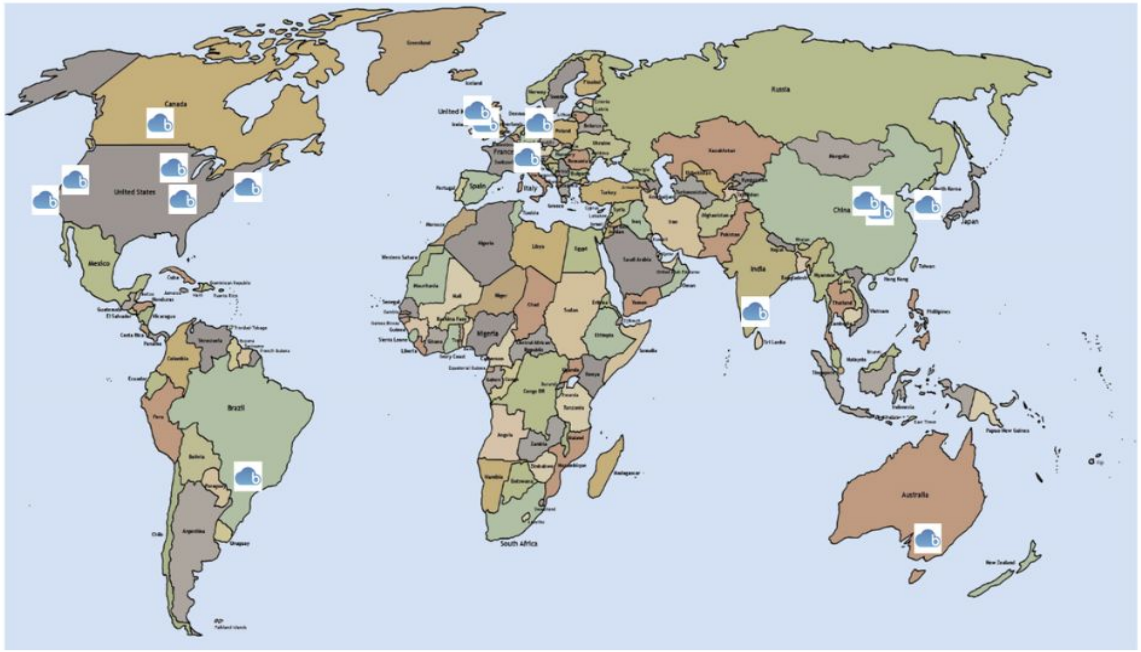
Bluesky Community Meeting

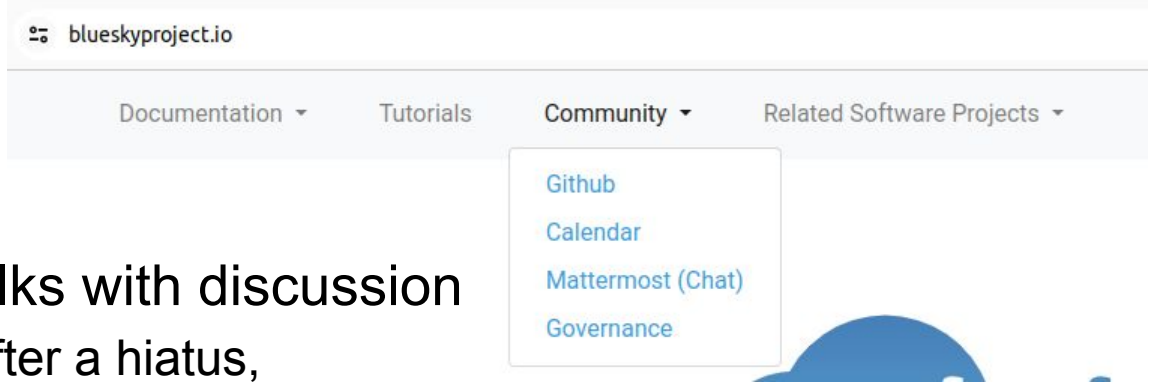
Data Acquisition

Project Updates

Bluesky Community Update



Meetings



- Invited community talks with discussion
  - Recently restarted after a hiatus, now with a proper Planning Committee
  - Aiming for 1-2 talks / month
  - Movable time (10:30, 18:00 NYC) to catch all time zones
  - Recordings posted publicly
- Weekly "dev" call
  - Issue and PR triage
  - Live discussion on architecture, design, and PR review

"Minimum Viable Governance"

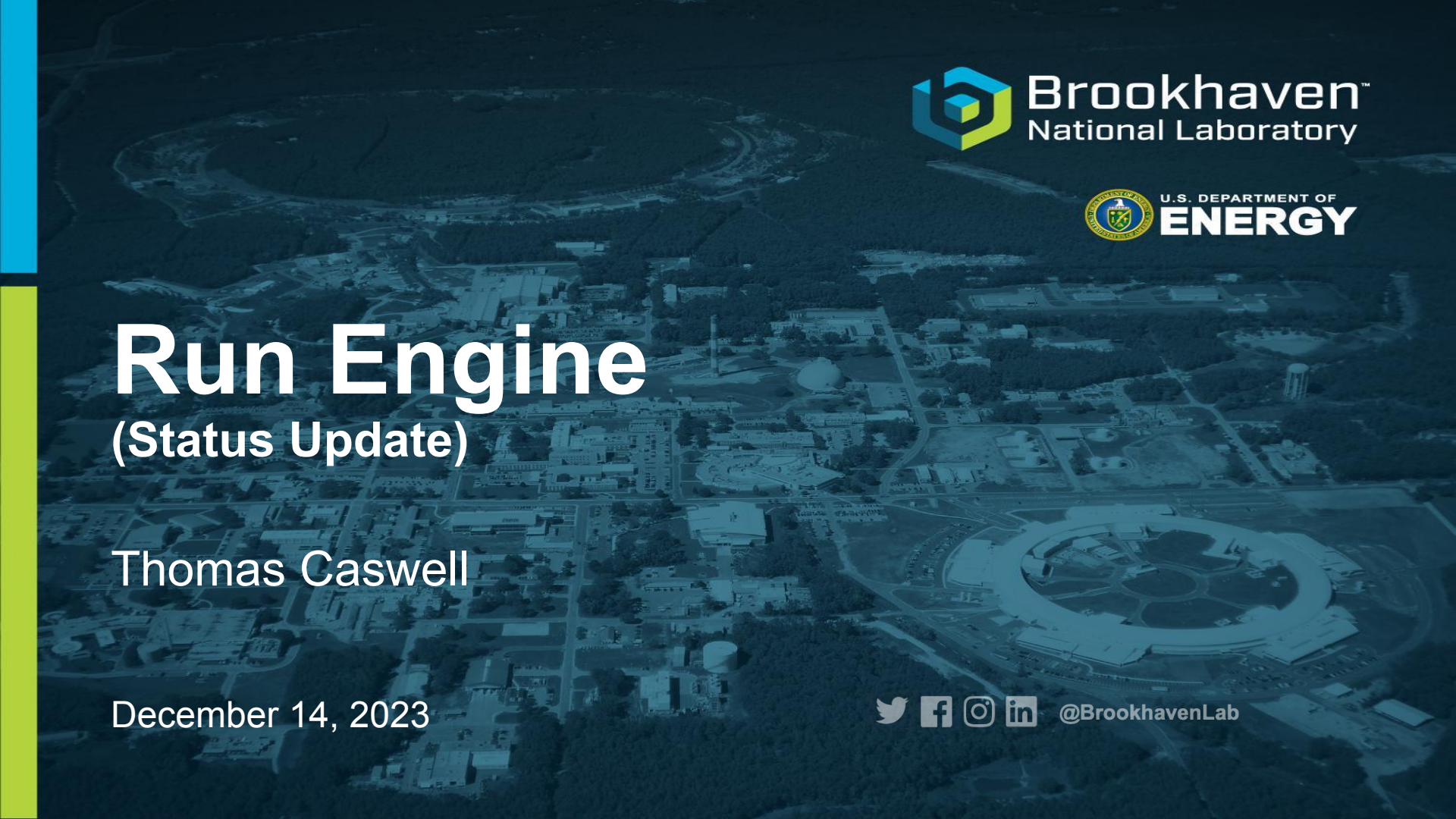
- Repo maintainers make decisions by rough consensus
- If consensus cannot be reached, Technical Steering Committee may intervene and vote if necessary
- Project Advisory Board is a forum for facility management to voice priorities
- github.com/bluesky/governance

Project Advisory Board

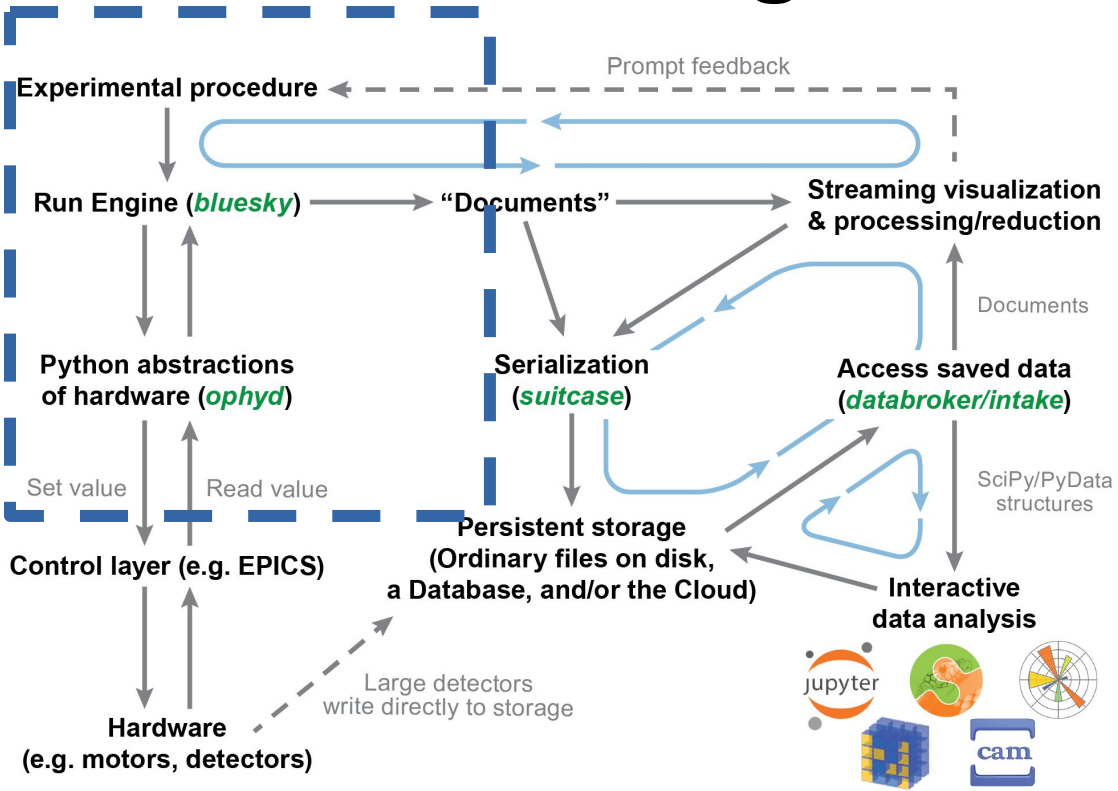
- Alun Ashton (PSI)
- Stuart Campbell (NSLS-II), Chair
- Joe Handford (DLS)
- Mark Heron (DLS)
- Alex Hexemer (ALS)
- Paul Martin (ANSTO)
- Alex Sandy (APS)
- Nicholas Schwarz (APS)
- Jana Thayer (SLAC)
- Stuart Wilkins (NSLS-II)

Technical Steering Committee

- Dan Allan (NSLS-II)
- Thomas Caswell (NSLS-II)
- Tom Cobb (DLS)
- Callum Forrester (DLS)
- Pete Jemian (APS)
- Zachary Lentz (LCLS)
- Dylan McReynolds (ALS)
- Max Rakitin (NSLS-II)
- Clinton Roy (ANSTO)
- Will Smith (HZB)
- Robert Tang-Kong (LCLS)


Brookhaven
National Laboratory
U.S. DEPARTMENT OF
ENERGY
Run Engine
(Status Update)
Thomas Caswell
December 14, 2023
@BrookhavenLab

What is the Run Engine



What does Run Engine do?

Core loop:

1. Consume messages from plan
2. Execute command on ophyd object
3. Return the results to the plan

Handles:

- errors, interrupts, & user feedback
- generating and emitting data

What is new in RE?

