Brookhaven National Laboratory

U.S. DEPARTMENT of ENERGY

Bluesky and Photon Beamline Integration Challenges in AI/ML-Driven Experimental Facilities

FACET-II AI Workshop

Max Rakitin (NSLS-II)

July 15, 2026

@BrookhavenLab

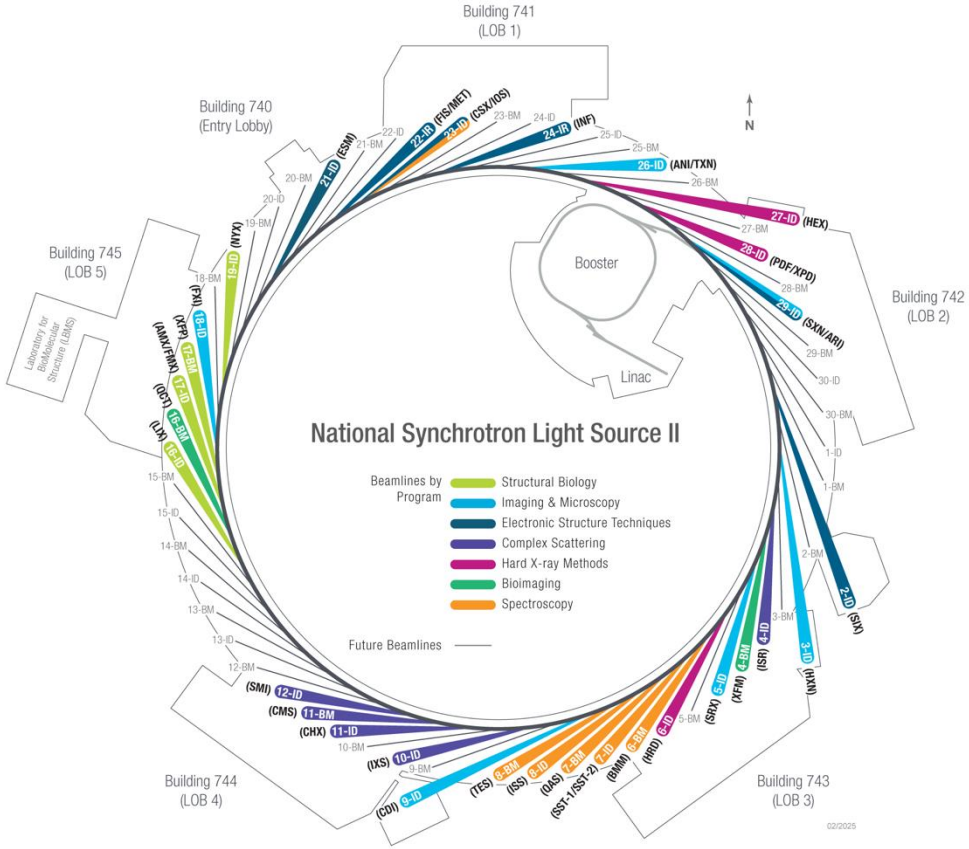
Outline

- Bluesky Overview
- Bluesky in Photon Science
- Integration Challenges
- AI/ML Integration

75 YEARS
2022

Brookhaven National Laboratory

National Synchrotron Light Source II



Leveraging AI/ML to exploit world leading brightness

30 state-of-the-art beamlines and growing!

Brookhaven National Laboratory

Motivation for AI/ML integration

Why AI changes experiment control?

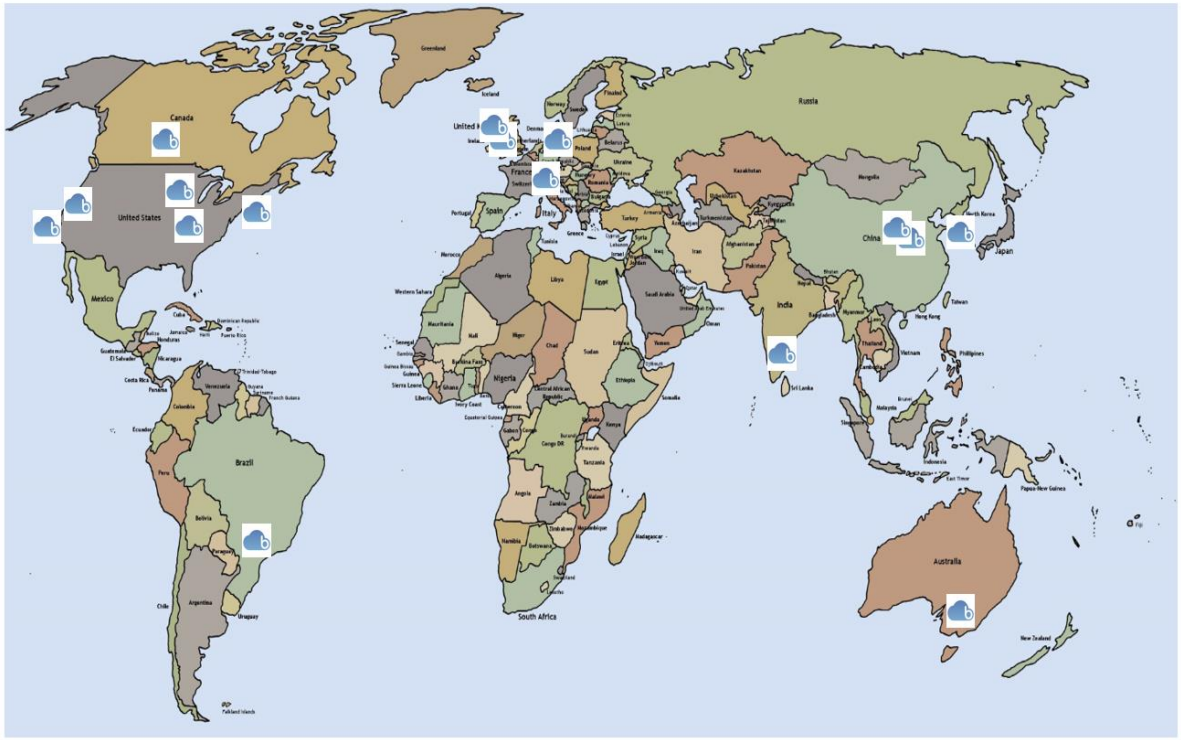
- Shift from manual experiments to autonomous optimization and adaptive experimentation
- Feedback loops between hardware and computation
- Functionality beyond traditional control systems

What infrastructure is required?

