An integrated platform for high performance data management and analysis at X-ray light sources

Nathan Cook*[a], Evan Carlin[a], Paul Moeller[a], Rob Nagler[a], Boaz Nash[a]
Maksim Rakitin[b], Andi Barbour,[b] Lutz Wiegart[b]

*ncook@radiasoft.net

a radiasoft
b Brookhaven National Laboratory

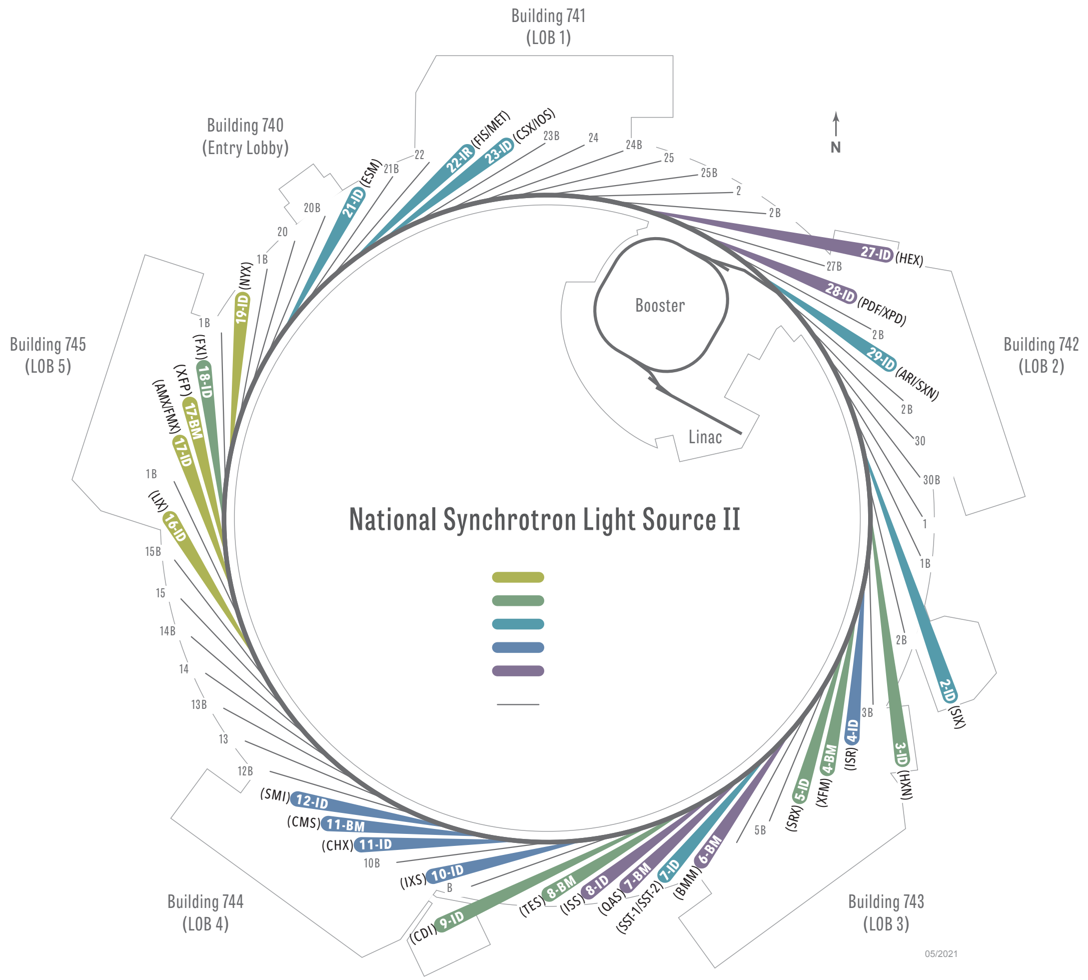
ICALEPCS International Conference on Accelerator and Large Experimental Physics Control Systems 2021

October 22, 2021.

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Science under Award Number DE-SC0021553

Light source user facilities are technology drivers

- X-ray light sources enable multidisciplinary scientific breakthroughs
  - 30 facilities worldwide, more than 8,000 refereed publications and 6,000 protein structures per year
- National Synchrotron Light Source II (NSLS-II)
  - State-of-the-art "third generation" synchrotron light source
  - 28 active beamlines, with 60-70 anticipated at full capacity
  - Serves ~1700 users annually
- User support provided by Photon Sciences Division
  - Beamline Science Programs
    - Instrument design and commissioning
    - Experimental planning, operations, and execution
    - *Directly engages with users*
  - Data Science & Systems Integration (DSSI)
    - Controls systems, computing resources, and software
    - *Supports beamline scientists and facility employees*
- User support is a significant investment!
  - Scientific staff spent 80% of their time on user support



User facility scientific workflows present unique challenges

- Successful experiments require cooperation between multiple parties with distinct expertise
  - End user – experimental lead, subject matter expert, and driver of scientific scope
    - Defines experimental scope. Provides samples for study. Customizes experimental procedure.
    - Works with beamline scientist to carry out experiment.
    - *May not be an expert in beamline controls system, software development, nor high performance computing.*
  - Beamline scientist – beamline and/or instrument expert and lead on operation and execution of experiment
    - Commissions and validates beamline for experimental operations.
    - Works with the end user to adapt beamline and analysis operations for their experiment.
    - Works with the computational scientist to provide software support for common measurement and analysis procedures.
    - *Has to juggle many different technical requests, requiring working knowledge of science, instruments, and software.*
  - Computational scientist – data acquisition and software expert – designs tools for data storage and analysis
    - Develops and implements data acquisition and analysis software. Maintains computational resources for all parties.
    - Works with the beamline scientist to deploy software at beamline.
    - May work with end user to coordinate facility-wide access to computational resources.
    - *May not be an expert in subject matter, but requires working knowledge of accelerator and/or beamline components.*

- *Our work seeks to support all three parties through improving the connectivity of software components*