- No backwards incompatible changes from v1.0 in 2017
- Improvements to flyscanning protocols
- Plan decorator
  - Optional decorator to warn if plan is not iterated over
- Timeout on wait
  - Do something else and try again later
- Add ability to pre-declare stream
  - Explicitly forces reading the configuration
- Add md_normalizer

StreamResource and StreamDatum

- Source multiple rows and columns at once
- Significant performance improvements
- Required changes to ophyd, bluesky, and tiled

Pre StreamResource/StreamDatum

event	A	B	C	D
1	5.4	D01	D08	D15
2	3.7	D02	D09	D16
3	6.6	D03	D10	D17
4	7.2	D04	D11	D18
5	1.0	D05	D12	D19
6	5.7	D06	D13	D20
7	2.3	D07	D14	D21

With StreamResource/StreamDatum

event	A	B	C	D
1	5.4	SD1		SD2
2	3.7			
3	6.6			
4	7.2			SD3
5	1.0			
6	5.7			
7	2.3			

New and flyscan methods

- ***prepare*** method
  - Between “stage” and “kickoff”
  - Can be passed parameters
  - Call multiple times per stage
- ***collect_pages*** method
  - Get (partial) event pages rather than list of events from flyers
- ***collect_asset_docs*** optional index parameter
  - Generate StreamResource/StreamDatum only up to a given index
  - Useful for aligning multiple devices

Ophyd Async

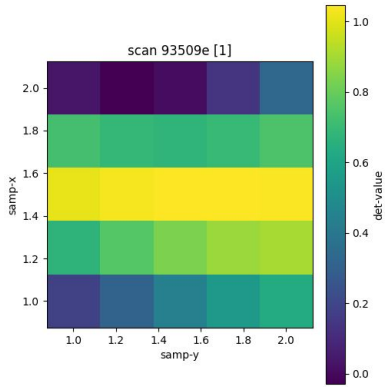
Bluesky Project Update

Very Short Update, Watch Our Talk!

Wednesday, 11:25

Overview

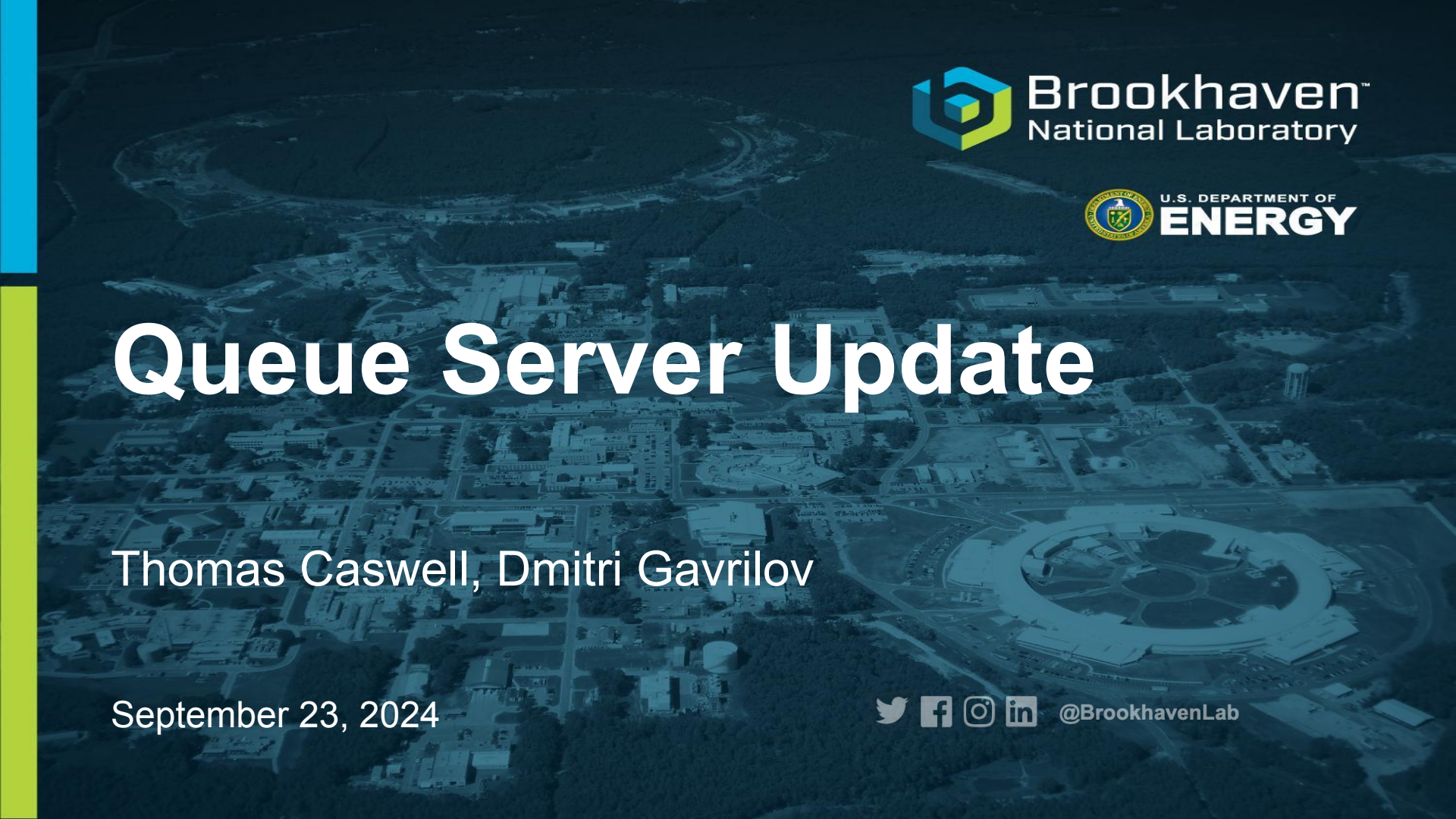
- Replacement for ophyd
- Parity with pymalcolm
- Asyncio
- https://blueskyproject.io/ophyd-async/main/explanations/design-goals.html



Generalisation

Roadmap

- Close to 1.0.0 (targeting late 2024/early 2025)
- Generalising to support Tango
- Trajectory scanning
- Finalising API
- Documentation
- https://github.com/bluesky/ophyd-async/milestone/3



Queue Server Update

Thomas Caswell, Dmitri Gavrilov

September 23, 2024

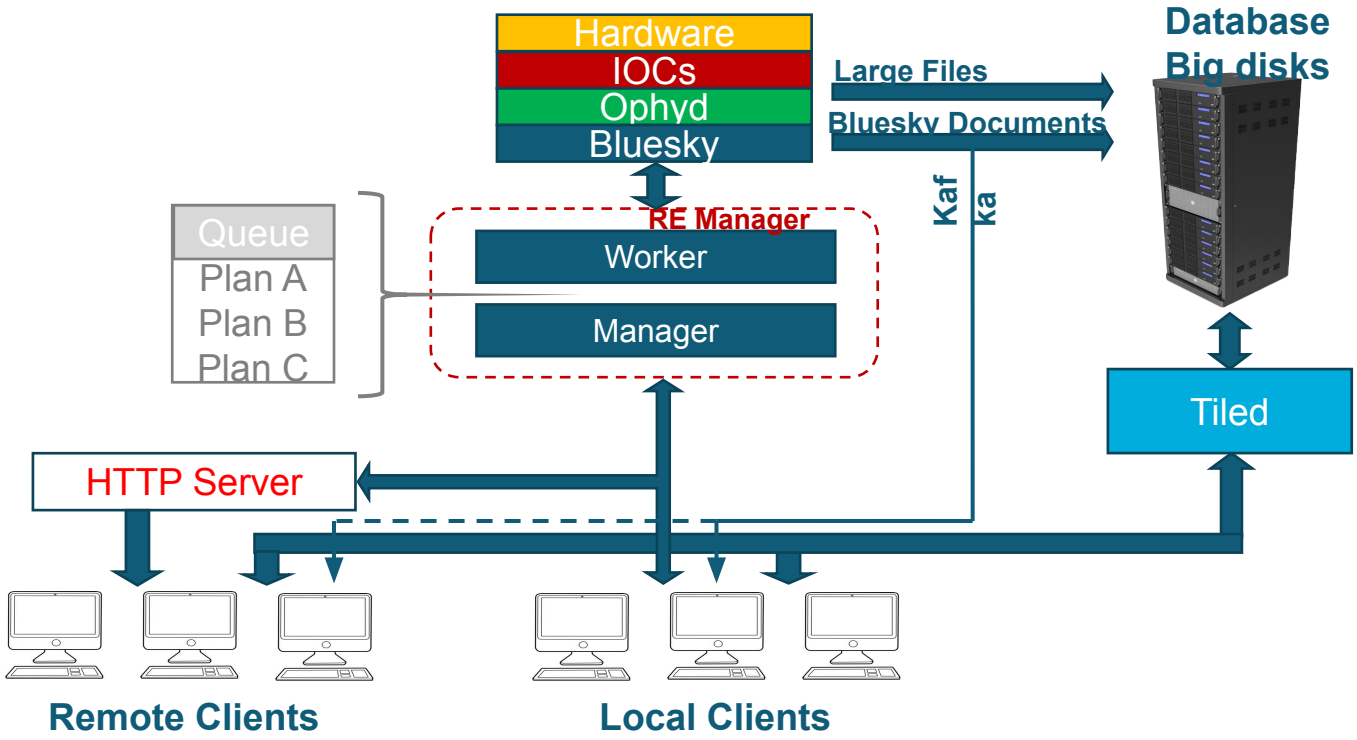
@BrookhavenLab

What does Queue Server do?

1. Manage an environment for plan execution as a service
2. API to directly execute plans or Python in the managed process
3. Manage a mutable-queue of plans scheduled to be executed
4. Enforce access controls to all APIs
5. Provide IPython access for debugging and development

Core layer intended to have user facing tools built on top of

Queue Server in Architecture



Brookhaven National Laboratory

Components of Queue Server stack

- **Run Engine Manager** (RE Manager) – the core component of the stack.
- **HTTP Server** – provides REST API for communicating with RE Manager, authentication and access control.
- **Python API** – user-friendly Python API for communicating with RE Manager directly (over 0MQ) or via HTTP Server (REST API).
- **RE Widgets** are part of Bluesky-Widgets package. The package also includes a generic ready-to-use ‘queue-monitor’ Qt GUI application, which supports basic Queue Server workflows.
- **Multiple UI** Locally developed web apps or desktop applications (*we should accumulate a list of these today!*)

Known Users

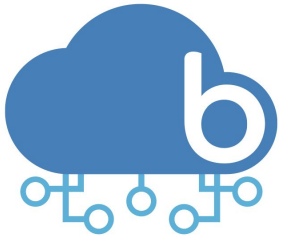
- NSLS-II
- ALS
- SLAC
- APS
- Australian Synchrotron
- BESSY
- Univ. of Wisconsin

Documentation

The **bluesky-queueserver, blueskyhttp-server** and **bluesky-queueserver-api** packages are extensively documented:

- **bluesky-queueserver** – tutorials, installation instructions, configuration notes, detailed API documentation: https://blueskyproject.io/bluesky-queueserver
- **bluesky-httpserver** – configuration notes, authentication and access control API documentation: https://blueskyproject.io/bluesky-httpserver
- **bluesky-queueserver-api** – API descriptions (also available as docstrings): https://blueskyproject.io/bluesky-queueserver-api/

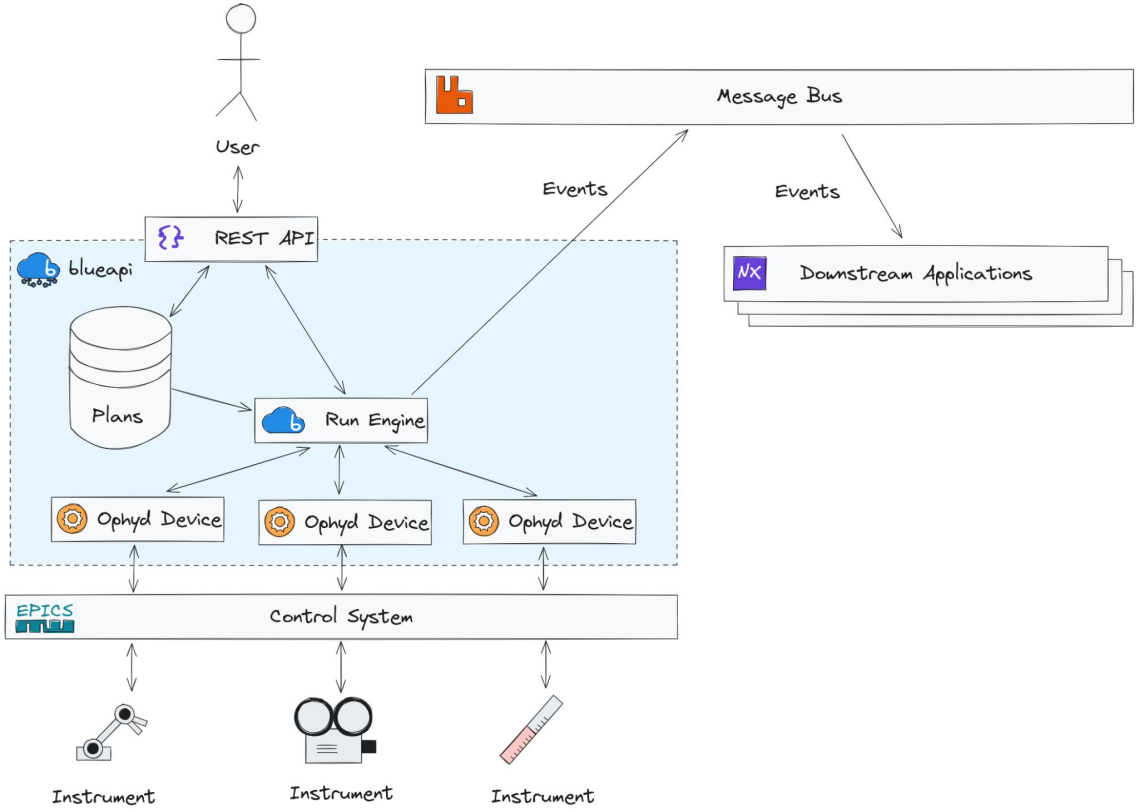
RE Widgets are documented in code. The `queue-monitor` GUI application can be used as a starting point for developing custom applications.