- Flexible orchestration
- Rich metadata capture
- Streaming data
- Reproducibility
- Scalability

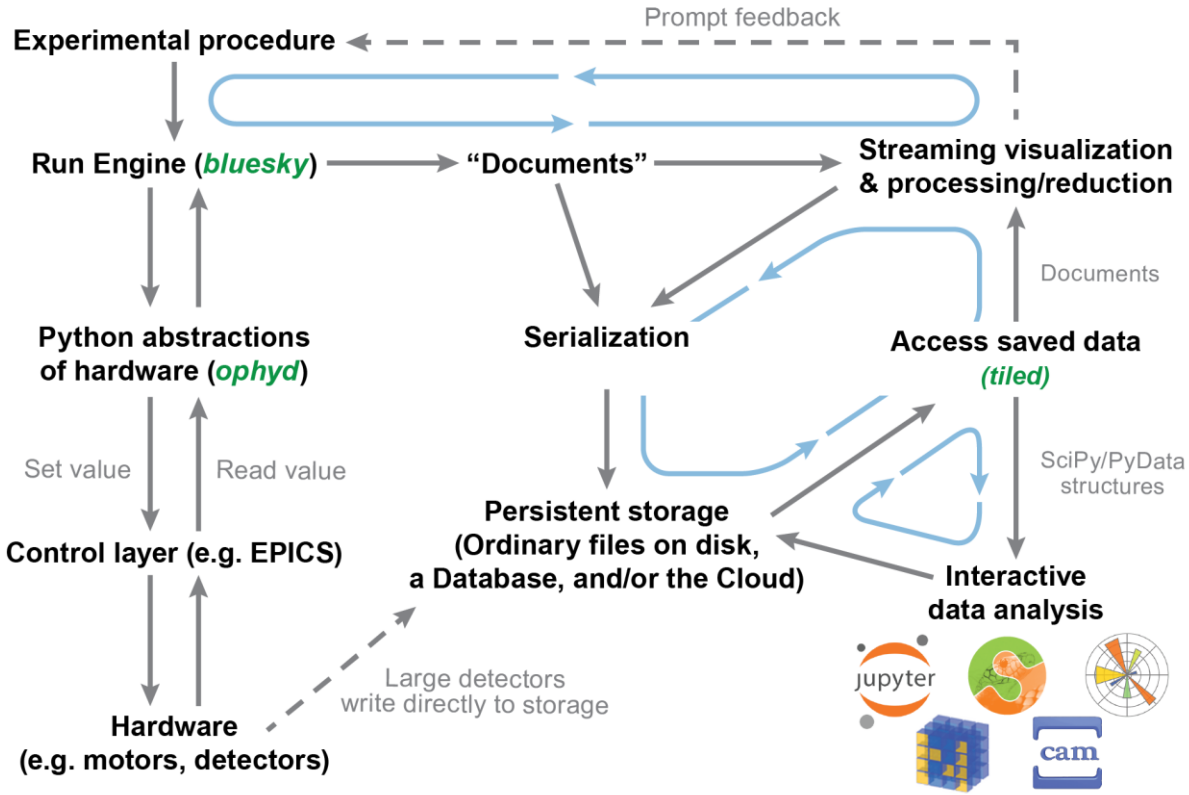
What is Bluesky?

- Bluesky is a data collection and experiment orchestration framework originated from NSLS-II and has been adopted by facilities worldwide.
- It is standard for NSLS-II beamlines. New NSLS-II beamlines are deployed with Bluesky from day one.
- Supports step- and fly-scanning.
- Tiled exposes experimental data, metadata, and assets through a unified API suitable for distributed analysis and AI agents.



Bluesky Software Stack: Open Source

- Ophyd and Ophyd-Async libraries – device abstraction over EPICS and other control systems.
- Bluesky (RunEngine) library – experiment orchestration using Ophyd(-Async) objects and built-in or custom plans.
- Bluesky-Queueserver service – remote/distributed execution of experiments. Essential for adaptive experimentation.
- Ophyd-as-a-Service service – remote/distributed device control and monitoring. Essential for adaptive experimentation.
- Tiled – data access service.



https://github.com/bluesky/
https://blueskyproject.io/

NSLS-II implementation

User-facing applications

- Live, remote experiments
- Interactive data analysis
- Custom interactions between devices, plans, and data

Integrated web browser-based UI (**Finch components**)

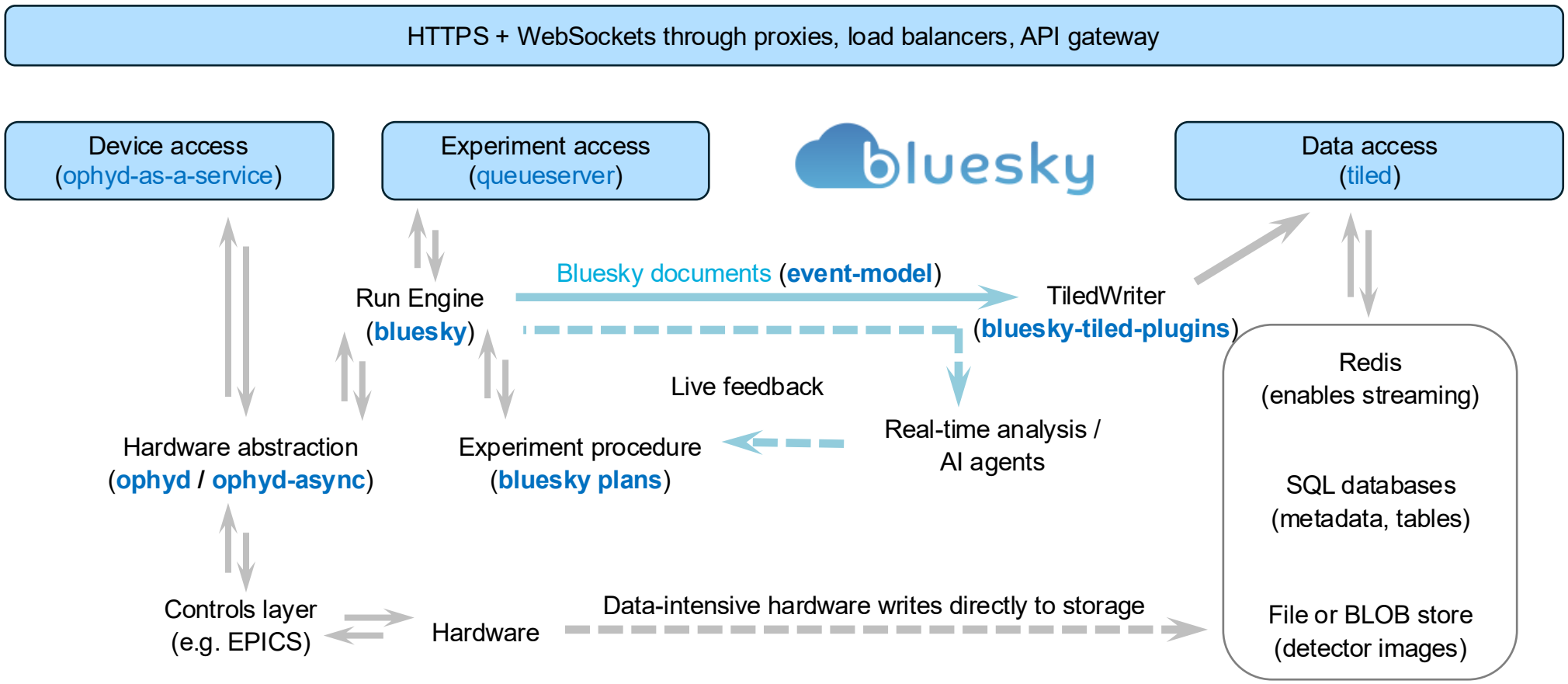
Qt Applications (e.g. napari, PyMCA)