Analysis Pipelines are diverse and specialized

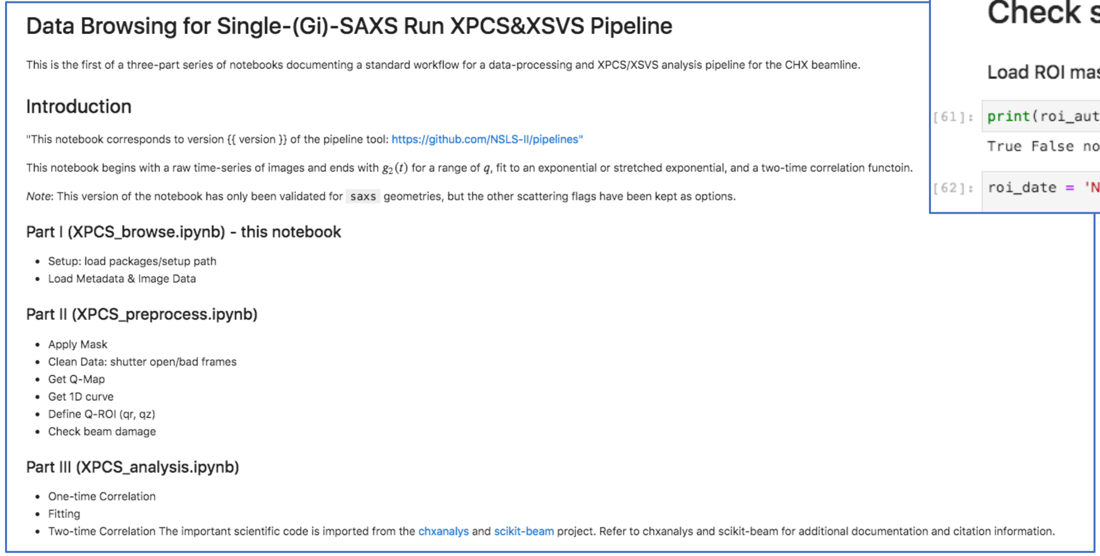
- Significant variations in dynamic range across similar beamlines, and even within a single beamline

Images removed in preparation for publication

- Existing workflows leverage custom libraries for online data processing and analysis
  - `PyCHX` (https://github.com/NSLS-II/pyCHX), `PyXRF` (https://github.com/NSLS-II/PyXRF), `scikit-beam` (https://github.com/scikit-beam)
  - No direct link to controls software (e.g. `bluesky`)
- Analysis pipelines should not compromise custom workflows

Beamline agnostic analysis requires comprehensive environments

- Encapsulate analysis within a self-contained, modular environment via Jupyter Notebooks
  - Python environment supports varied analysis and visualization tools
    - Markdown enables rich text documentation, organization, and formatting
    - Backend supports inline rendering of datasets, images, and analyses
  - Versioning and deployment can be supported via continuous integration
    - This workflow is commonly adopted for NSLS-II operation (https://github.com/NSLS-II/profile-collection-ci)
  - Notebook can be modified and run manually, or templated and automatically executed (e.g. via `Papermill`)
    - Automatically generate and export reduced datasets, figures, and reports



Data Browsing for Single-(Gi)-SAXS Run XPCS&XSVS Pipeline

This is the first of a three-part series of notebooks documenting a standard workflow for a data-processing and XPCS/XSVS analysis pipeline for the CHX beamline.

Introduction

"This notebook corresponds to version {{ version }} of the pipeline tool: https://github.com/NSLS-II/pipelines"

This notebook begins with a raw time-series of images and ends with $g_2(t)$ for a range of q, fit to an exponential or stretched exponential, and a two-time correlation functoin.

Note: This version of the notebook has only been validated for `saxs` geometries, but the other scattering flags have been kept as options.

Part I (XPCS_browse.ipynb) - this notebook

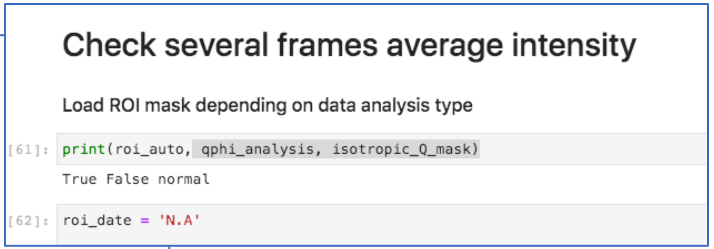
- Setup: load packages/setup path
- Load Metadata & Image Data

Part II (XPCS_preprocess.ipynb)

- Apply Mask
- Clean Data: shutter open/bad frames
- Get Q-Map
- Get 1D curve
- Define Q-ROI (qr, qz)
- Check beam damage

Part III (XPCS_analysis.ipynb)

- One-time Correlation
- Fitting
- Two-time Correlation The important scientific code is imported from the chxanalys and scikit-beam project. Refer to chxanalys and scikit-beam for additional documentation and citation information.



Check several frames average intensity

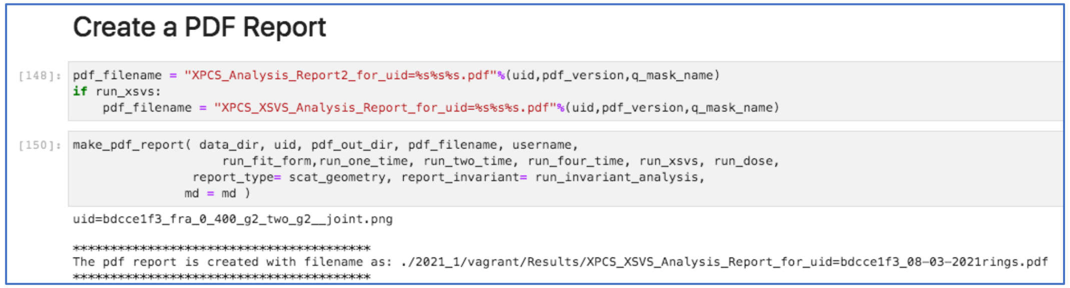
Load ROI mask depending on data analysis type

```
[61]: print(roi_auto, qphi_analysis, isotropic_Q_mask)
True False normal
[62]: roi_date = 'N.A'
```