Blueapi

Callum Forrester

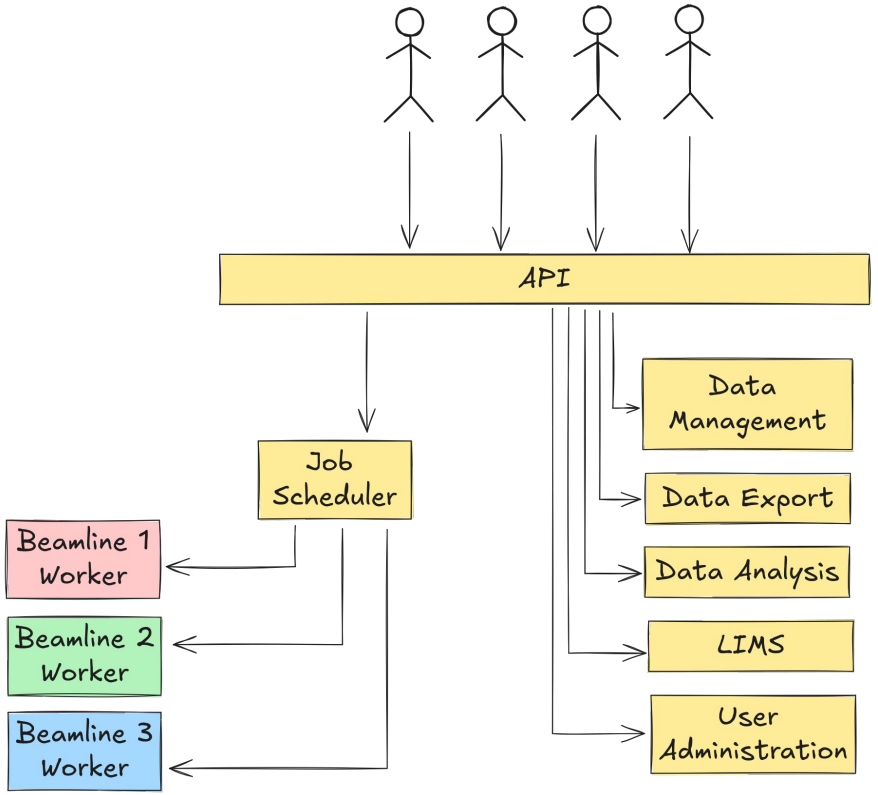
Overview



https://github.com/DiamondLightSource/blueapi

DLS (Aspirational) Model

- Only run on a beamline what *needs* to run on a beamline
- IOCs, RunEngine
- One service, one job, but not microservices



Comparison to Queueserver

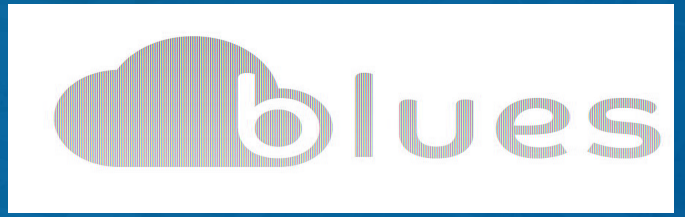
- Similar to “Worker” component
- No queue
- No ZMQ layer
- Automated version control (planned)

Roadmap

- Close to 1.0.0 (targeting late 2024/early 2025)
- Improved configuration
- Plan customisation UX
- API refinements
- Documentation
- https://github.com/DiamondLightSource/blueapi/milestone/16

2024-09-18

hklpy: Bluesky Diffractometers
Status and Future



Pete R JEMIAN
Physicist
Advanced Photon Source
Argonne National Laboratory

Ken LAUER (original author)
Max RAKITIN
Padraic SHAFER

NSLS-II
Brookhaven National Laboratory

hklpy: Bluesky Diffractometer Support

Where we are

- Current package ***hklpy*** (v1.1.1, 2024-08): https://blueskyproject.io/hklpy/
- Common diffractometer geometries are supported. A local community is forming.
- Scan in real- or reciprocal-space, save & restore orientation (UB matrix)
- Easy to create simulations of any supported diffractometer geometry. Some are pre-built.
- Lots of documentation, including for transition from certain legacy software.

Where we are going

- Major upgrade in development now: ***hklpy2***: https://prjemian.github.io/hklpy2/
- Working with *Hkl/Soleil* developer for new diffractometer geometries. ***Hklpy*** & ***hklpy2*** use same backend library.
- Be able to choose backend library from a list. Currently only *Hkl/Soleil*.
- Lots of documentation (planned).
- Clearer design than (previous) ***hklpy***.

What we need

- Tools should be much more convenient for both new and experienced users.
- Simplify the process to add a new diffractometer geometry.
- More time to develop hklpy2 for first release.


Argonne
NATIONAL LABORATORY
Advanced
Photon Source

Facility Updates

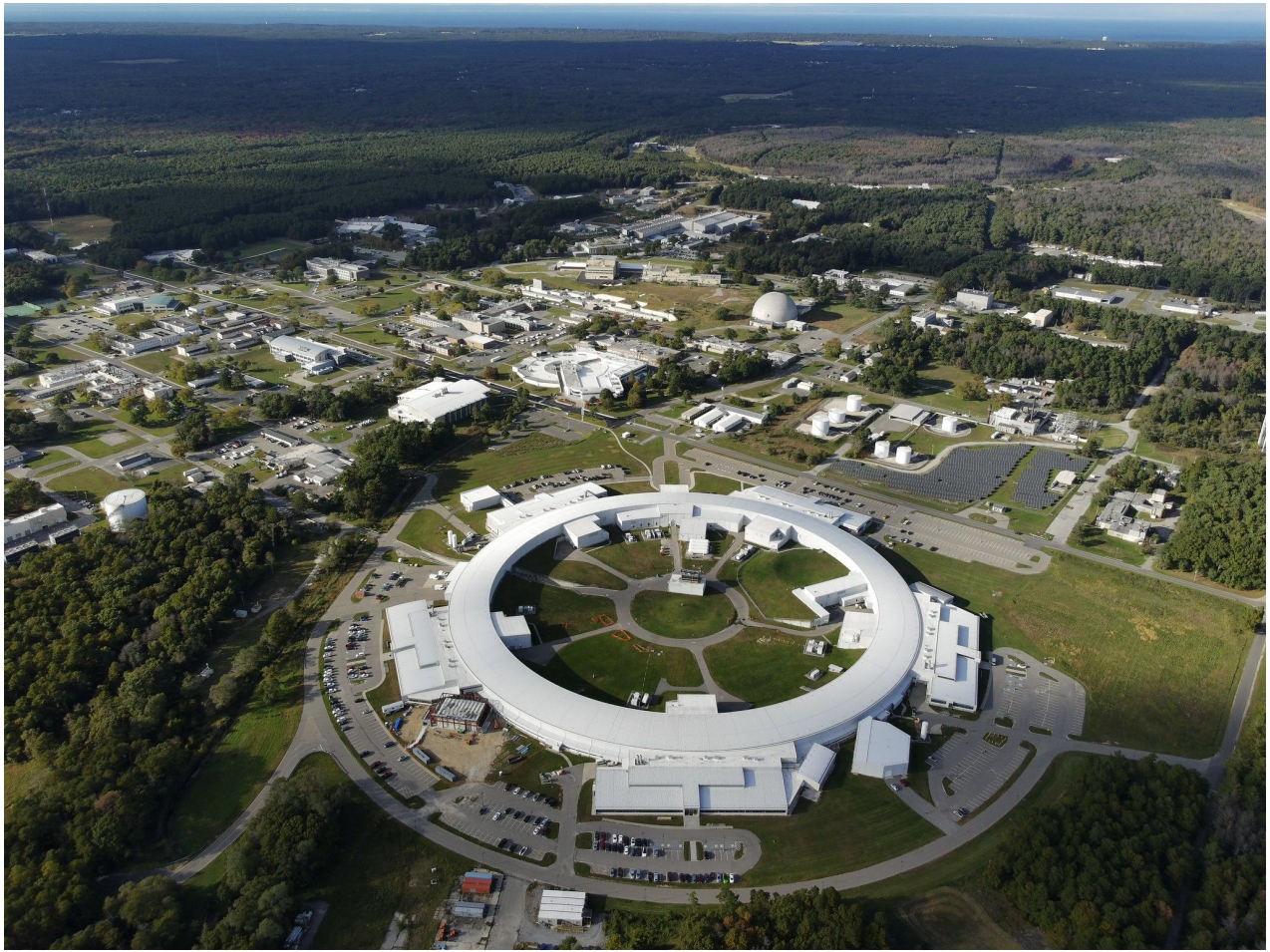
National Synchrotron Light Source II

Brookhaven National Laboratory

U.S. DEPARTMENT OF ENERGY

NSLS-II Update

Date



National Synchrotron Light Source II

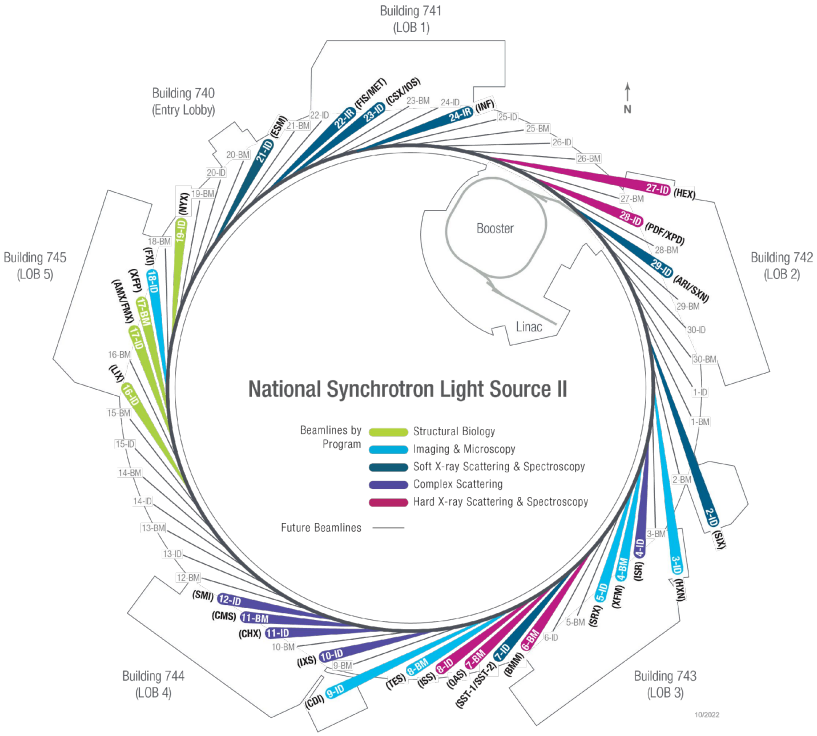
National Synchrotron Light Source II

U.S. Department of Energy Facility

- 3 GeV electron beam energy
- Infrared to Hard X-rays
- 792m in circumference
- Designed for current up to 0.5A
- Can host ~ 60 Beamlines;
- Currently 29 in user operations
- 3 beamlines in construction
- 5 more on the drawing board

Unprecedented ramp-up

- *First light October 2014*
- GU ops started July 2015
- 27 Operating Beamlines in < 5 years



Who is here at NOBUGS

Stuart Campbell

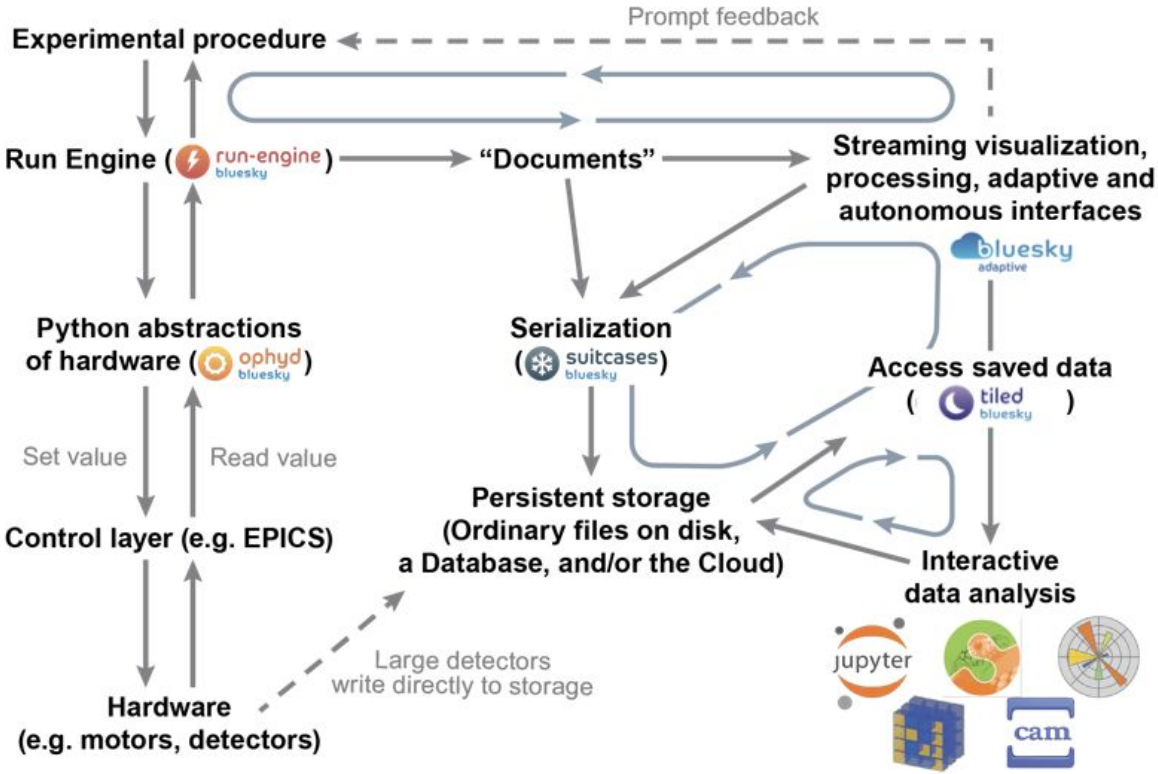
Thomas Caswell

Dan Allan

Robert Schaffer

Stuart Wilkins (arriving Tues)

What parts of bluesky are we using?