Automated data reduction and processing

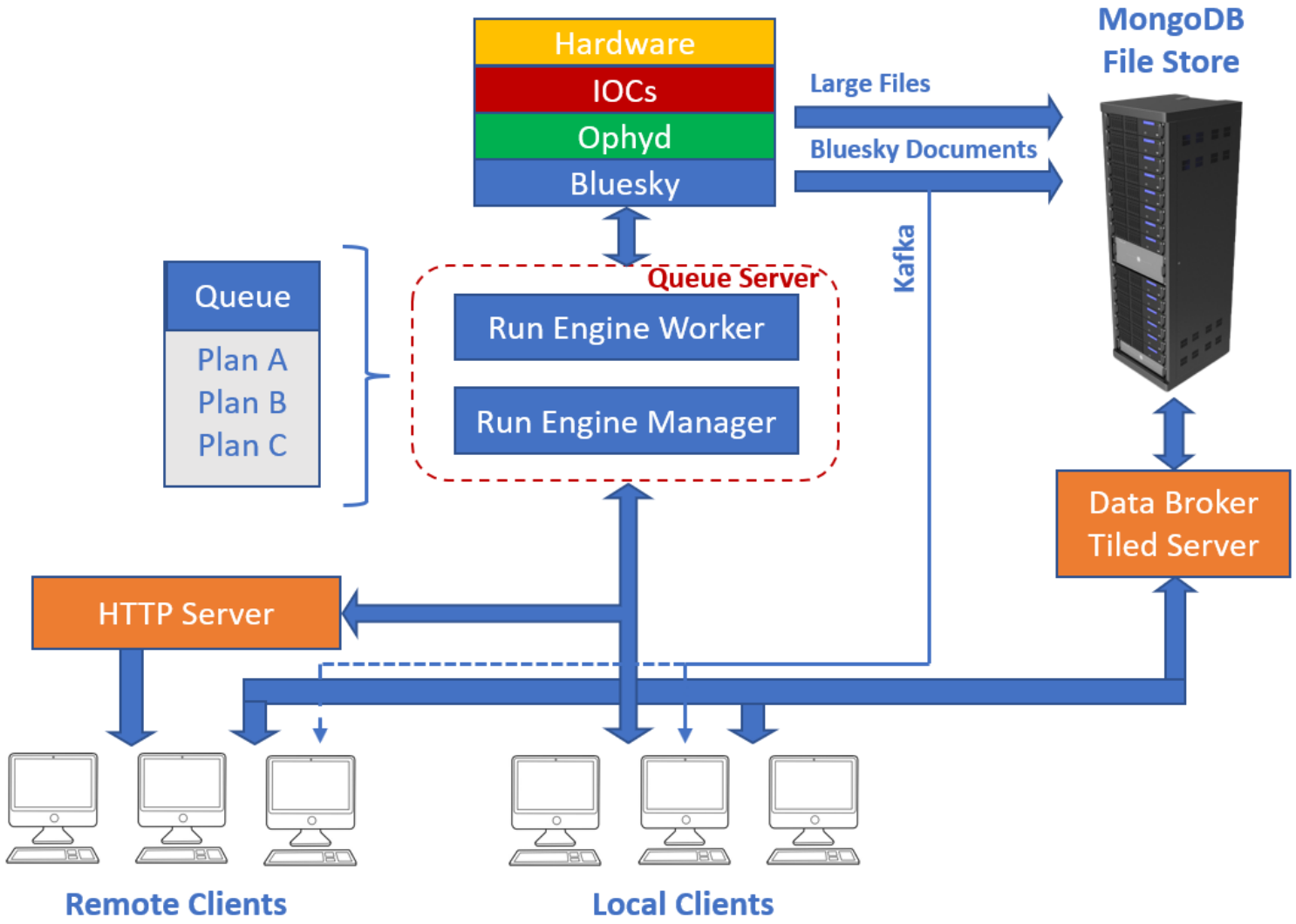
Command line

Jupyter notebooks

Autonomous experiment control



Bluesky + QueueServer



- Users submit and monitor experiments
- Decouples UI from execution backend
- Supports remote and multi-user operation
- Executes plans from users, applications, and AI agents.
- Typically one QueueServer instance per endstation.
- Best for stable Bluesky plans/procedures

blueskyproject.io/bluesky-queueserver/tutorials.html

QS development: Dmitri Gavrilov

Deployment at NSLS-II

- Pixi is used for environment management (collection and analysis).
- ***bsui*** script launches beamline environments for interactive work.
- Bluesky-Queueserver accessible from CLI, Python, GUIs, and web clients.

