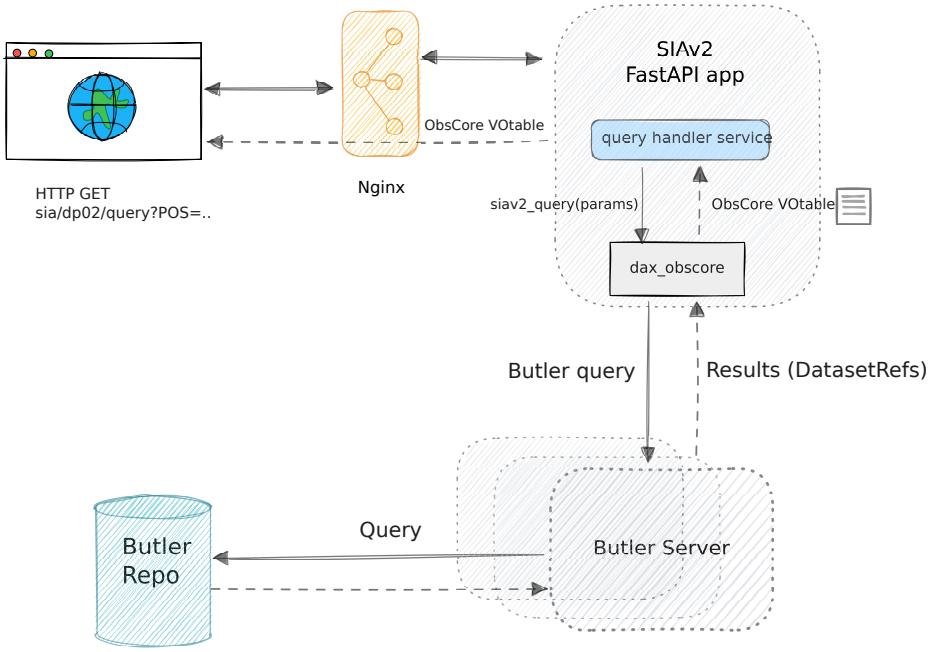

Figure 1. Architecture diagram showing how the SIAv2 service interfaces with the Butler.

We used a layered approach to implementation as shown in Fig. 1. The service itself (Voutsinas & Jenness 2024) is written in Python and FastAPI using Rubin's standard internal development platform Phalanx.[1] This provides us with our standard authentication layer and deployment capabilities.

The service takes the raw SIAv2 parameters and passes them, along with the SIAv2 configuration specific for the attached Butler repository, to the `dax_obscore`[2] library which parses the parameters, converts them to Butler queries, runs the queries, and returns the standardized results as an Astropy VOTable that can be packaged up by the service and returned. The resulting table schema is defined by a Felis data model for consistency (McCormick et al. 2025). The service works transparently with both the original "direct" Butler and the new client/server remote Butler (Jenness et al. 2024).

Separating the service layer from the SIAv2 query handling allows us to test the SIA-to-Butler queries much more easily. Additionally, the `dax_obscore` package,

[1] `https://phalanx.lsst.io`

[2] `https://github.com/lsst-dm/dax_obscore`

which can be installed from PyPI, provides a command-line interface to make it easier for people to learn the SIAv2 parameter system and experiment with queries locally.

3. Current Status

A service has been deployed on the Rubin Science Platform and is ready to be used with commissioning data. The `dax_obscore` package supports the following SIAv2 query parameters at this time: `MAXREC`, `INSTRUMENT`, `POS`, `TIME`, `BAND`, `EXPTIME`, `CALIB`.

Butler itself has native support for region and time queries. Support for `ID`, `TARGET`, `FACILITY`, and `COLLECTION` are coming soon. For the `FACILITY` keyword we intend to match the AAS Facility names of "Rubin:Simonyi" for the Simonyi Survey Telescope and "Rubin:1.2m" for the Rubin Auxiliary Telescope. The Butler does not have a concept of facility but the correct value will be derived from the instrument name. Butler does have a concept of target but in general the value of the field is not of interest since it will usually refer to a field name generated by the survey scheduler.

4. Model Mismatches

The Butler data model as used by Rubin Observatory does result in some implementation difficulties.