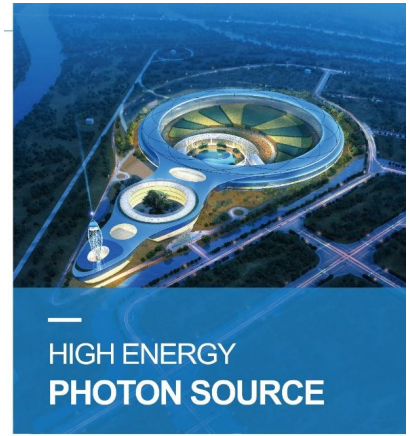


Where are we focusing

- Building the bluesky community
- Adaptive Experimentation
  - Bluesky Adaptive
  - Queueserver
- Tiled
- Fly Scanning

Institute of High Energy Physics

The *Mamba* software project for HEPS

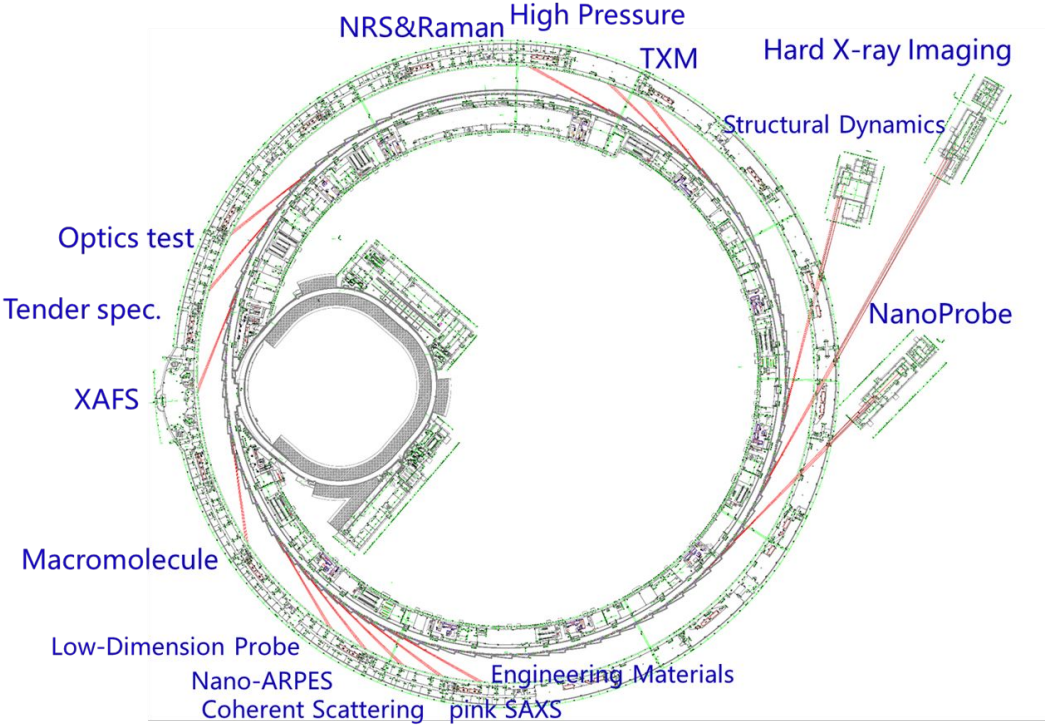


MAMBA
Data acquisition software

bluesky

Yi Zhang

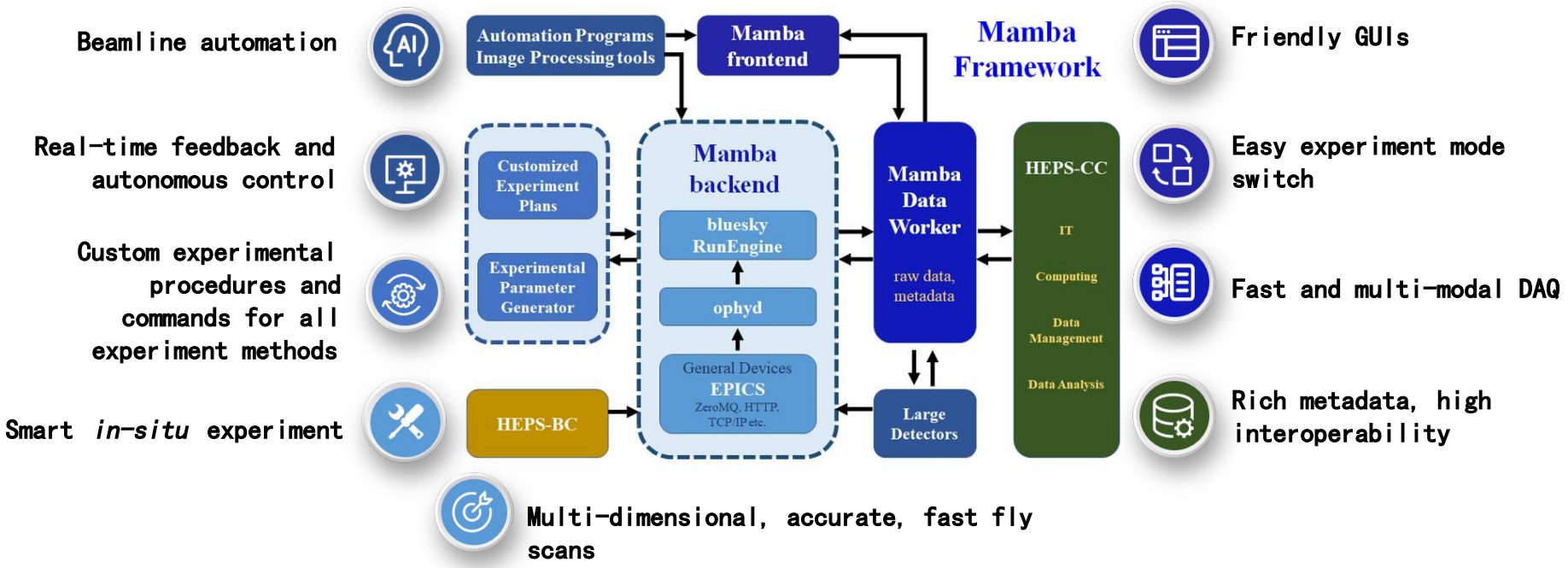
Phase I project for HEPS



- 15 beamlines for Phase I project
- Multiple experimental modes and methods for single beamline
- Up to 30 suites of acquisition software delivered by 2025
- Limited personnel and lack of experience
- A systematic solution for all Phase I and future beamlines

An unified synchrotron experiment operating software system

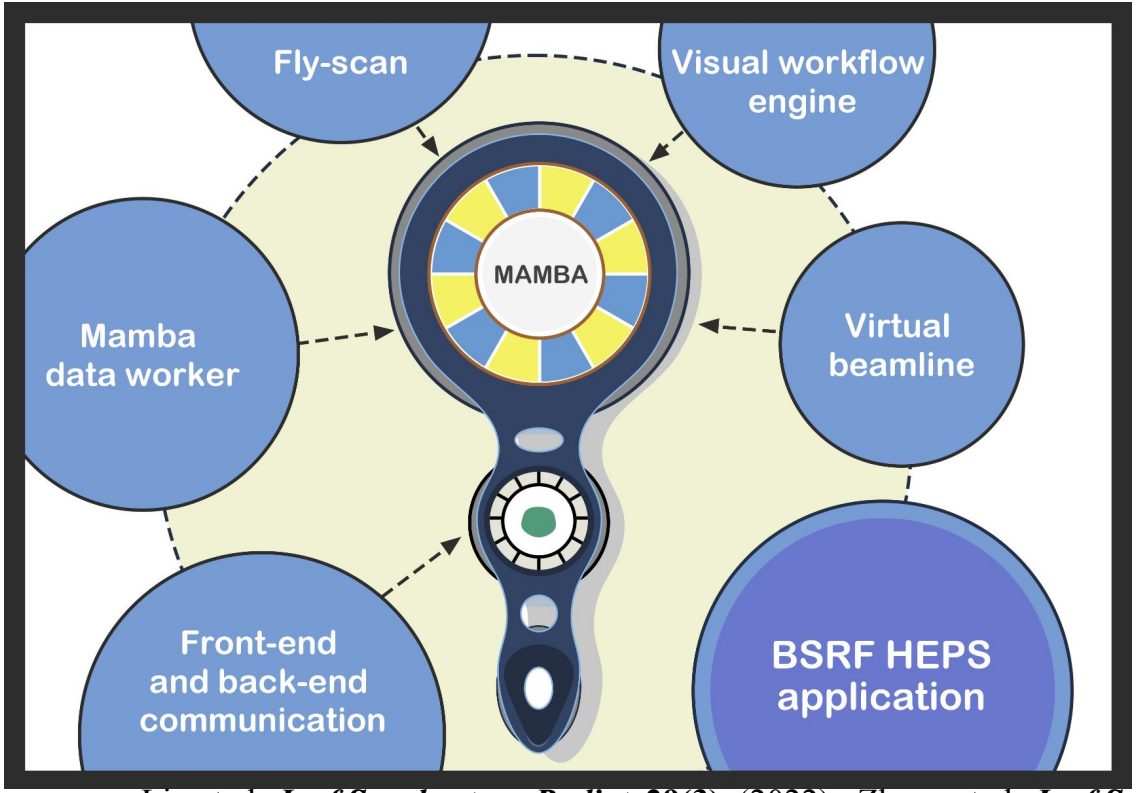
(*Mamba*)



Mamba: a systematic software solution for beamline experiments at HEPS. ***Journal of Synchrotron Radiation***, 2022

Explore key techniques of new generation control and acquisition system

- **Already**



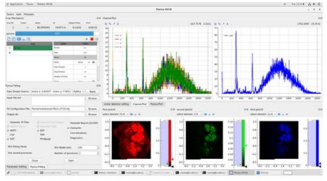
- **On going**

- Python IOC, Systematic detector integration and High-performance readout
- Beamline and experiment specific plan and GUI library
- Versatile attitude tuning framework and beamline automation
- Closed-loop control based on real-time data analysis
- AI-Enabled experimental control

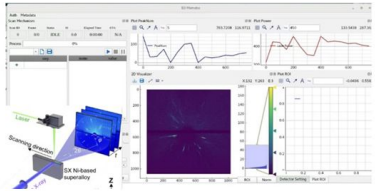
Liu et al. ***J. of Synchrotron Radiat.*** **29(3)**, (2022) ; Zhang et al. ***J. of Synchrotron Radiat.*** **30(1)**, (2023) ; Li et al. ***J. of Synchrotron Radiat.*** **30(6)**, (2023); Li et al. ***Synchrotron Radiat. News***, (2023); Wang et al. ***J. of Synchrotron Radiat,*** (2024);

Progress of *Mamba* project

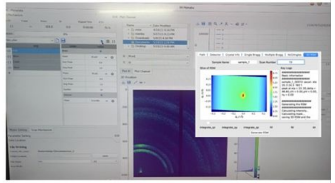
Unified framework for all HEPS beamlines



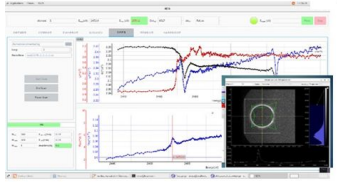
XRF fly scan



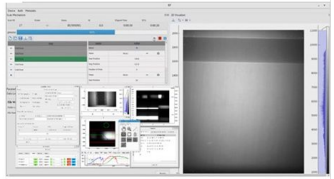
Online feedback in additive manufacturing XRD