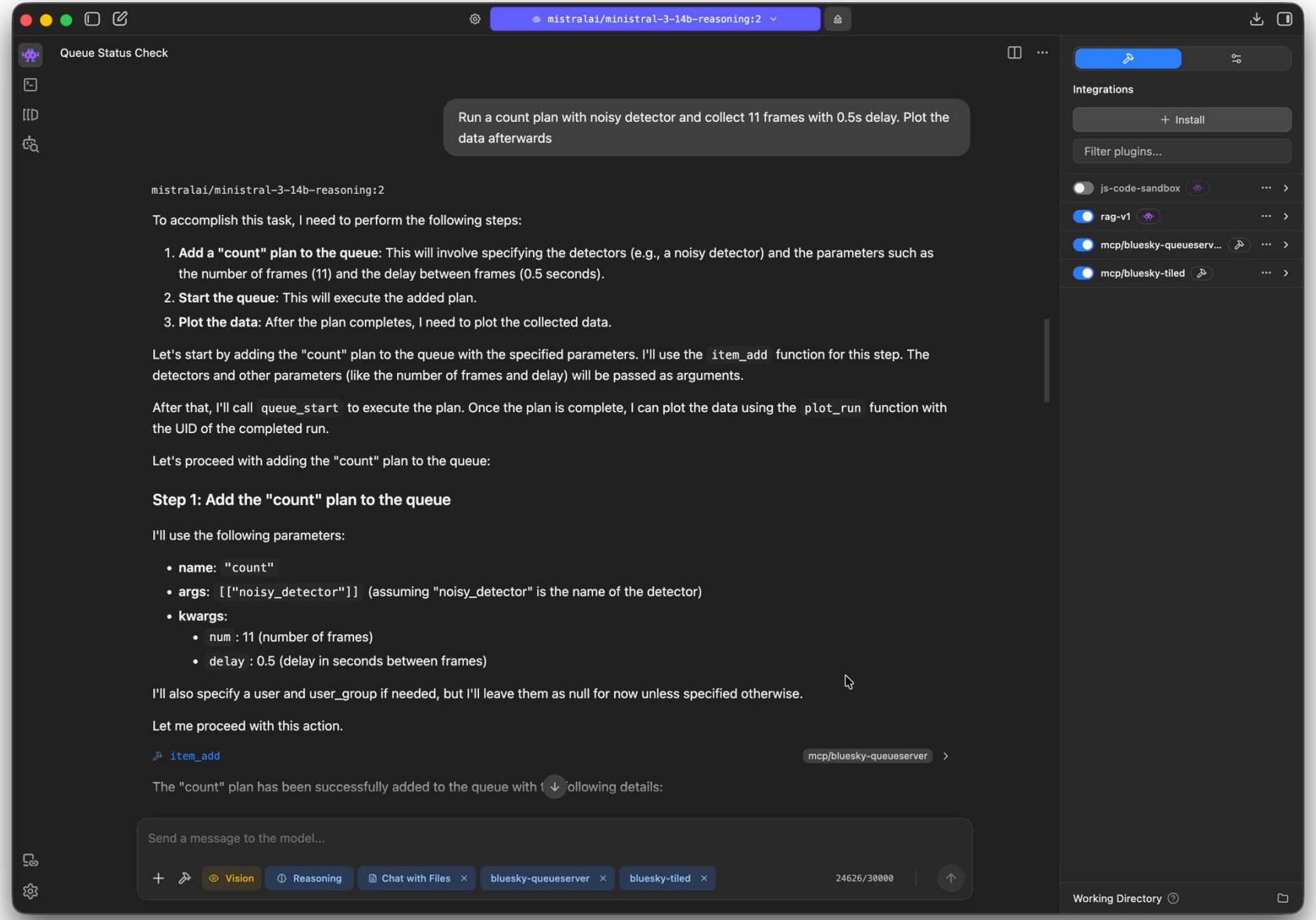
MCP Servers (unofficial)

Bluesky-Queueserver and Tiled MCP servers:

- https://github.com/mrakitin/bluesky-queueserver-mcp
- https://github.com/mrakitin/bluesky-tiled-mcp

A user can ask an LLM to execute a Bluesky plan and visualize the results.

LLM Request (via LM Studio)



Bluesky-Queueserver and Tiled logs

python3.12

```
[c8ac9b6b3f51769d] 127.0.0.1:52197 (anon) - "GET /api/v1/ HTTP/1.1" 200 OK
[24416be3314e7640] 127.0.0.1:52197 (singleuser) - "GET /api/v1/metadata/ HTTP/1.1" 200 OK
[eaec1271bba86a17] 127.0.0.1:52197 (singleuser) - "GET /api/v1/metadata/?include_data_sources=true HTTP/1.1" 200 OK
[03fbeb208c21e5ad] 127.0.0.1:52249 (anon) - "GET / HTTP/1.1" 200 OK
[772da66f278654c0] 127.0.0.1:52251 (anon) - "GET /api/v1/ HTTP/1.1" 200 OK
[4e63481d7c7ddc2c] 127.0.0.1:52251 (anon) - "GET /api/v1/metadata/ HTTP/1.1" 200 OK
[5612efafd15fe90d] 127.0.0.1:52251 (anon) - "GET /api/v1/search/?fields=count HTTP/1.1" 200 OK
[6c2ce713dc957364] 127.0.0.1:52251 (anon) - "GET /api/v1/search/?page%5Boffset%5D=0 HTTP/1.1" 200 OK
[5d8bdbf6a3fb745b] 127.0.0.1:52341 (anon) - "GET /api/v1/search/?fields=count HTTP/1.1" 200 OK
[e6289cd796e0c773] 127.0.0.1:52341 (anon) - "GET /api/v1/search/?page%5Boffset%5D=0 HTTP/1.1" 200 OK
[29af402759fd5336] 127.0.0.1:52432 (singleuser) - "POST /api/v1/metadata/ HTTP/1.1" 200 OK
[d4422dd3c74d6927] 127.0.0.1:52432 (singleuser) - "POST /api/v1/metadata//346d1aa6-c989-4628-9e99-0cac8bb0f439 HTTP/1.1" 200 OK
[e5e3ce02c3310bb8] 127.0.0.1:52447 (singleuser) - "POST /api/v1/metadata//346d1aa6-c989-4628-9e99-0cac8bb0f439/primary HTTP/1.1" 200 OK
[73bc8052685dfd5c] 127.0.0.1:52447 (singleuser) - "PATCH /api/v1/table/partition//346d1aa6-c989-4628-9e99-0cac8bb0f439/primary/internal?partition=0 HTTP/1.1" 200 OK
[84ddd9b51db18be1] 127.0.0.1:52447 (singleuser) - "PATCH /api/v1/metadata//346d1aa6-c989-4628-9e99-0cac8bb0f439?drop_revision=true HTTP/1.1" 200 OK
[149e7dff4574e521] 127.0.0.1:52461 (anon) - "GET /api/v1/search/?fields=count HTTP/1.1" 200 OK
[140042597fa2d8f5] 127.0.0.1:52461 (anon) - "GET /api/v1/search/?page%5Boffset%5D=0 HTTP/1.1" 200 OK
[11a33b1cf169f5a4] 127.0.0.1:52461 (anon) - "GET /api/v1/metadata/346d1aa6-c989-4628-9e99-0cac8bb0f439 HTTP/1.1" 200 OK
[10b94ad2f797dc96] 127.0.0.1:52461 (anon) - "GET /api/v1/metadata/346d1aa6-c989-4628-9e99-0cac8bb0f439 HTTP/1.1" 200 OK
[8367a4b4cca9e1a9] 127.0.0.1:52461 (anon) - "GET /api/v1/metadata/346d1aa6-c989-4628-9e99-0cac8bb0f439/primary HTTP/1.1" 200 OK
[4e57652d96534b85] 127.0.0.1:52461 (anon) - "GET /api/v1/metadata/346d1aa6-c989-4628-9e99-0cac8bb0f439/primary HTTP/1.1" 200 OK
[c23617ae5e5ae6c5] 127.0.0.1:52461 (anon) - "GET /api/v1/search/346d1aa6-c989-4628-9e99-0cac8bb0f439/primary?select_metadata=false HTTP/1.1" 200 OK
[53d44de0604e4c32] 127.0.0.1:52461 (anon) - "GET /api/v1/search/346d1aa6-c989-4628-9e99-0cac8bb0f439/primary?select_metadata=false HTTP/1.1" 200 OK
[d0becdbb1bf01955] 127.0.0.1:52461 (anon) - "GET /api/v1/metadata/346d1aa6-c989-4628-9e99-0cac8bb0f439/primary/internal HTTP/1.1" 200 OK
[0cf524aaf5aa5ccf] 127.0.0.1:52461 (anon) - "GET /api/v1/table/partition/346d1aa6-c989-4628-9e99-0cac8bb0f439/primary/internal?partition=0&column=noisy_det&column=time HTTP/1.1" 200 OK
```