Compress Data

- Generate a compressed data with filename
- Replace old mask with a new mask with removed hot pixels
- Do average image
- Do each image sum
- Find badframe_list for where image sum above bad_pixel_threshold
- Check shutter open frame to get good time series

```
[66]: photon_occ = len( np.where(avg_img)[0] ) / ( imgsa[0].size)
compress =  photon_occ < .4  #if the photon ocupation < 0.5, do compress
print ("The non-zeros photon occupation is %s."%( photon_occ))
print("Will " + 'Always ' + ['NOT', 'DO'][compress]  + " apply compress process.")
The non-zeros photon occupation is 0.047912584240105756.
Will Always DO apply compress process.
```



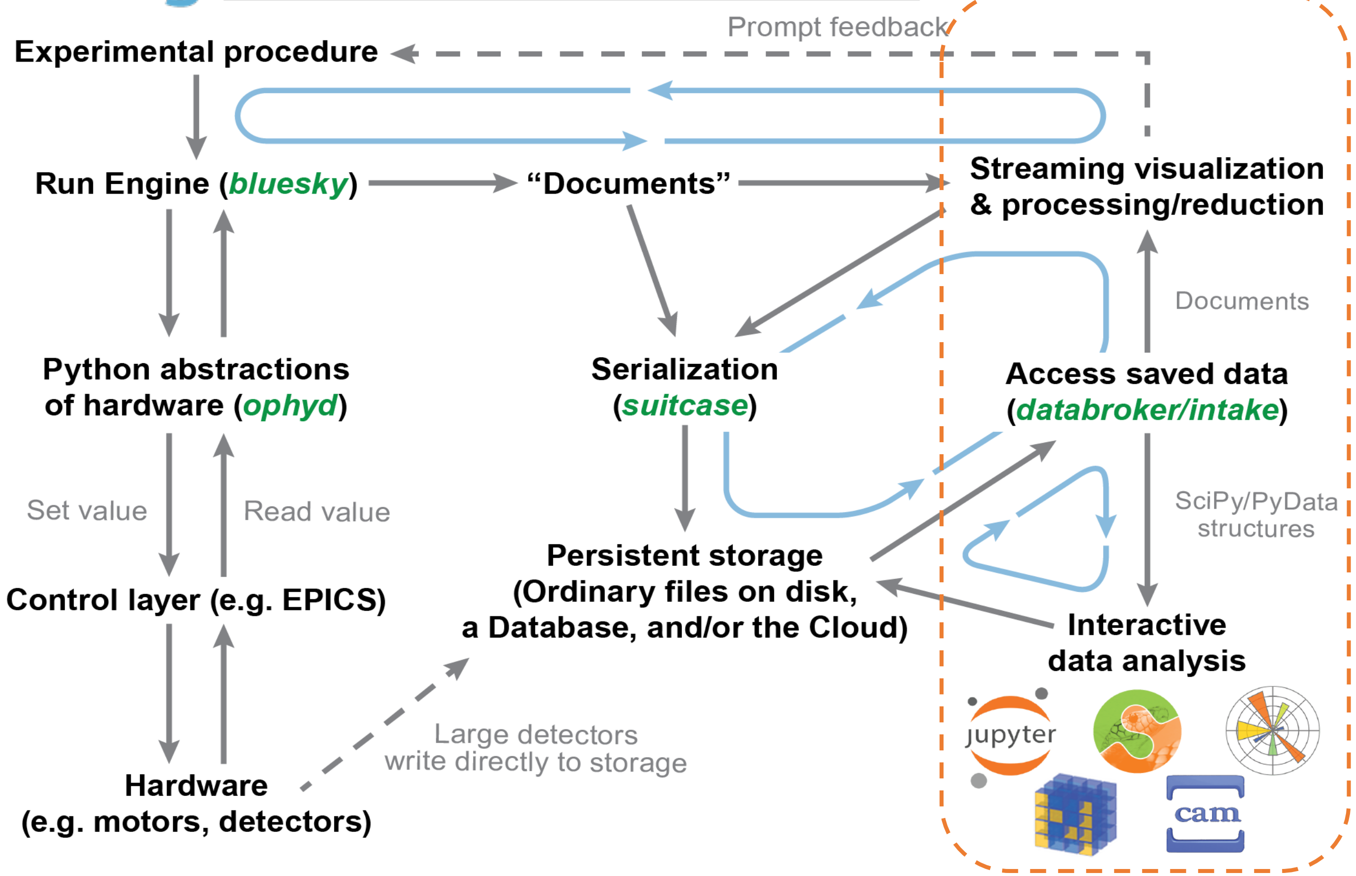
Create a PDF Report

```
[148]: pdf_filename = "XPCS_Analysis_Report2_for_uid=%s%s%s.pdf"%(uid,pdf_version,q_mask_name)
if run_xsvs:
    pdf_filename = "XPCS_XSVS_Analysis_Report_for_uid=%s%s%s.pdf"%(uid,pdf_version,q_mask_name)

[150]: make_pdf_report( data_dir, uid, pdf_out_dir, pdf_filename, username,
                    run_fit_form,run_one_time, run_two_time, run_four_time, run_xsvs, run_dose,
                report_type= scat_geometry, report_invariant= run_invariant_analysis,
               md = md )
uid=bdcce1f3_fra_0_400_g2_two_g2__joint.png

****************************************
The pdf report is created with filename as: ./2021_1/vagrant/Results/XPCS_XSVS_Analysis_Report_for_uid=bdcce1f3_08-03-2021rings.pdf
****************************************
```

Control and Data Collection Workflows are Sophisticated

- Experimental procedures are well defined via descriptive schemas and equipment protocols