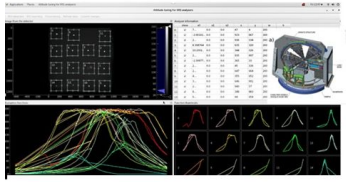
GISAXS & GIWAXS



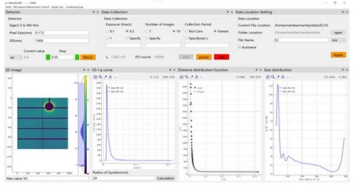
XAS & XES



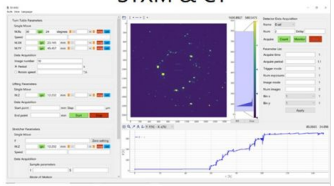
STXM & CT



One-shot attitude adjustment for Raman spectrometer

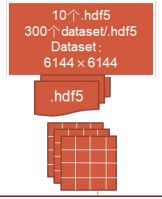


Combined SAXS & WAXD

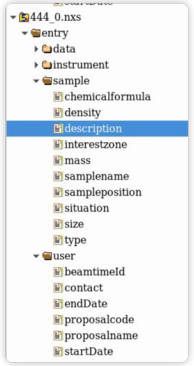


In-situ 3D XRD

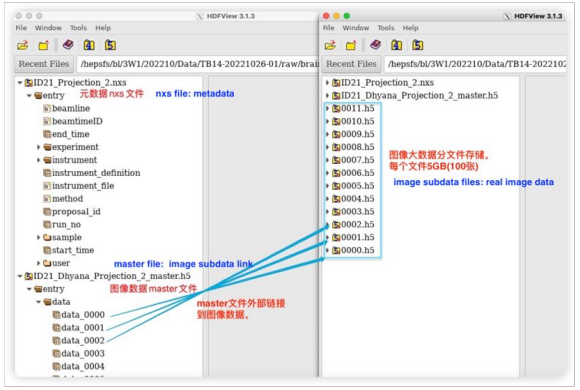
Unified data format and standards



- 假设每300张（可调）存成一个data文件，
- 文件（1个master文件，10个hdf5文件）
  - ✓ ID21_count(scan)_Dhyana_*RunID*_master.h5
  - ✓ ID21_count(scan)_Dhyana_*RunID_00000*_data.h5
  - ✓ ID21_count(scan)_Dhyana_*RunID_00001*_data.h5
  - ✓ …
  - ✓ ID21_count(scan)_Dhyana_*RunID_00009*_data.h5



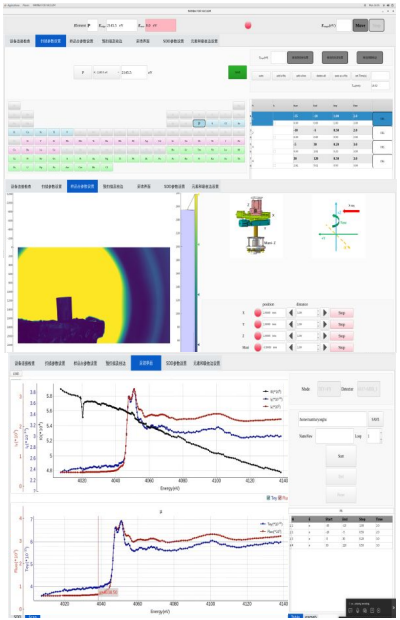
Single nxs file

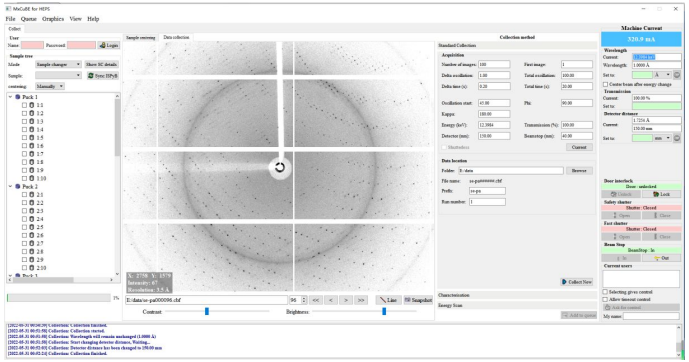


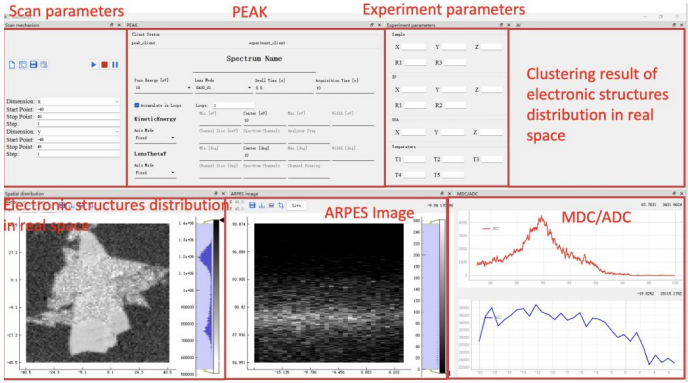
nxs+master+ data0+data1+…

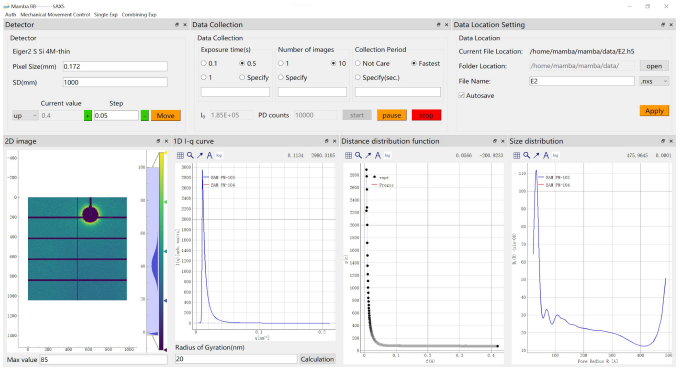
Leading the way in making scientific data FAIR in China

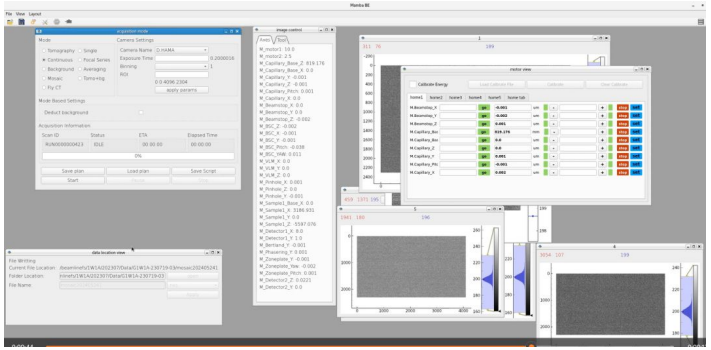
Progress of *Mamba* project





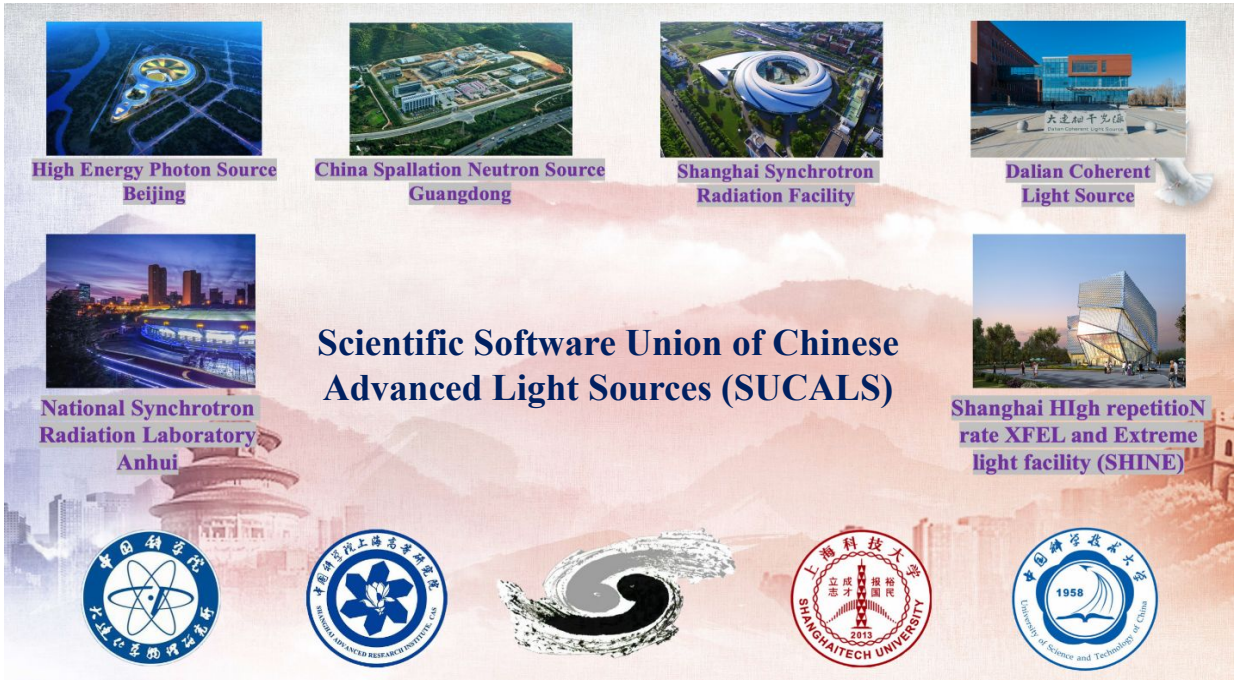






- **The first batch of *Mamba* software for *HEPS* beamline is underway**

Domestic and international collaboration



Funding from:

- National Key Research and Development Project of China;
- Chinese Academy of Sciences;

We are planning to form and develop a nation-wide united scientific software solution in China.

Internationally, we are looking for collaboration from other synchrotron sources.

Australian Government

ANSTO

Bluesky Satellite Meeting

Data Acquisition

NOBUGS 2024 · ILL & ESRF · Grenoble, France

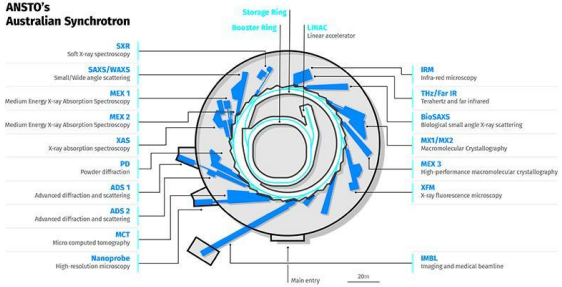
Monday 23rd September

Letizia Sammut

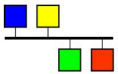
Senior *Scientific Computing Software Engineer*
Australian Synchrotron, ANSTO

Science. Ingenuity. Sustainability.

ANSTO



- 3 GeV Machine, first light 2006
- Staff of ~150
- 5500+ visits per year
- 10 original operating experimental beamlines
- 8 new (BRIGHT) beamlines (3 already operating)
- 586 Journal Publications in 2022
- Generate 2.5 PB of data each year
- EPICS



OPAL multi-purpose reactor

Clayton | VIC

Australian Synchrotron

Lucas Heights | NSW

Main campus

ANSTO

The Scientific Computing Team

Scientific Computing founded in June 2017

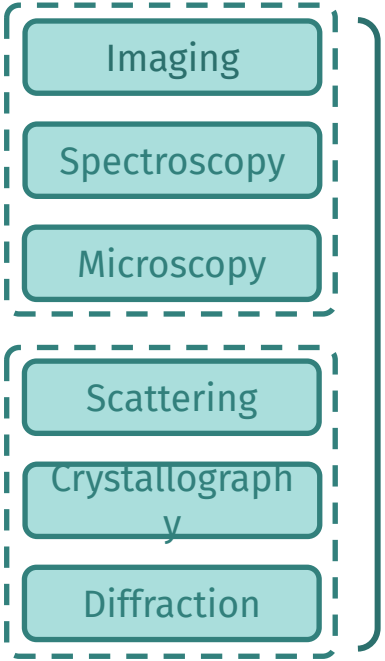
Support Science and Users

- Experiment Control
- Data Acquisition
- Data Processing
- Data Analysis

Our Team

- 1 manager
- 17 members
  - 1 principal engineer
  - 10 PhDs
  - 47% gender split

Beamline Groups



Cross-functional teams

- Experiment Control
- Data Acquisition
- Processing
- GUI

ANSTO

Data Collection

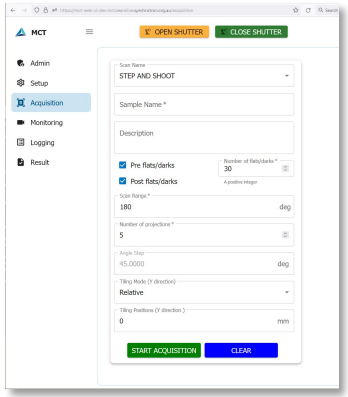
Hardware abstraction

ophyd
bluesky

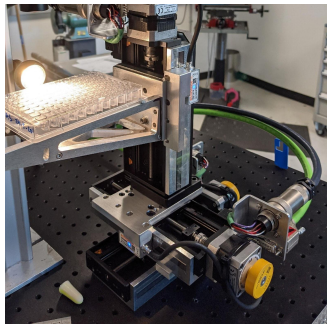
Orchestration

bluesky
PREFECT

python bluesky



Hardware



Documents/Metadata

kafka
mongoDB.