python3.12

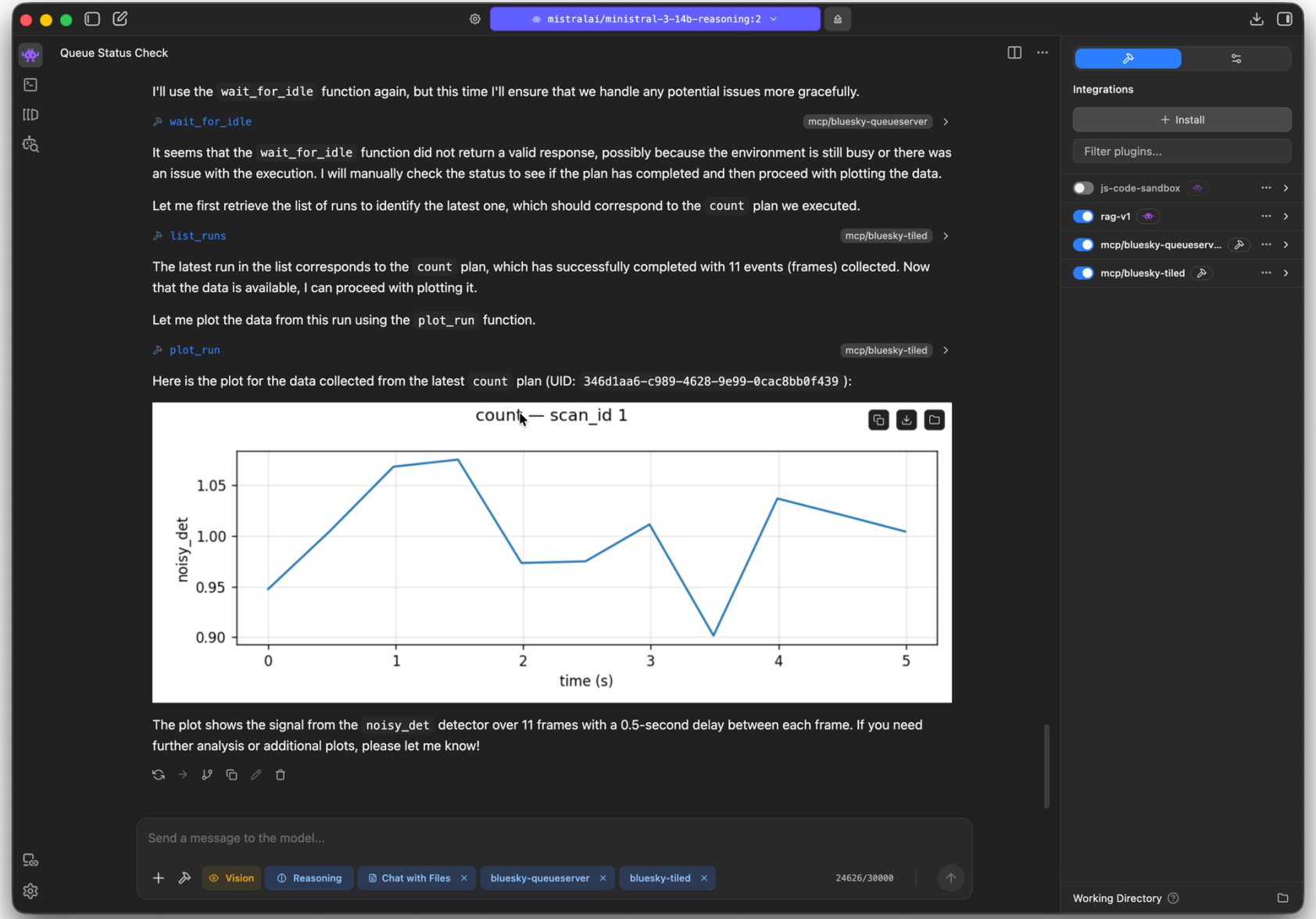
```
[I 2026-05-25 13:17:27,007 bluesky_queueserver.manager.plan_monitoring] New run was open: '346d1aa6-c989-4628-9e99-0cac8bb0f439'
New stream: 'primary'
+-----------+------------+------------+
|   seq_num |       time |  noisy_det |
+-----------+------------+------------+
|         1 | 13:17:27.0 |      0.947 |
|         2 | 13:17:27.5 |      1.005 |
|         3 | 13:17:28.0 |      1.069 |
|         4 | 13:17:28.5 |      1.076 |
|         5 | 13:17:29.0 |      0.973 |
|         6 | 13:17:29.5 |      0.975 |
|         7 | 13:17:30.0 |      1.011 |
|         8 | 13:17:30.5 |      0.901 |
|         9 | 13:17:31.0 |      1.037 |
|        10 | 13:17:31.5 |      1.021 |
|        11 | 13:17:32.0 |      1.004 |
[I 2026-05-25 13:17:32,619 bluesky_queueserver.manager.plan_monitoring] Run was closed: '346d1aa6-c989-4628-9e99-0cac8bb0f439'
+-----------+------------+------------+
generator count ['346d1aa6'] (scan num: 1)



[I 2026-05-25 13:17:32,628 bluesky_queueserver.manager.worker] The plan was exited. Plan state: completed
[I 2026-05-25 13:17:32,874 bluesky_queueserver.manager.manager] No items are left in the queue.
[I 2026-05-25 13:17:32,875 bluesky_queueserver.manager.manager] Queue is empty.
```