4.1. Instrument for coadds

Currently, as defined by the Rubin Science Pipelines team, the Butler registry does not have an instrument associated with co-adds. The instrument is recorded in the datasets and in the collection names but when the pipelines were initially defined it was deemed to be unimportant to include the instrument information in the dataset type definition itself. This causes difficulties in SIAv2 in a Butler repository where LATISS and LSSTCam data are both available since there is no way to know from the query system which is which. Currently it is deemed to be too disruptive to modify the pipeline connections to propagate the instrument into the co-adds Butler dataset definition. Once we implement full provenance tracking it should be possible to determine the instrument using the provenance to determine the original datasets that went into the co-add.

4.2. Exposure time for co-adds

Individual observations do include exposure times in the Butler registry metadata but this information is not available to co-adds. This is because the median exposure time for a co-add is a derived quantity that is not know when the Butler coordinate space is defined. Storing derived metadata, which is similar to calculating the seeing and then allowing a query based on seeing, is on the future development roadmap.

4.3. Observing dates for co-adds

Butler does know the observing dates of individual observations but for a co-add this information is lost. In the future when a full Butler provenance system is implemented it may be possible to derive the date range for co-adds, but it is not possible at this time.

4.4. Dataset Types

Butler makes extensive use of what we call "dataset types" which define each product type in a pipeline. Examples can be `visit_image`, `difference_image`, and `deep_coadd`. Currently there is no standardized way in SIAv2 for a query to specify these. We are looking at adding an extension, potentially using a `DPSUBTYPE` query parameter that will map directly to the Butler dataset type, possibly with an 'lsst' prefix.

5. Conclusions

Implementing SIAv2 over the Data Butler was a relatively simple process. Separating the development into a standalone SIAv2 parameters handler in `dax_obscore` with a service that forwards parameters to this layer allowed for parallel development. An additional advantage is that there is now a command-line tool that can be used to perform SIAv2-style queries directly on any Butler repository. Some data model mismatches are causing difficulties but we hope to fix most of those issues with the additional support for querying on derived metadata and provenance.

Acknowledgments. This material or work is supported in part by the National Science Foundation through Cooperative Agreement AST-1258333 and Cooperative Support Agreement AST1836783 managed by the Association of Universities for Research in Astronomy (AURA), and the Department of Energy under Contract No. DE-AC02-76SF00515 with the SLAC National Accelerator Laboratory managed by Stanford University.

References

Dowler, P., Bonnarel, F., & Tody, D. 2015, IVOA Simple Image Access Version 2.0, IVOA Recommendation 23 December 2015. URL `https://doi.org/10.5479/ADS/bib/2015ivoa.spec.1223D`

Jenness, T., Bosch, J. F., Salnikov, A., Lust, N. B., Pease, N. M., Gower, M., Kowalik, M., Dubois-Felsmann, G. P., Mueller, F., & Schellart, P. 2022, in Software and Cyberinfrastructure for Astronomy VII, vol. 12189 of Proc. SPIE, 1218911. `arXiv:2206.14941`, URL `https://doi.org/10.1117/12.2629569`

Jenness, T., Irving, D. H., Bosch, J. F., Salnikov, A., Lust, N. B., & Allbery, R. 2024, in Software and Cyberinfrastructure for Astronomy VIII, edited by J. Ibsen, & G. Chiozzi, vol. 13101 of Proc. SPIE, 131013G. URL `https://doi.org/10.1117/12.3019130`

Louys, M., Tody, D., Dowler, P., Durand, D., Michel, L., Bonnarel, F. a., Micol, A., & IVOA DataModel Working Group 2017, Observation Data Model Core Components, its Implementation in the Table Access Protocol Version 1.1, IVOA Recommendation 09 May 2017. URL `https://doi.org/10.5479/ADS/bib/2017ivoa.spec.0509L`

McCormick, J., Dubois-Felsmann, G. P., Salnikov, A., van Klaveren, B., & Jenness, T. 2025, in ADASS XXXIV, edited by A. DeMarco, & J. Said (San Francisco: ASP), vol. TBD of ASP Conf. Ser., 999 TBD

Mueller, F., et al. 2023, in ADASS XXXII, edited by S. Gaudet, S. Gwyn, P. Dowler, D. Bohlender, & A. Hincks (San Francisco: ASP), vol. TBD of ASP Conf. Ser., 999 TBD. URL `https://dmtn-243.lsst.io`

Salnikov, A. 2022, ObsCore as a View of Butler Registry Tables. Vera C. Rubin Observatory Data Management Technical Note DMTN-236, URL `https://dmtn-236.lsst.io/`

Voutsinas, S., & Jenness, T. 2024, SIAv2 over Butler FastAPI service. Vera C. Rubin Observatory SQuaRE Technical Note SQR-095, URL `https://sqr-095.lsst.io/`