Data Streaming

ØMQ
redis

Data Collection/ Aggregation/Consumers

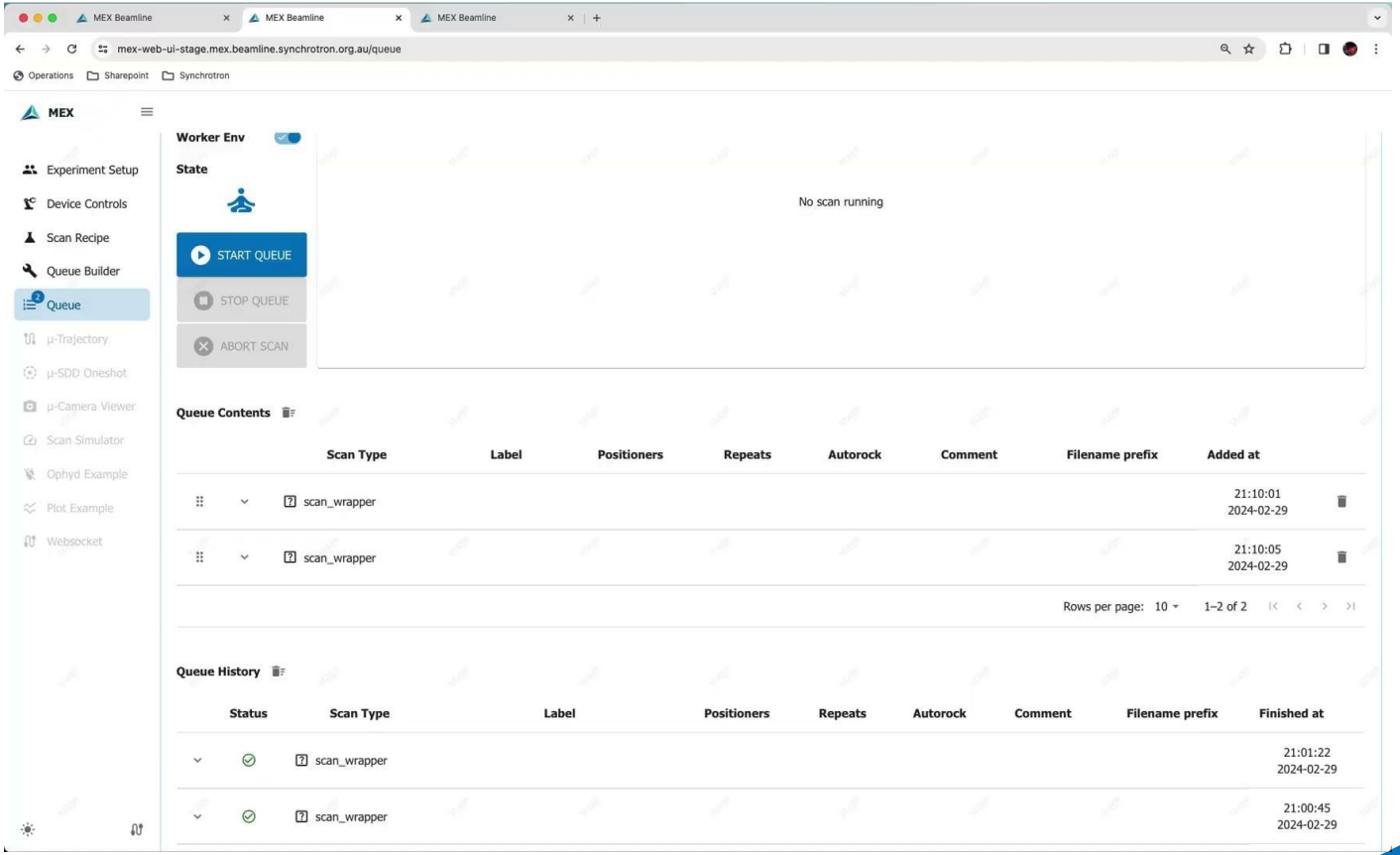
FastAPI
python

Local Storage

Data Product

ANSTO

Example: queue interface



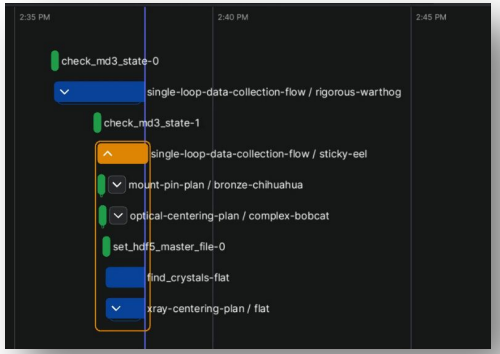
Thanks

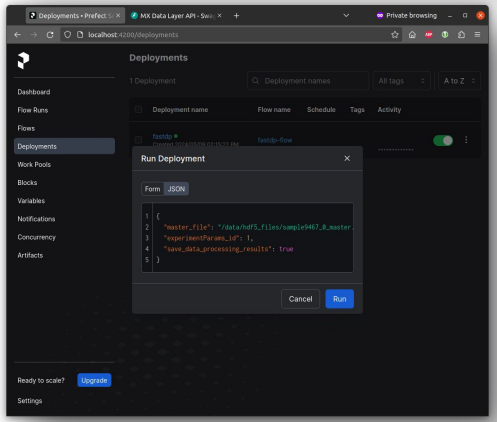
ANSTO

Experiment Orchestration with Prefect

Prefect as the main workflow engine

- Bluesky plans are executed as prefect flows using a local Run Engine
- Plans are queued by a parent prefect flow
- The state of all flows can be easily tracked using the Prefect client



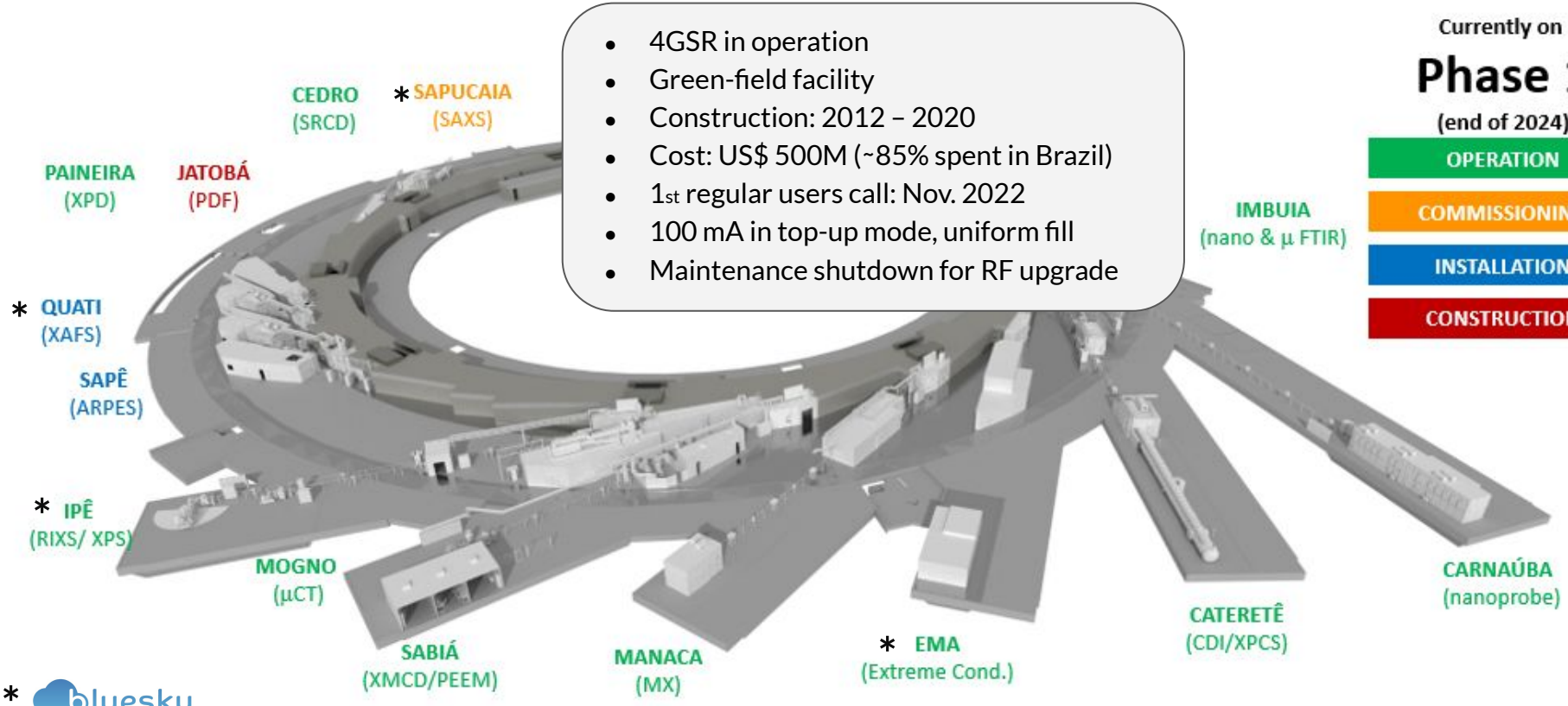


Advantages

- Visualisation of flows via the Prefect UI
- Fewer services to be maintained
- Easier debugging

SIRIUS: overview and status

- 4GSR in operation
- Green-field facility
- Construction: 2012 – 2020
- Cost: US$ 500M (~85% spent in Brazil)
- 1st regular users call: Nov. 2022
- 100 mA in top-up mode, uniform fill
- Maintenance shutdown for RF upgrade



* bluesky

https://www.lnls.cnpem.br/beamlines/

Bluesky at SIRIUS

- Who is here from your facility?

  Ana Clara de Souza Oliveira (ana.clara@lnls.br)

  Igor Ferreira Torquato (igor.torquato@lnls.br)

- What pieces of Bluesky are you using?
  - Bluesky project packages:

  ophyd, bluesky , queueserver, httpserver, queueserverapi, databroker, bluesky-widgets

  - Other facilities related initiatives:

  ophyd registry

  - Inspirations from

  apstools, haven, psdsdevices

- What topics or future developments are you especially interested in?
  - ergonomic solutions for flyscan (ophyd-async)
  - ergonomic solutions for dynamic signals for devices (ophyd-async)
  - improvements in remote RE control
    - possibility of controlling background tasks as you would with the RE (queueserver)
    - better fast api documentation for routes definition (httpserver)
  - more and better generic data visualization tools (bluesky callbacks)
    - options to interact with the data (plot several curves, choose curves visibility, etc)
    - customize views (choose legends, colors, plot with log axis, etc)
    - more complex views (histograms, 3d plots, etc)
  - web GUI solutions

Science and Technology Facilities Council

ISIS Neutron and Muon Source

A Brief Bluesky Update from ISIS

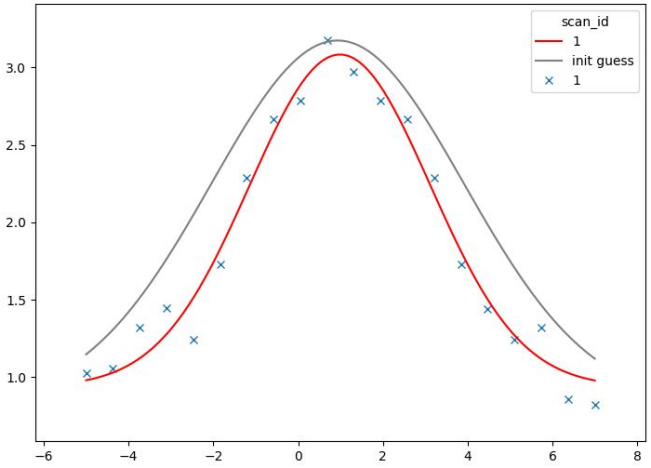
(On behalf of ISIS Experiment Controls)

Science and Technology Facilities Council

ISIS Neutron and Muon Source

Dynamic Guesses on Fits

- Neutron data often has low statistics/low numbers of points.
- To get fitting functions to converge reliably we need better initial guesses than users are able to provide up-front.
- We're dynamically generating initial guesses from the scan data we've collected so far and updating these guesses on every scan point.
- This seems to work quite well in our early prototypes, especially for "easy" fit functions like gaussian peaks etc.

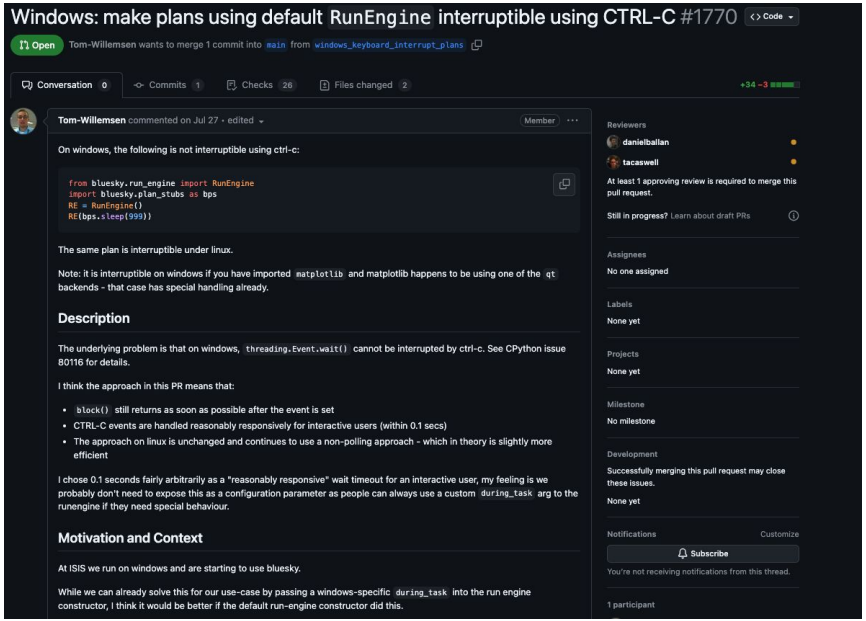


Uncertainty Handling & Propagation

- Uncertainty propagation is fundamental for neutron data as the counts are often relatively low compared to X-ray sources so uncertainties are much more significant.
- We plan to record uncertainties in output data files which are beneficial for displaying as error bars on plots.
- Passing uncertainties through as weights to fitting functions so that points are weighted correctly according to standard deviation.
- The scipp libraries which are inspired by Xarray (https://scipp.github.io/, https://github.com/scipp/scippneutron) are of interest to help deal with uncertainty propagation and lightweight unit conversions from neutron time-of-flight to wavelength or energy etc.