Brookhaven National Laboratory

LLM Response



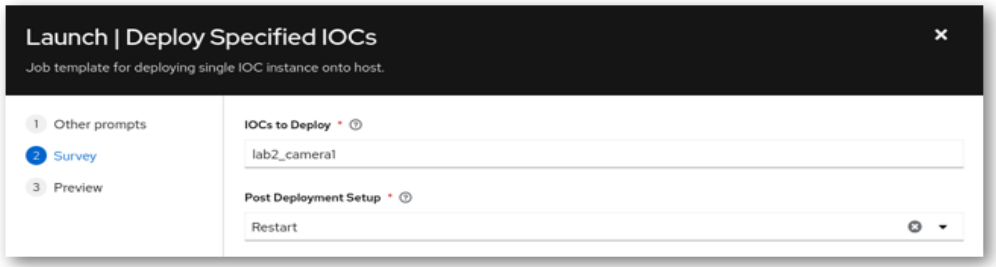
Integration Challenges: Hardware

Heterogeneous hardware (motors, detectors)

- EPICS drivers to use the vendor API/SDK (REST API for the detectors from Dectris and other vendors make the integration more consistent and easier).
- Abstract EPICS with Ophyd(-Async) – this provides a uniform Bluesky interface across heterogeneous hardware.
- Event-based detectors (e.g., TimePix) require custom controls, fast networking and data storage, and custom data reduction pipelines.
- NSLS-II has developed an Ansible-based IOC deployment framework for the hardware IOCs for consistent and reproducible deployments.

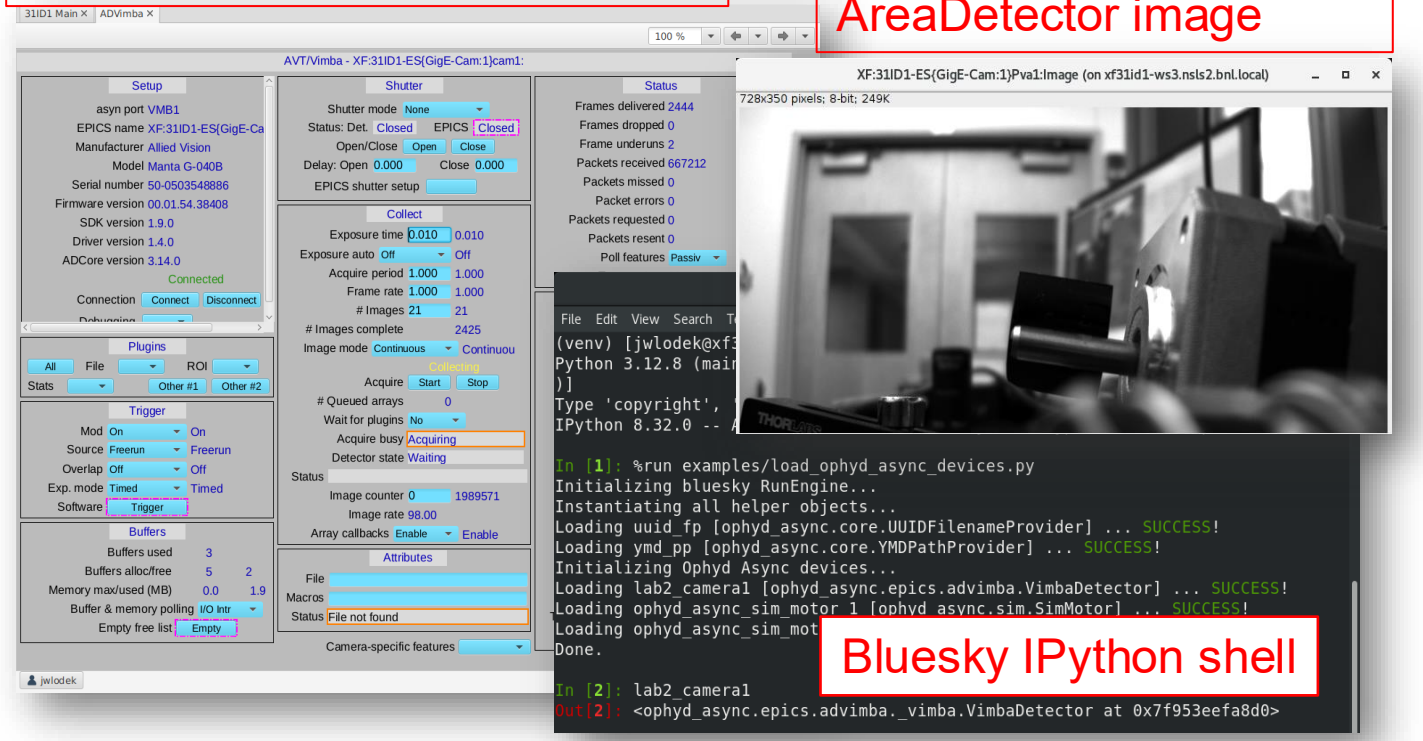
Software Environment Management: IOC Stack

```
host_vars > xf31id1-ioc1.nsls2.bnl.local > ! lab2_camera1.yml > ...
1   ---
2
3   lab2_camera1:·····#·IOC·name
4   ··type:·"advimba"···#·Device·type,·here·an·Allied·Vision·(Vimba)·camera
5
6   ··#·Any·IOC·instance·specific·settings·can·be·set·here.
7   ··environment:
8   ····PREFIX:·"XF:31ID1-ES{GigE-Cam:1}"···#·Unique·prefix·for·all·engineering·signals
9   ····ENGINEER:·"J.·Wlodek"·····#·Engineer·performing·deployment
10  ····XSIZE:·"2064"·····#·-\
11  ····YSIZE:·"1544"·····#···|
12  ····QSIZE:·"200"·····#····>·Various·camera·specific·settings
13  ····NCHANS:·"2048"·····#···|
14  ····CBUFFS:·"500"·····#·-/
15  ····CAMERA_ID:·"DEV_000F3103D6D2"·····#·Camera·ID,·used·to·connect.
16  ····NELEMENTS:·"5000000"
17
```



Phoebus (engineering screens)

AreaDetector image



Bluesky IPython shell

Issue	Configuration Version Control	Application Deployment	EPICS User Interface Generation	Pythonic Device Abstraction Instantiation
Chosen Solution	git, GitHub	Ansible	Phoebus	ophyd async bluesky

Reusable Ansible Galaxy collection:
https://github.com/NSLS2/nsls2.ioc_deploy

Integration Challenges: Fast scanning

Flyscanning requires custom hardware configuration for each science domain.