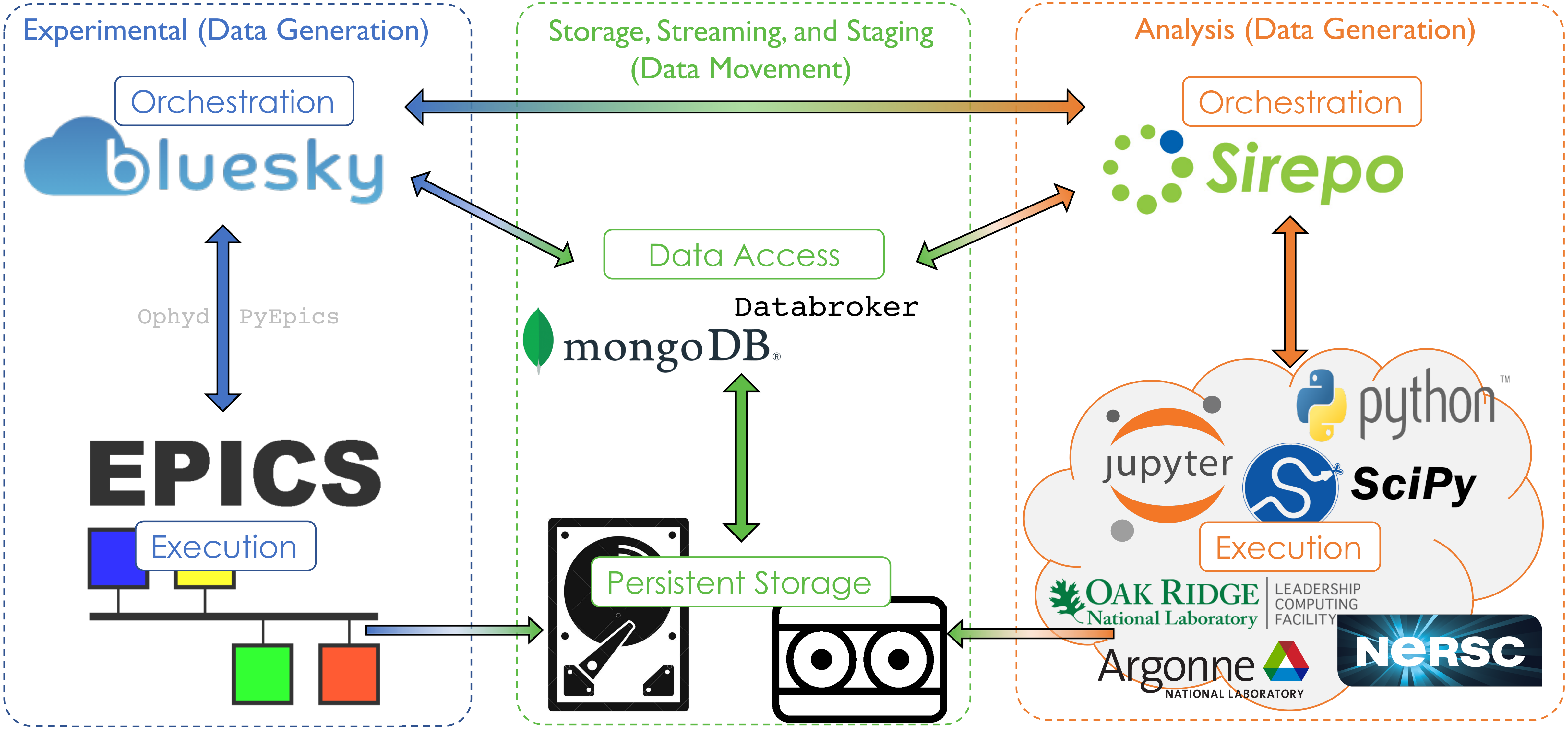
bluesky https://blueskyproject.io/bluesky/



- Analysis procedures may be de-coupled from this ecosystem
- Custom callbacks enable integration of specialized analysis tools and data management
- Coordinating the required tools, resources, and feedback systems remains a huge challenge!

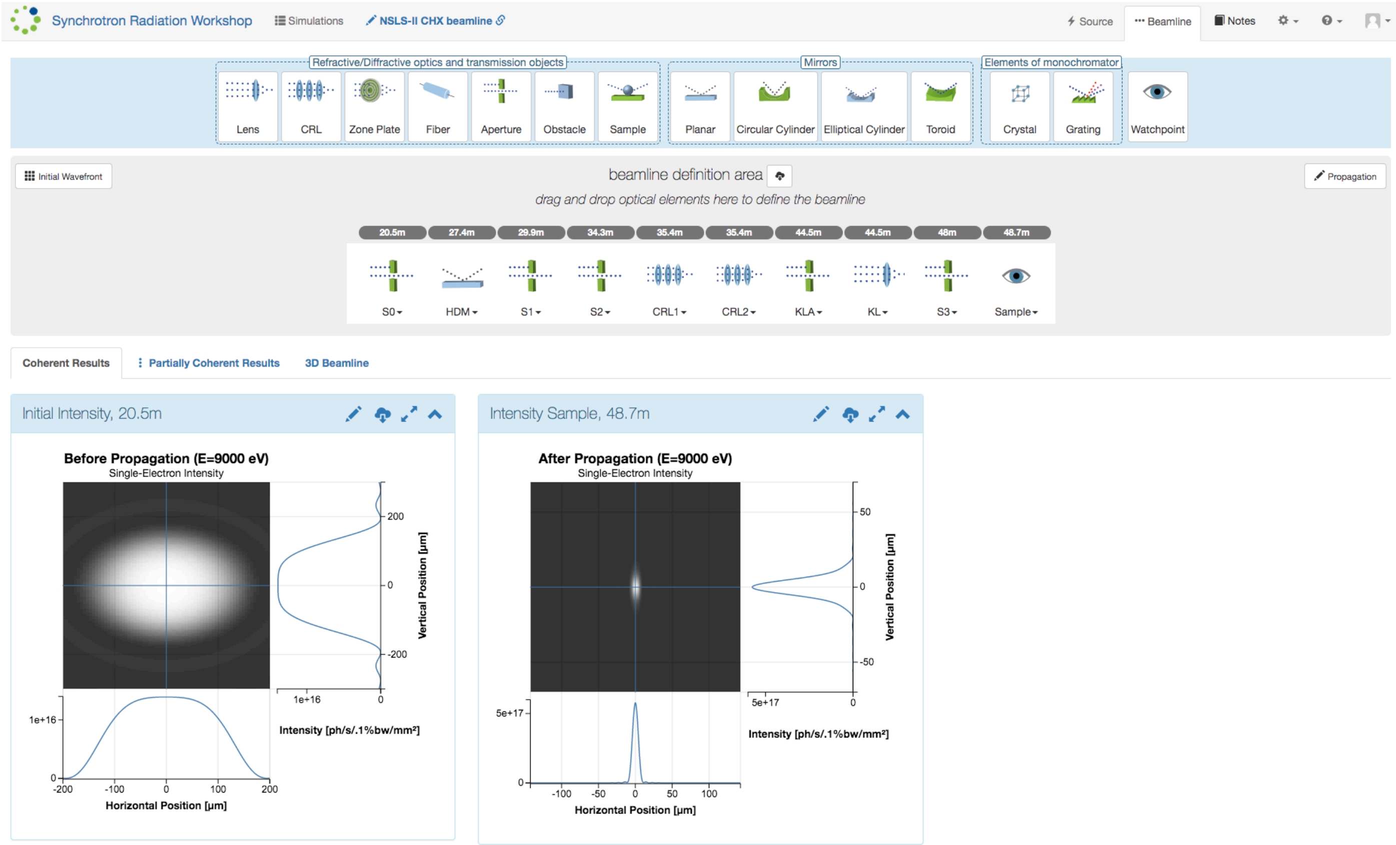
- Integration of disparate pipelines will enhance performance and streamline user experience