Running Windows

- Running on windows has raised a few issues in both Bluesky and Ophyd-async.
- Some PRs have already been submitted upstream, but there is more to follow in time from Experiment Controls.



Windows: make plans using default `RunEngine` interruptible using CTRL-C #1770

Code

Open · Tom-Willemsen wants to merge 1 commit into `main` from `windows_keyboard_interrupt_plans`

Conversation 0 · Commits 1 · Checks 26 · Files changed 2 · +34 −3

Tom-Willemsen commented on Jul 27 • edited · Member

On windows, the following is not interruptible using ctrl-c:

```
from bluesky.run_engine import RunEngine
import bluesky.plan_stubs as bps
RE = RunEngine()
RE(bps.sleep(999))
```

The same plan is interruptible under linux.

Note: it is interruptible on windows if you have imported `matplotlib` and matplotlib happens to be using one of the `qt` backends - that case has special handling already.

Description

The underlying problem is that on windows, `threading.Event.wait()` cannot be interrupted by ctrl-c. See CPython issue 80116 for details.

I think the approach in this PR means that:

- `block()` still returns as soon as possible after the event is set
- CTRL-C events are handled reasonably responsively for interactive users (within 0.1 secs)
- The approach on linux is unchanged and continues to use a non-polling approach - which in theory is slightly more efficient

I chose 0.1 seconds fairly arbitrarily as a "reasonably responsive" wait timeout for an interactive user, my feeling is we probably don't need to expose this as a configuration parameter as people can always use a custom `during_task` arg to the runengine if they need special behaviour.

Motivation and Context

At ISIS we run on windows and are starting to use bluesky.

While we can already solve this for our use-case by passing a windows-specific `during_task` into the run engine constructor, I think it would be better if the default run-engine constructor did this.

Reviewers: danielballan, tacaswell

At least 1 approving review is required to merge this pull request.

Still in progress? Learn about draft PRs

Assignees: No one assigned

Labels: None yet

Projects: None yet

Milestone: No milestone

Development: Successfully merging this pull request may close these issues. None yet

Notifications · Customize · Subscribe

You're not receiving notifications from this thread.

1 participant

The Not So Novel Stuff We're Doing

- Using Ophyd-async to communicate Bluesky back and forth with EPICS.
- Having a "Core" repo for common devices, plan stubs and RE configuration etc.
- A beamline specific configuration is likely to live in existing areas "owned" by science groups.
- Basic scans such as 1-D step-scans for the most part, maybe ramp scans in a few places.
- Currently not using queue server/blueapi solution, just running interactively.
  - Not necessarily excluding this route in the future through…
- Already have matplotlib built into IBEX and this seems to work with Bluesky plotting callbacks.

General Design Mentality

- In general, we are trying our best to stay in-line with Bluesky's intended usage of Bluesky as best as possible.
- Some aspects of our implementation may go against this, but this is to be expected when trying to fit a new framework in with a long standing controls software project.

HELMHOLTZ

BLUESKY AT DESY

Development of Tango support for Ophyd-Async and structured metadata for operando catalysis

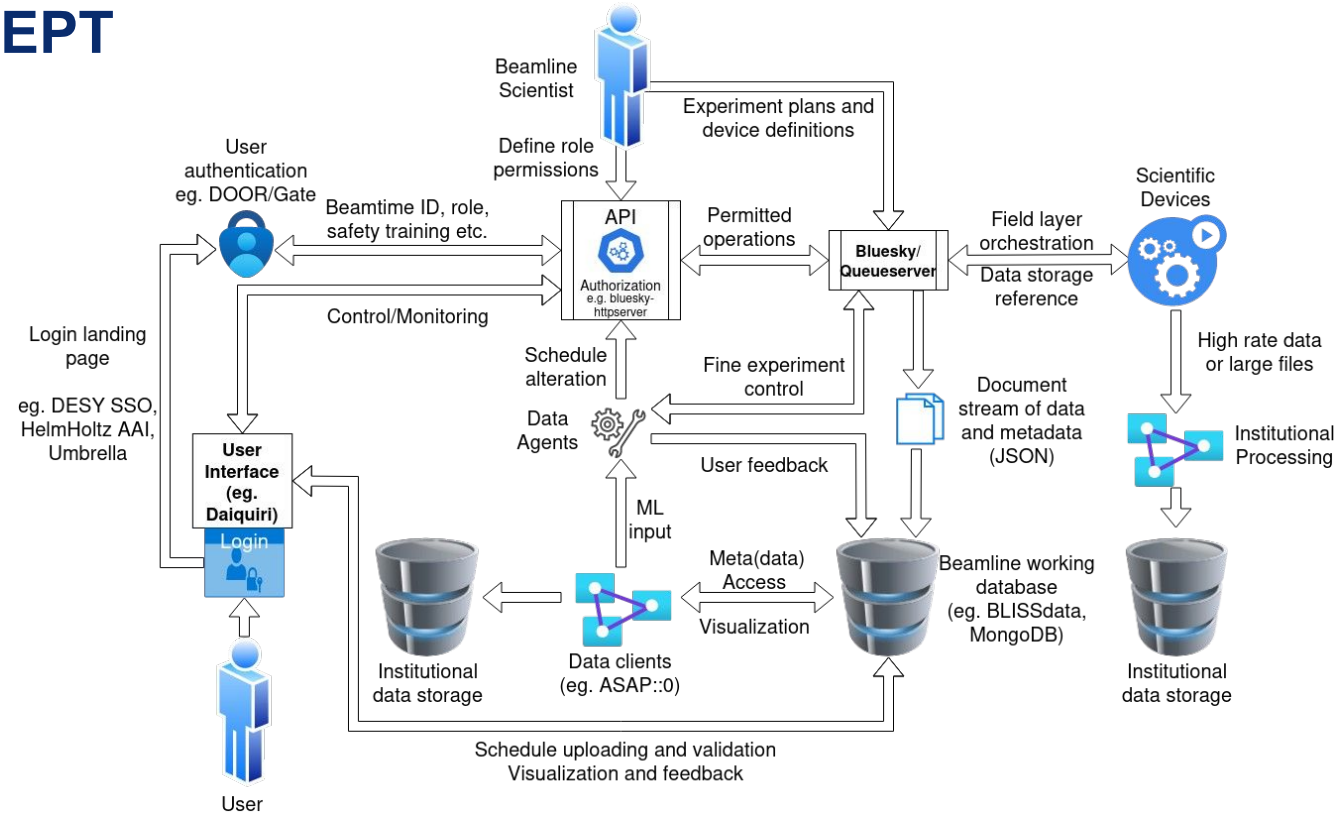
Devin Burke on behalf of DESY and the ROCK-IT project

OUR USE OF BLUESKY IS MOTIVATED BY THE ROCK-IT PROJECT

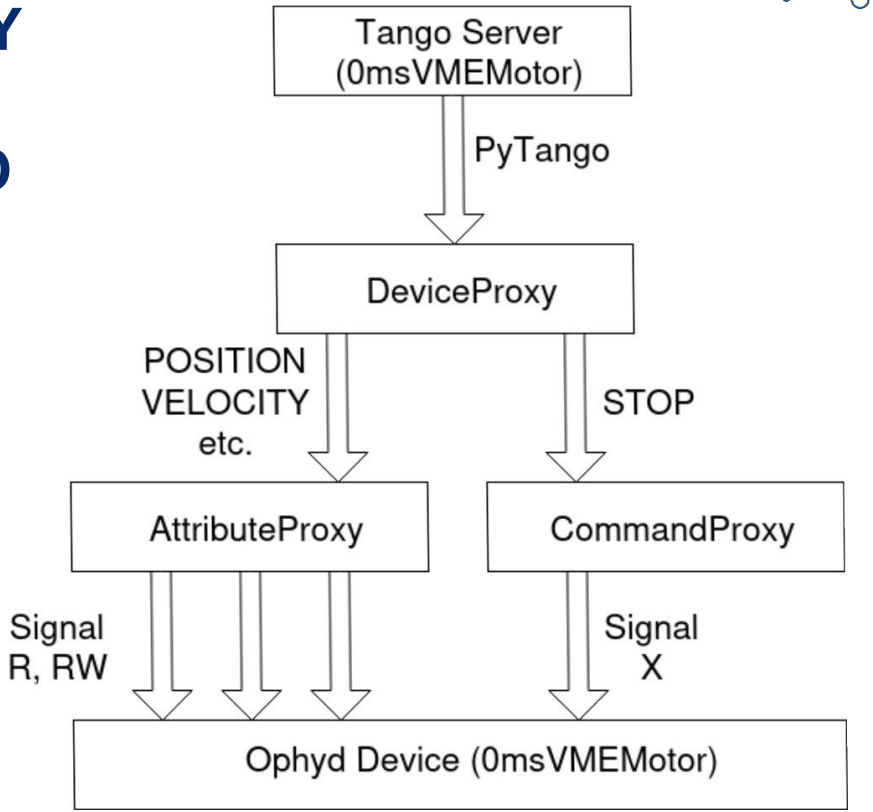
Project Goals

- Create user-friendly automated experiment environments for non-experts and industry users.
- Extend in-situ and operando mail-in experiment capabilities with remote access.
- Improve instrument accessibility by enabling remote experiment control via web browser.
- Develop a general purpose set of tools which can be easily transferrable to new types of instruments and experiments.
- Implement machine-learning for automated experiments, real-time analysis, and robotic sample handling.