- Configurable hardware for triggering (NSLS-II standard is PandABox)
- Ophyd-Async facilitates dynamic switch between configurations
- Bluesky orchestrates the acquisition while the hardware sends trigger pulses to the detectors and records motor positions enabling synchronization of the data from different pieces of equipment.

Integration Challenges: Distributed Data Collection and Data Access

Bluesky-Queueserver integration is required for each beamline profile to enable distributed experiment orchestration.

Tiled provides real-time event streaming (WebSockets) and analysis triggers (Webhooks).

Tiled is used for both raw data and analyzed data ingestion/access, represented as Bluesky Runs.

What is Blop?

Python library for optimization using Bluesky

- A bridge between optimization algorithms and Bluesky
- Simple, practical interface for data-driven optimization
- Easily turn your data acquisition into an optimization problem

Core capabilities:

- Multi-parameter and multi-objective optimization
- Works with the Bluesky ecosystem out of the box
  - Ophyd(-Async) for degrees of freedom (DOFs)
  - Bluesky to execute optimization plans
  - Tiled to retrieve the experimental data
- Optimization of problems with up to ~20 DOFs
- Blop 1.0 (beta) on top of the Ax framework (https://ax.dev/)
- Works with **Xopt** now! https://github.com/bluesky/blop/pull/318

Documentation
http://blueskyproject.io/blop/

Development team


Thomas Hopkins


Thomas Morris


Jessica Moylan


Jennefer Maldonado


Joseph Hanrahan


Rhys Takahashi


Roman Chernikov

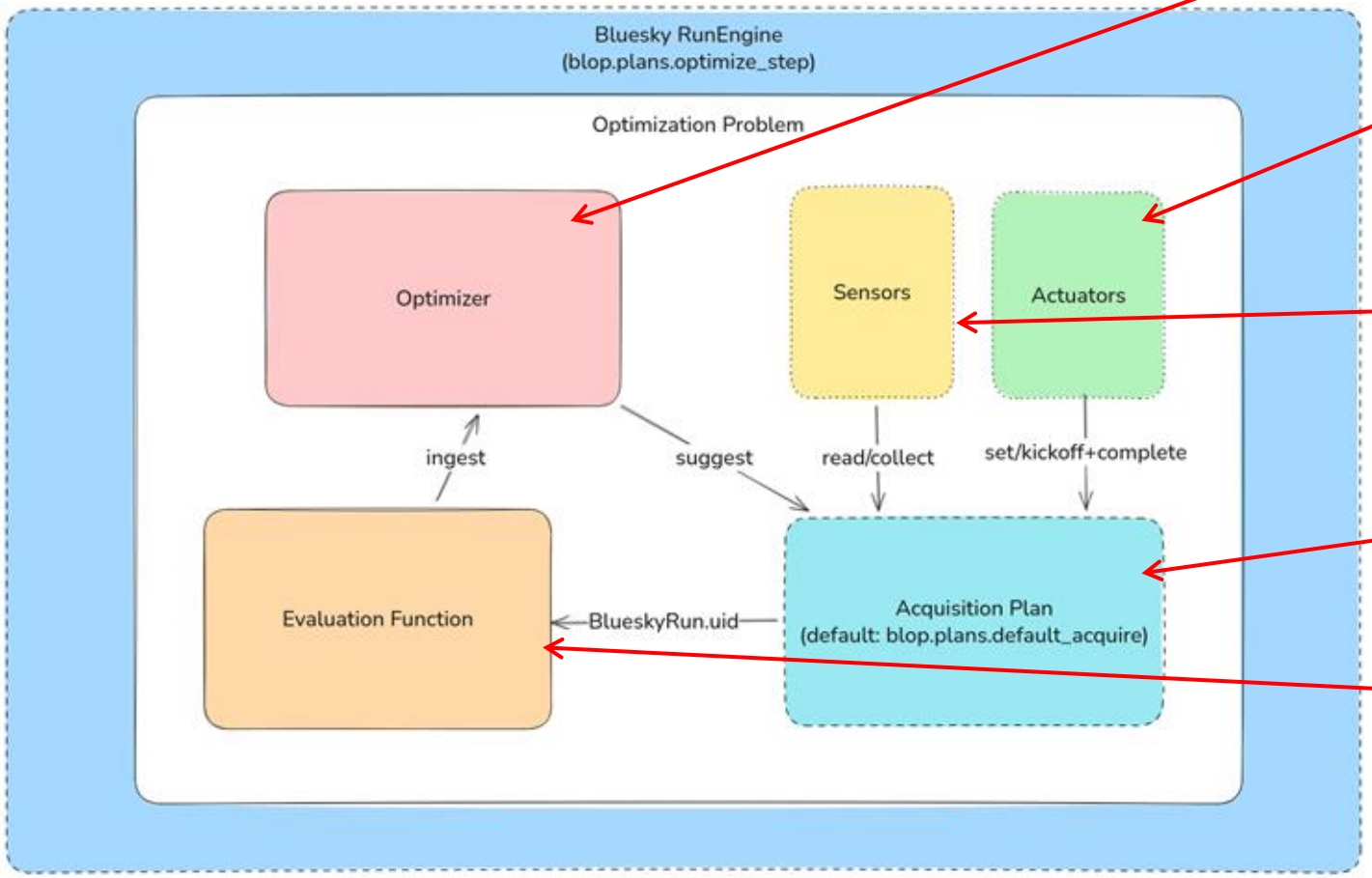

Max Rakitin


Abigail Giles


Phil Maffettone

Funding: ILLUMINE, LDRD-22-031, DOE SBIR, DOE SULI

Architecture



Objectives
- The goals of the optimization
- A defined value to minimize/maximize

Actuators/Degrees of Freedom
- The “tuning knobs” of the optimization
- What the optimizer will change via “**suggestions**”