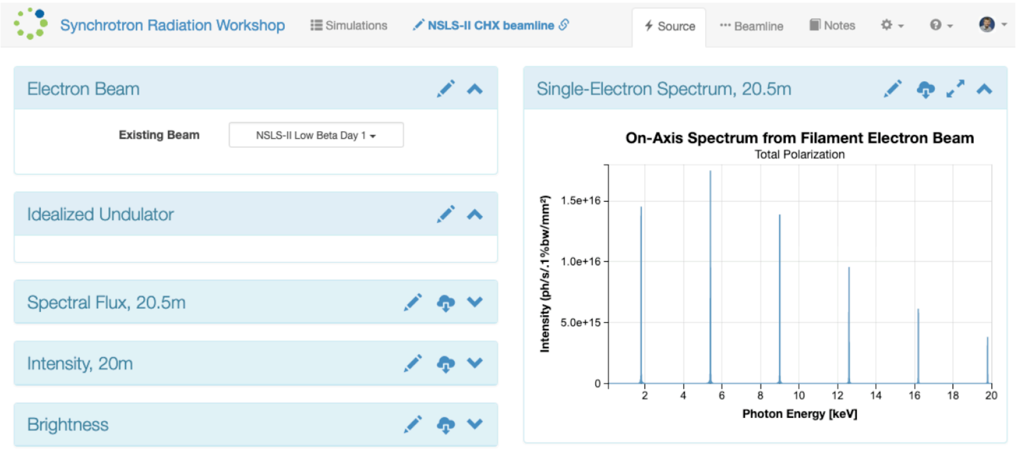
Integrating Experiment and Analysis Workflows with Sirepo



Sirepo supports beamline simulations and Bluesky integration

- Sirepo is a cloud-based platform for supporting accelerator codes, analysis tools, and controls libraries
- Sirepo provides an interactive interface to the Synchrotron Radiation Workshop (SRW) code
  - Native support for several NSLS-II beamlines including CHX and CSX models
- Sirepo simulations have been coupled to Bluesky* plans in support of beamline studies
  - SRW has been demonstrated to reproduce relevant experimental scans at the CHX beamline†





† O. Chubar et al. "Simulation of experiments with partially coherent x-rays using Synchrotron Radiation Workshop". In: Proc. SPIE. Vol. 10288. Aug. 2017

† L. Wiegart et al. "Towards the simulation of partially coherent x-ray scatteing experiments". AIP Conf. Proc. **2054**, 060079 (2019).

*M. S. Rakitin et al. "Introduction of the Sirepo-Bluesky interface and its application to the optimization problems". In: Proc. SPIE. Vol. 11493. Aug. 2020.

Characterizing a common analysis workflow

Stage → Pre-process → Analyze → Document → Post-process

1. Staging

Query from available UIDs and down-select from metadata

- `databroker` v2 API implements catalog-based retrievals from MongoDB
- Standardized queries permit standardized interface

2. Pre-process

Data cleaning and preparation

- Load, modify, compose masks
- Select and apply ROIs

Data reduction

- Time series sampling and compression for follow-on analysis

3. Analysis

Sequence of specialized real-time calculations

- Custom libraries leveraged for most analysis
- Templated pipelines used across many runs
- Resource bottleneck

4. Documentation

Compile figures, tag directories, and generate reports

- May use `Olog`, `databroker`, or other logboook software
- Consistent run-to-run, but will vary across beamlines