BLUESKY AND THE QUEUESERVER WILL FORM THE CORE OF THE ROCK-IT AUTOMATION CONCEPT

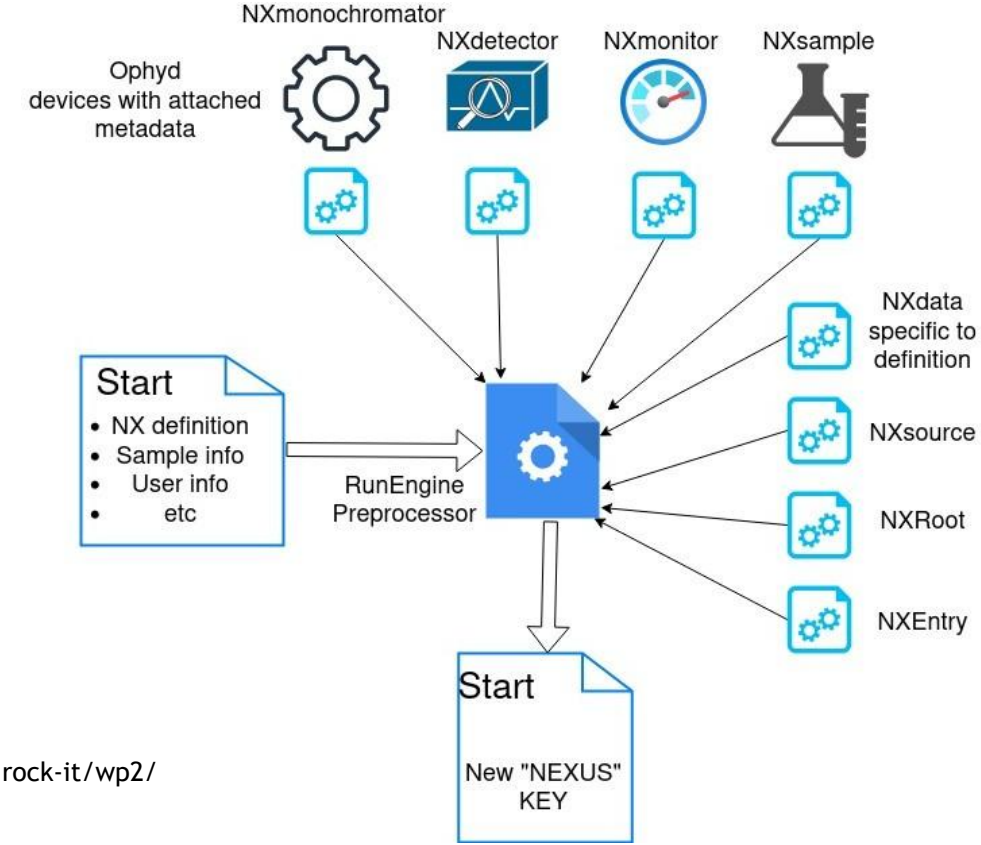


USE OF BLUESKY AT DESY IS ENABLED BY DEVELOPMENT OF TANGO SUPPORT FOR OPHYD-ASYNC



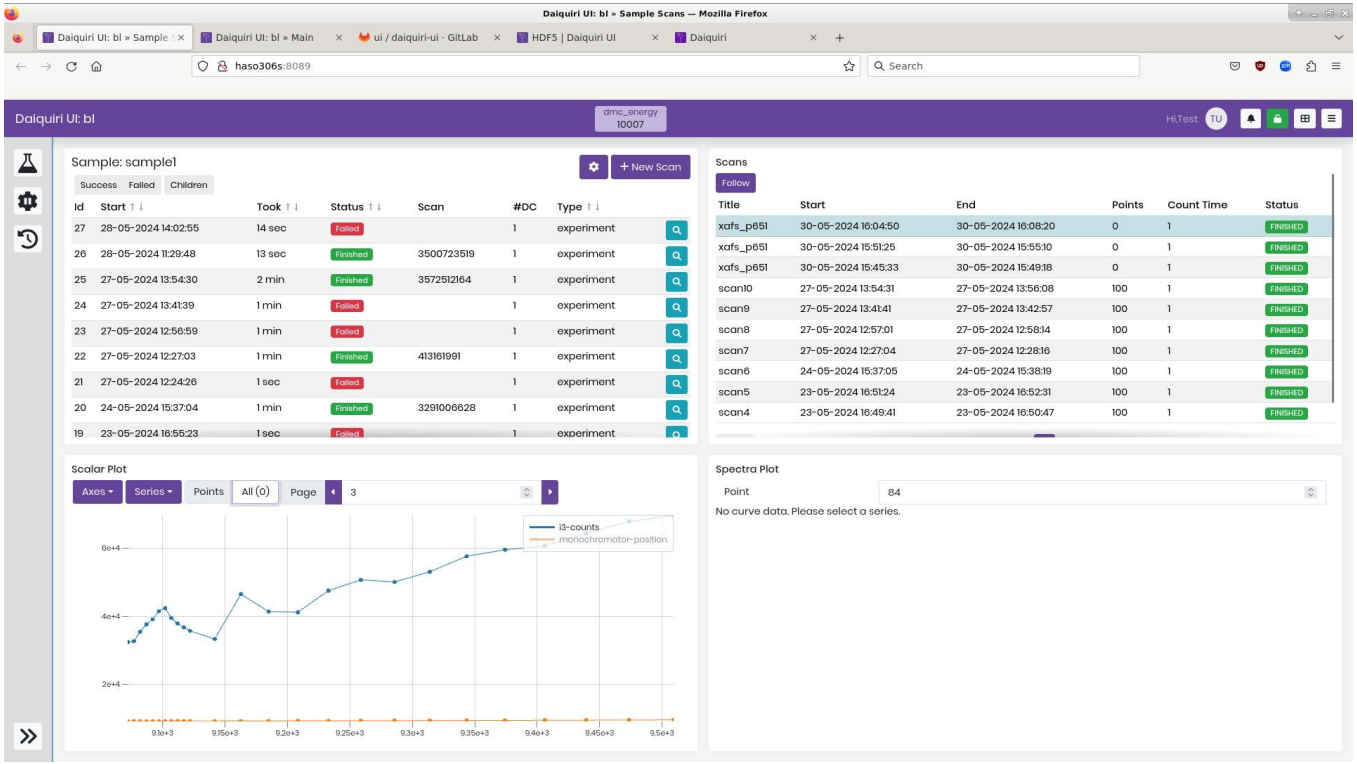
(CURRENTLY UNDER REVIEW)

IN COLLABORATION WITH HZB WE ARE PROTOTYPING NEW METHODS OF SERIALIZING FILE STRUCTURES



https://codebase.helmholtz.cloud/rock-it/wp2/structured_metadata

INTERFACING THE QUEUESERVER API WITH BLISSDATA ENABLES GRAPHICAL CONTROL VIA DAIQUIRI



Thank you for your attention!

Devin Burke (DESY / WP2)
devin.burke@desy.de

Argonne APS Data Acquisition Facility Updates

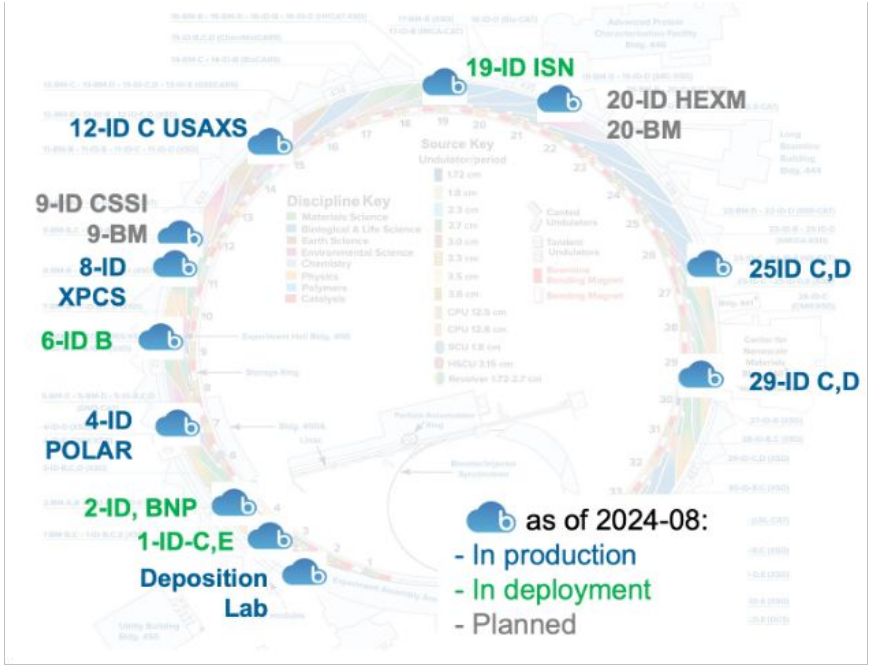
Eric Codrea

Who is attending on behalf of the APS?

- Eric Codrea
- Fanny Rodolakis
- Hannah Parraga
- Nicholas Schwarz

What is currently going on at Argonne?

- More deployment (We are in an installation phase)
- Copier template (testing and rollout)
- APStools:
  - APSu device support continues to grow
- HKLPY
  - Common diffractometer geometries are supported. A local community is forming.
  - Scan in real- or reciprocal-space, save & restore orientation (UB matrix)
  - Easy to create simulations of any supported diffractometer geometry. Some are pre-built.



What we are looking to do?

- We are not using writable tiled servers
  - As a result -> Databroker 1.2.5
- Begin Using Kafka
- Ophyd Async
- HKLPY
  - Working with Hkl/Soleil developer for new diffractometer geometries. Hklpy & hklpy2 use same backend library.
  - Be able to choose backend library from a list. Currently only Hkl/Soleil

What we are eager to hear more about?

- Happi Integration
- Deployment Methodology
- Generic Visualization Tools for Beamline Staff
- User-facing tools for Beamline Staff

ACKNOWLEDGEMENTS

Bluesky Contributors that weren’t able to attend:

- Peter Jemian
- Mark Wolfman

HKLPY Contributors:

- Peter Jemian
- Ken Lauer
- Max Rakitin
- Padraic Shafer

BESSY II - Bluesky Status Update

HZB Helmholtz Zentrum Berlin

This week @ NOBUGS 2024:

- **Luca Porzio**: Beamline Control System Engineer
- **William Smith**: Beamline Control System Engineer
- **Sonal R. Patel**: Data Steward
- **Peter Wegmann**: Sample Environment Control System

Status:
- **8 Beamlines / 9 experiments** running Bluesky
- **4 more beamlines** planned for this year

- Bluesky support for *ROCK-IT* Project (https://www.rock-it-project.de/)

Bluesky Components in use:

- **ophyd** for HW Abstraction Layer
- **RunEngine** for plan execution
- **tiled** for data access with connection to *MongoDB*
- **suitcase** for file export
  - CSV Files
  - SPEC Files
- **queueserver** for plan management

Custom integrations:
- **HAPPI**: device instantiation and indexing (https://pcdshub.github.io/happi/v2.5.0/)
- **PyQT GUI** for *queueserver*
- **Containerised** environment

Future Developments and interests:
- **queueserver** improvements
- **ophyd-async** for HW triggered scanning
- **data access** best practices
- Automatic data export to **NeXus**
- **Daiquiri** for web UI (https://ui.gitlab-pages.esrf.fr/daiquiri/)

Bluesky at Advanced Light Source

Bluesky Satellite Meeting

NOBUGS 2024

Seij De Leon

Computer Systems Engineer

September 23rd, 2024

ALS Overview

-3rd gen synchrotron

-40 beamlines (28 run by ALS)

-230+ staff

-7+ Computing Staff

-7+ Beamline Controls Staff

-ALS-Upgrade (Mid 2026)

- Mixture of LabVIEW, EPICS/Bluesky, others.

ALS Staff @ NoBUGS 2024

Dylan McReynolds

Computing Systems Engineer

Seij De Leon

Computing Systems Engineer

Bluesky Data Access

COSMIC Scattering (7.0.1.1)

Databroker 

NeXus

Tender SAXS / FastXAS (5.3.1)

Databroker

AMBER (6.0.1)

MLExchange

Tiled

Interested Future Bluesky Developments

- Async Tiled Client
- Dedicated Zarr Endpoint in Tiled
- Expanded Tiled Web Client
- Dedicated Javascript/Typescript Tiled Client
- Built-in downsampling in Tiled

Diamond Light Source

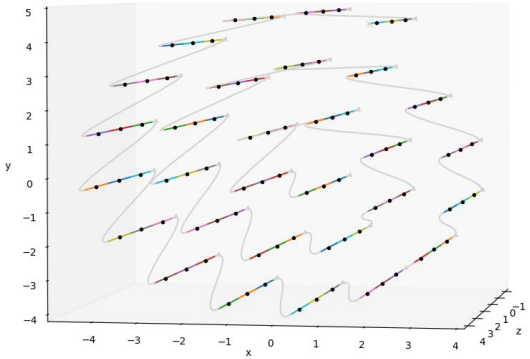
Data Acquisition

Diamond

- UK national synchrotron
- Storage ring upgrade: 2027-29
- Flagship beamline programme: 2028-2030
- Software modernisation programme: 2022-2030
- Bluesky Technical Steering Committee/Board of Governance

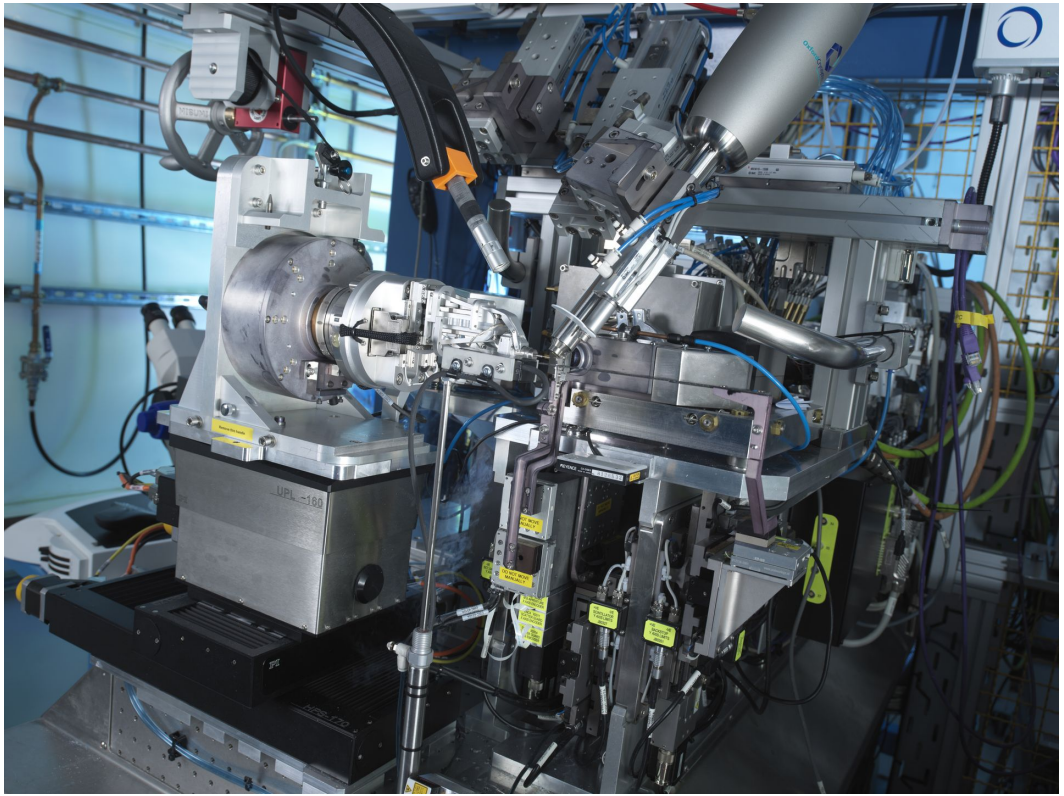
Flyscanning

- Flagship beamline: SWIFT
- Already in use at most beamlines
- https://github.com/DiamondLightSource/pymalcolm
- More flexible system required for long-term



Unattended Data Collection

- Flagship Beamline: K04
- ~75 seconds per sample
- Executing sub-plans in parallel



Flexible Experiment Procedures

- Flagship Beamline: CSXID
- Editable python (plans)
- Adaptive scanning