5. Post-Process

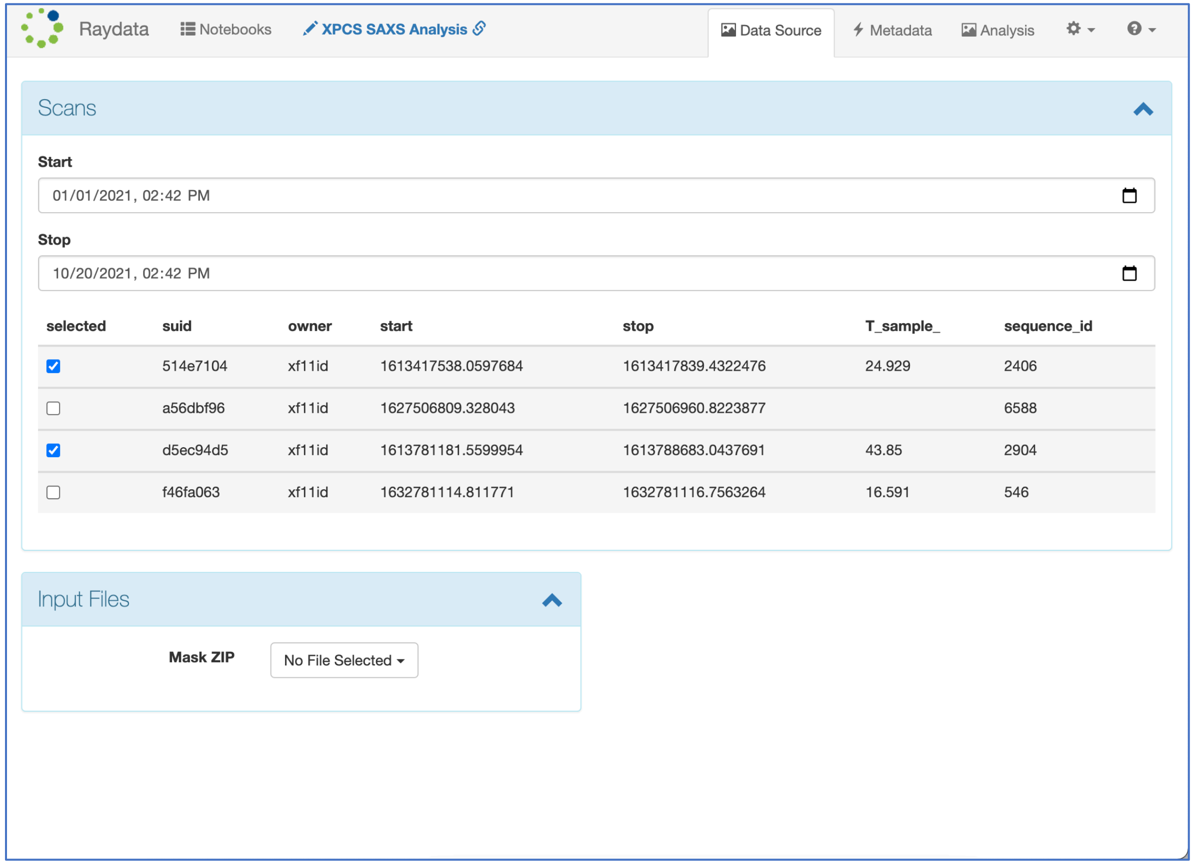
User-directed follow ups

- Long-term data access, and potentially larger computational resource needs
- More relaxed timeframe
- Largely independent of experimental operation

A Prototype Sirepo Interface for Real-Time Analysis (I)

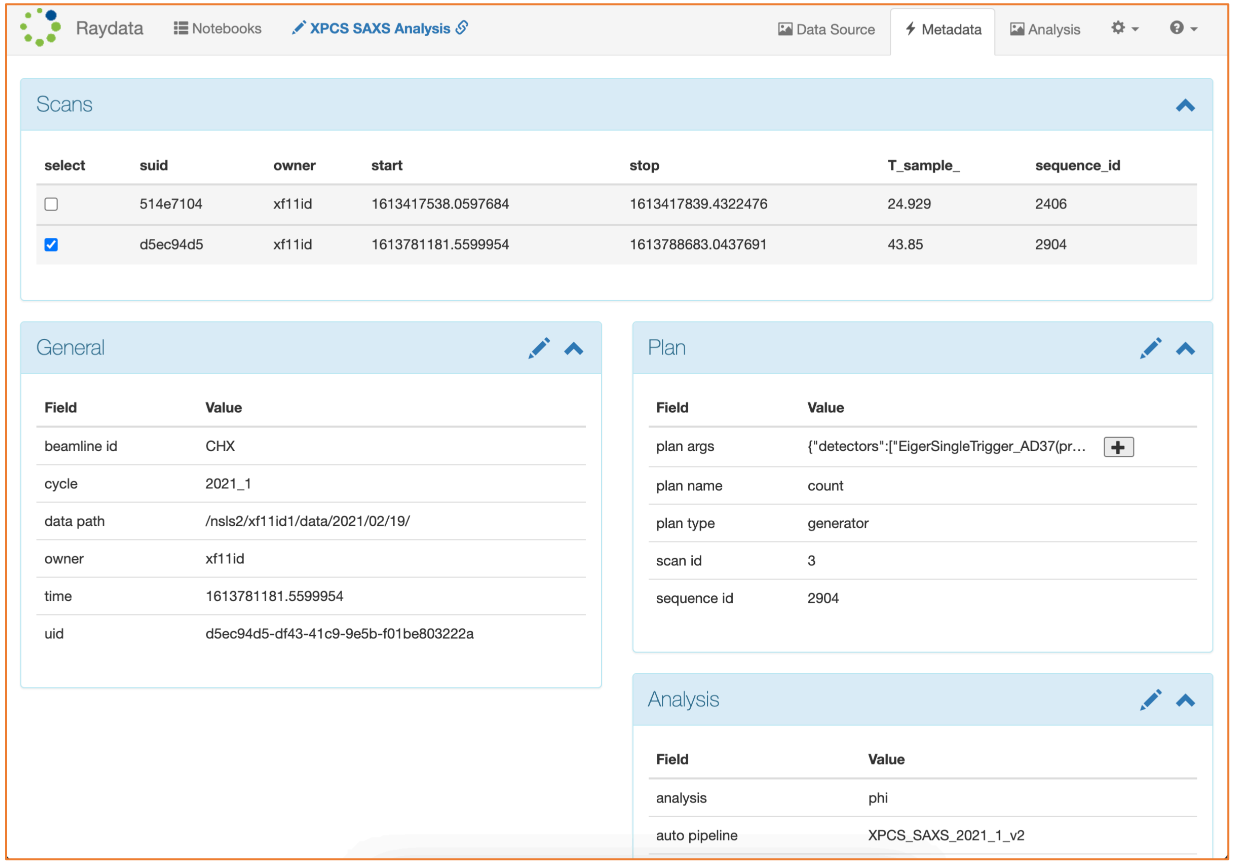
Selection of runs for inspection and analysis

- Select from available runs at the beamline using `databroker` catalog infrastructures
- Searchable and sortable by UID, date/time, and other descriptive metadata



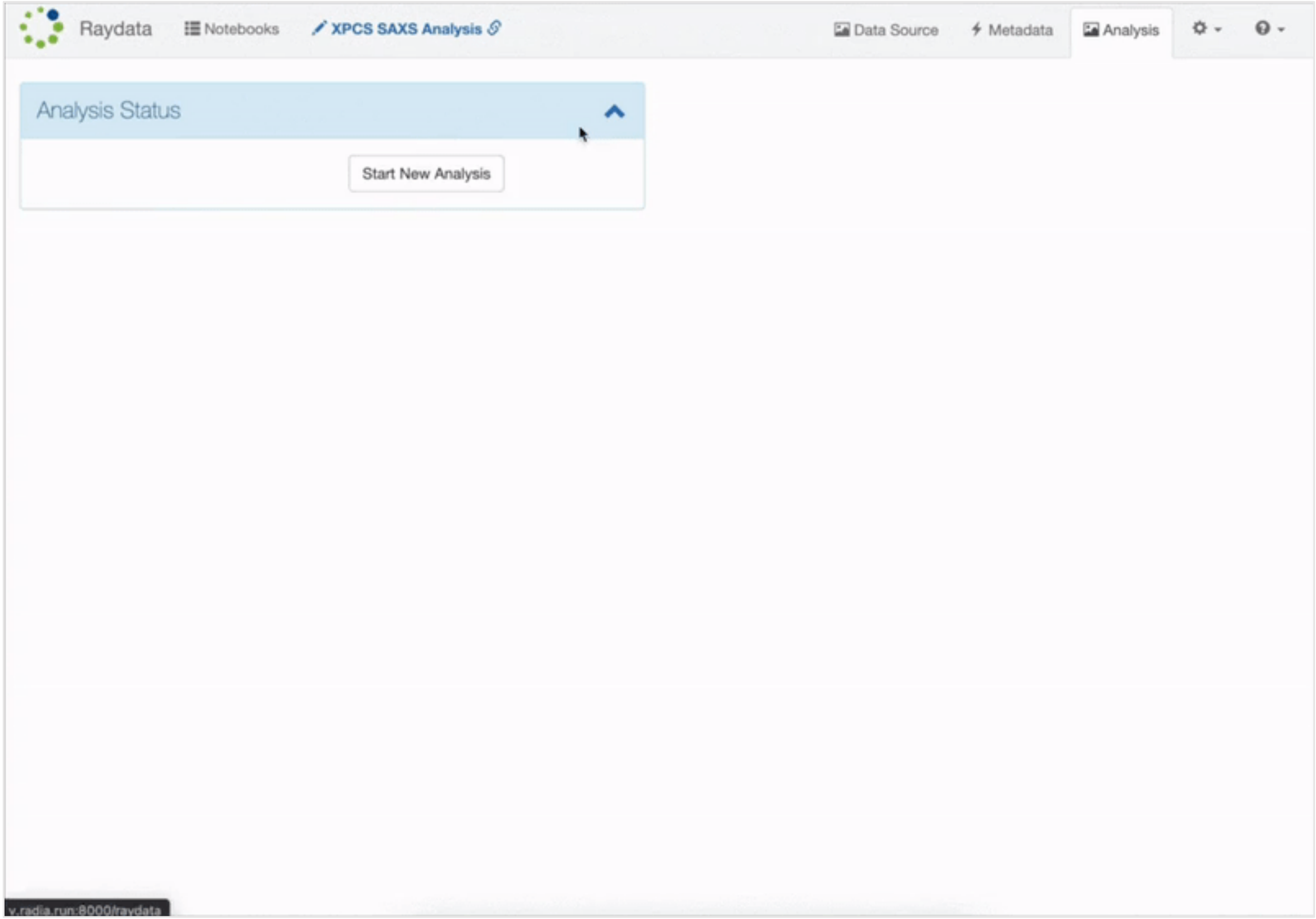
Inspection of metadata and analysis protocols

- Leverage catalog schemas to populate high level metadata for quickly browsing each run
- Preview relevant parameters prior to launching analysis



A Prototype Sirepo Interface for Real-Time Analysis (II)

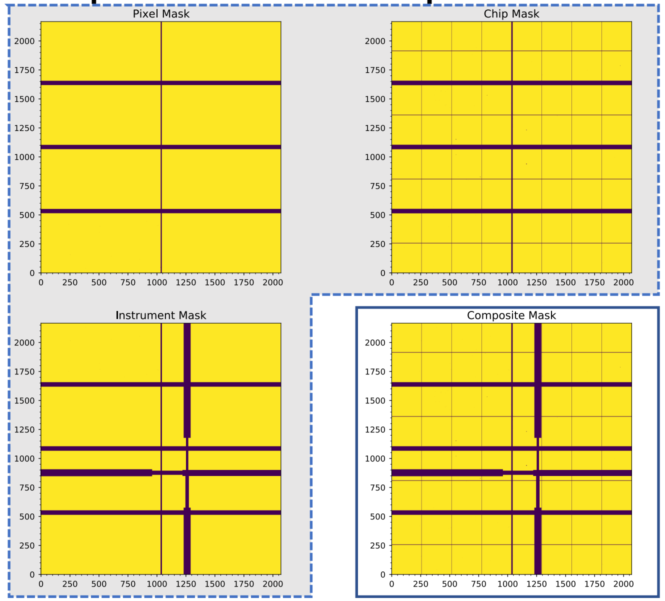
- Jupyter notebooks are deployed via pre-built Docker images
  - Easily reconfigurable for different dependencies and environments
- Active resource management
  - Run locally at the beamline, on a site cluster, or at NERSC
  - Queueing support in progress
    - Native first-in, first-out
    - Users can escalate priority
- Dynamic report generation
  - Figures provided in real time
  - Synthesized documents (PDF reports) produced as specified.



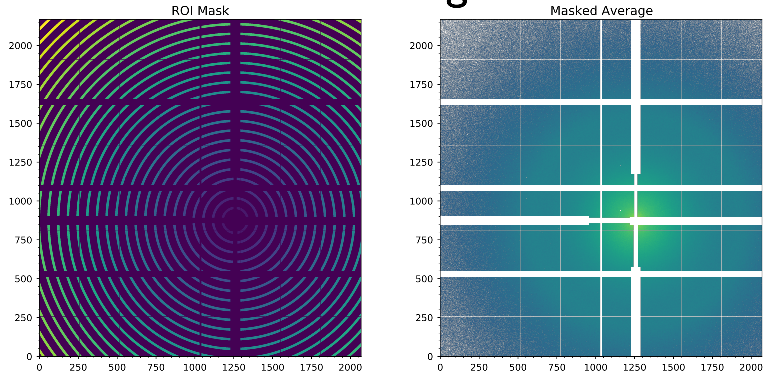
In-development mechanisms for real-time feedback

Adaptive Pre-Processing

- Upload, preview, and compose masks

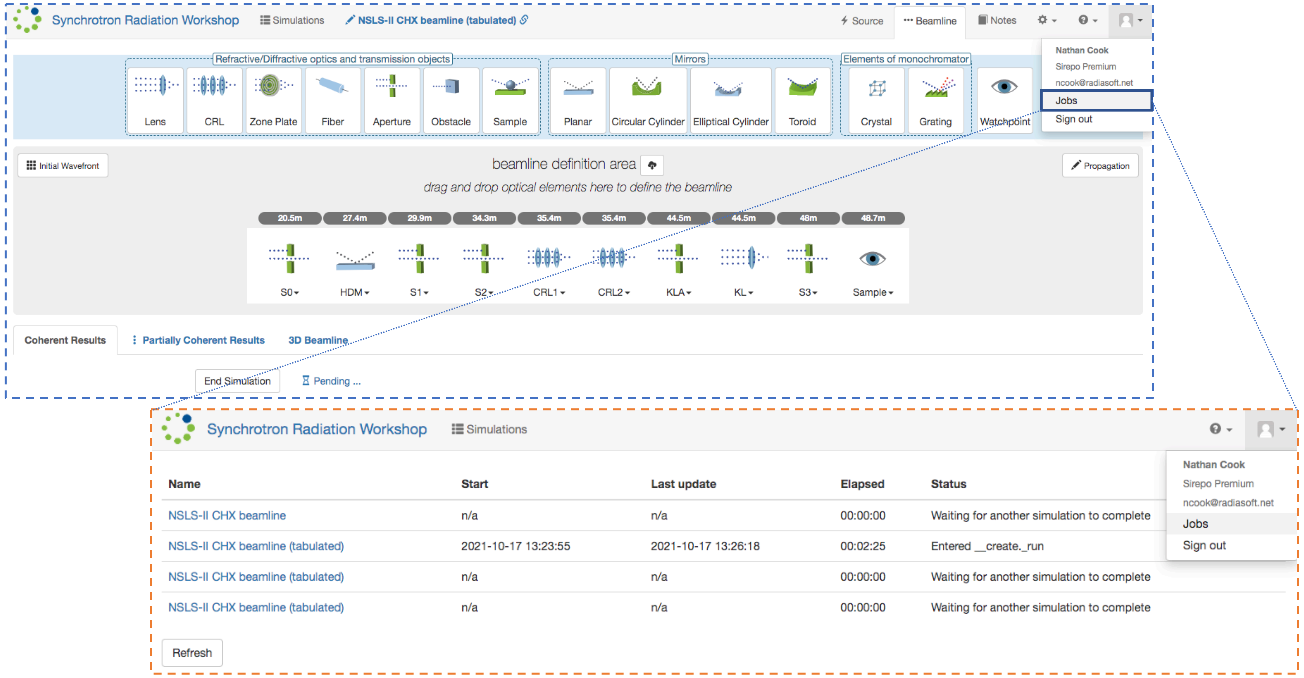


- Customize ROI and image subtraction



Dynamic Queueing

- Sirepo job manager supports asynchronous execution and multiple jobs per user
  - Jobs can introspect relevant metadata to guide the analysis



- Analysis workflow will support user-configurable queueing to re-prioritize UIDs of interest
- Static or dynamic resource allocation
  - Local, on-site cluster, or NERSC execution modes are supported

Disclaimer

Extras

Bluesky is designed to address complete experimental workflows

- Library for experimental control and collection of scientific data and metadata
  - `Bluesky` – experimental design and execution via "plan" schema
  - `Ophyd` – hardware abstraction layer integrates beamline equipment via high level device protocols
  - `Databroker` – I/O library permits access to data in myriad formats via customizable plugins
  - `Suitcase` – serialization capabilities for storage and sharing across networks
- Worldwide community of users and developers
  - NSLS-II – Development home. Deployment across all beamlines.
  - Advanced Photon Source (APS) – Deployment and testing at X-Ray Science Division (XSD) beamlines
  - BESSY II – Berlin, Germany – Bluesky data acquisition and EPICS integration at some beamlines
  - Fritz-Haber-Institut – Berlin, Germany – Bluesky data acquisition and EPICS integration across institute
  - Pohang Light Source II – Pohang, Korea – Bluesky data acquisition for the past year
  - MAX IV – Lund, Sweden – Ophyd integration with Tango for experimental control
  - Additional ongoing efforts to integrate bluesky-queueserver (https://github.com/bluesky/bluesky-queueserver)
- An open source suite of tools
  - Designed to interface with detector tools and related software

bluesky

https://blueskyproject.io/bluesky/

Individual beamlines present unique requirements on workflows

Coherent Hard X-Ray (CHX) Beamline

1. Bluesky launches experimental plan
2. Experimental logging via `Olog`
3. Automated image pre-processing
   1. No background subtraction
4. Jupyter notebook analysis environment
   1. Fixed template with high-level flags
   2. Papermill automates analysis notebook execution pipeline
   3. Decoupled from experimental procedure
   4. Analysis includes: XPCS
5. Analysis saved to separate database
6. Re-tuning on the order of minutes
   1. GBs of data produced every minute
   2. Analysis is ~100x slower than experiment

Coherent Soft X-Ray (CSX) Beamline

1. Bluesky launches experimental plan
2. Experimental logging via `databroker`
3. Manual image pre-processing
   1. Custom background subtraction
4. Jupyter notebook analysis environment
   1. User customization is routine
   2. Notebook execution does not follow an automated pipeline
   3. Decoupled from experimental procedure
   4. Analysis includes: XPCS, CDI, pytchography, …
5. Analysis saved to separate database
6. Re-tuning on the order of hours
   - GBs of data produced every minute
   - Analysis still ~10x slower than experiment