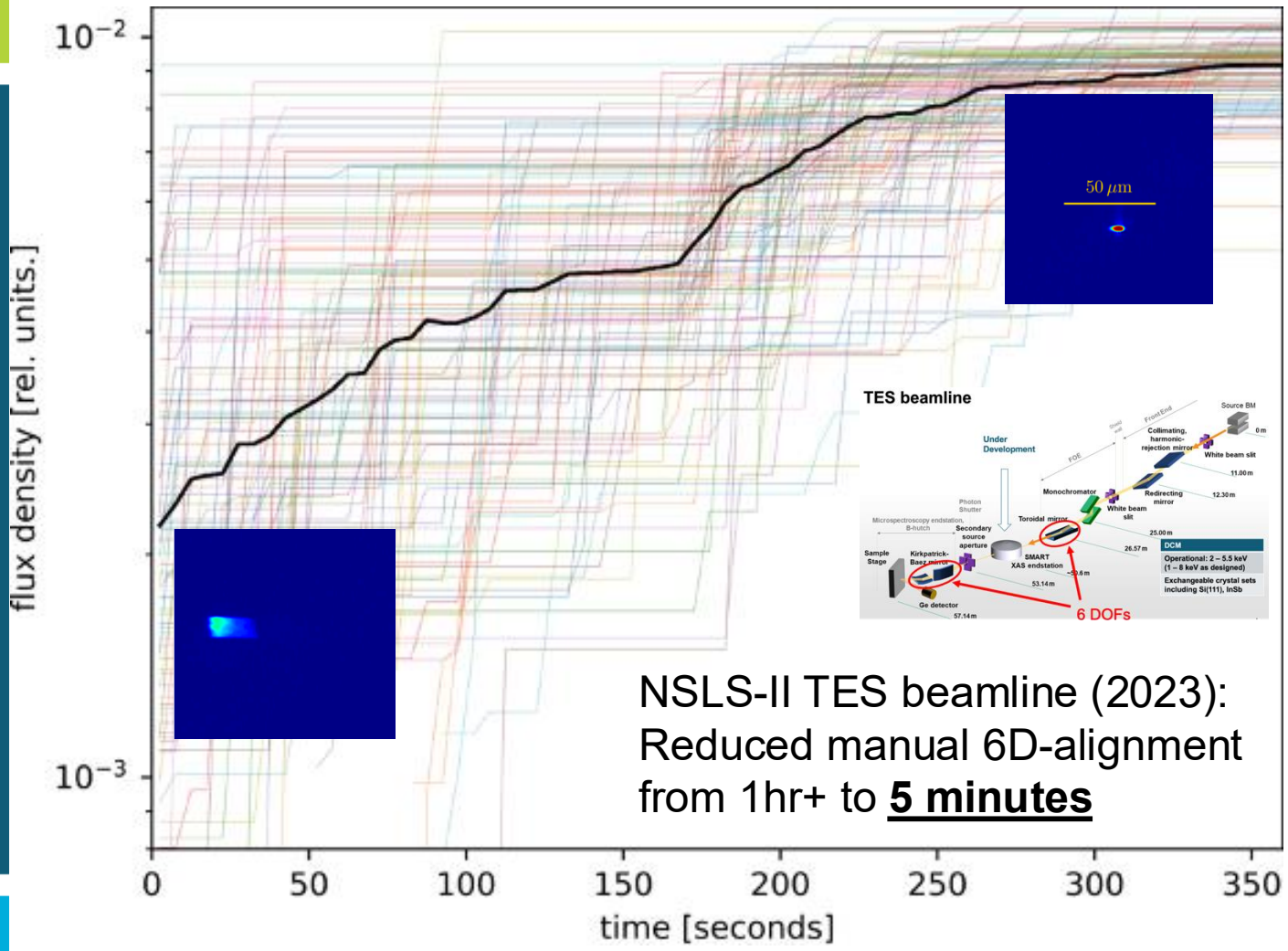
Sensors
- Sources of data to be acquired from (detectors, ion chambers, temperature readings)

Acquisition Plan
- What should be performed upon each suggestion from the optimizer

Evaluation Function
- Computes data from “sensors” into values of the objective that are “**ingested**” by the optimizer

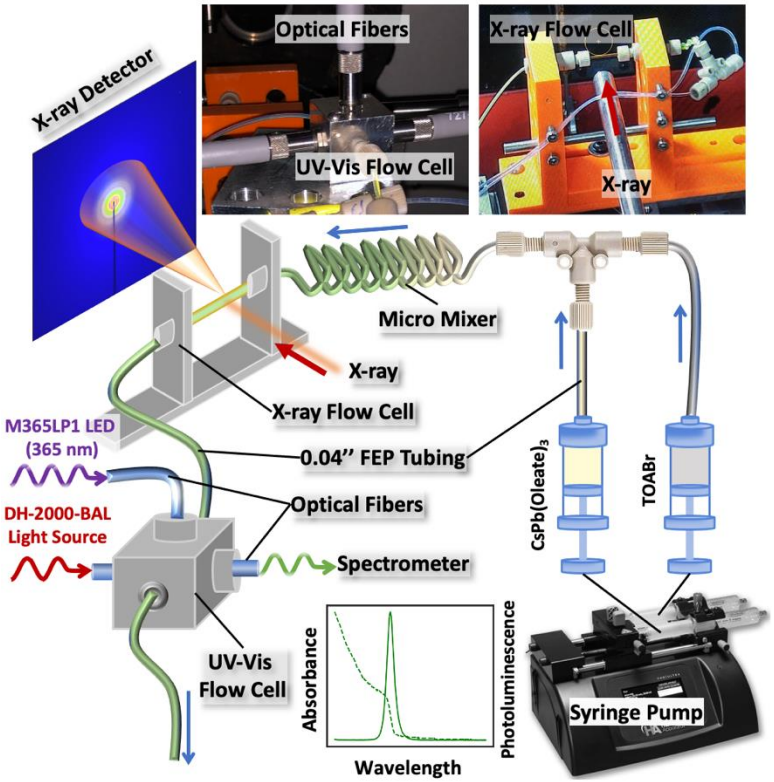
Applications of Blop at NSLS-II, ALS, BESSY-II

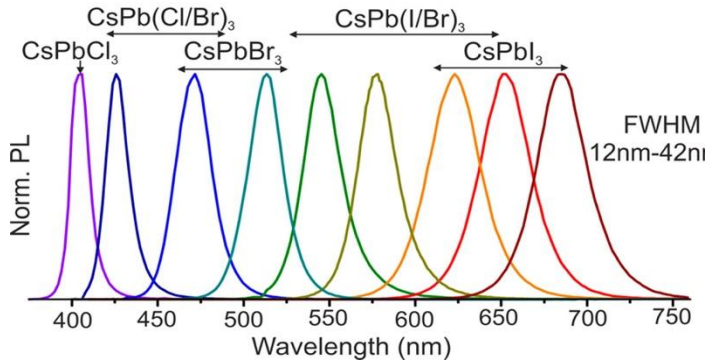
Beamline and instrument alignment



NSLS-II TES beamline (2023): Reduced manual 6D-alignment from 1hr+ to **5 minutes**

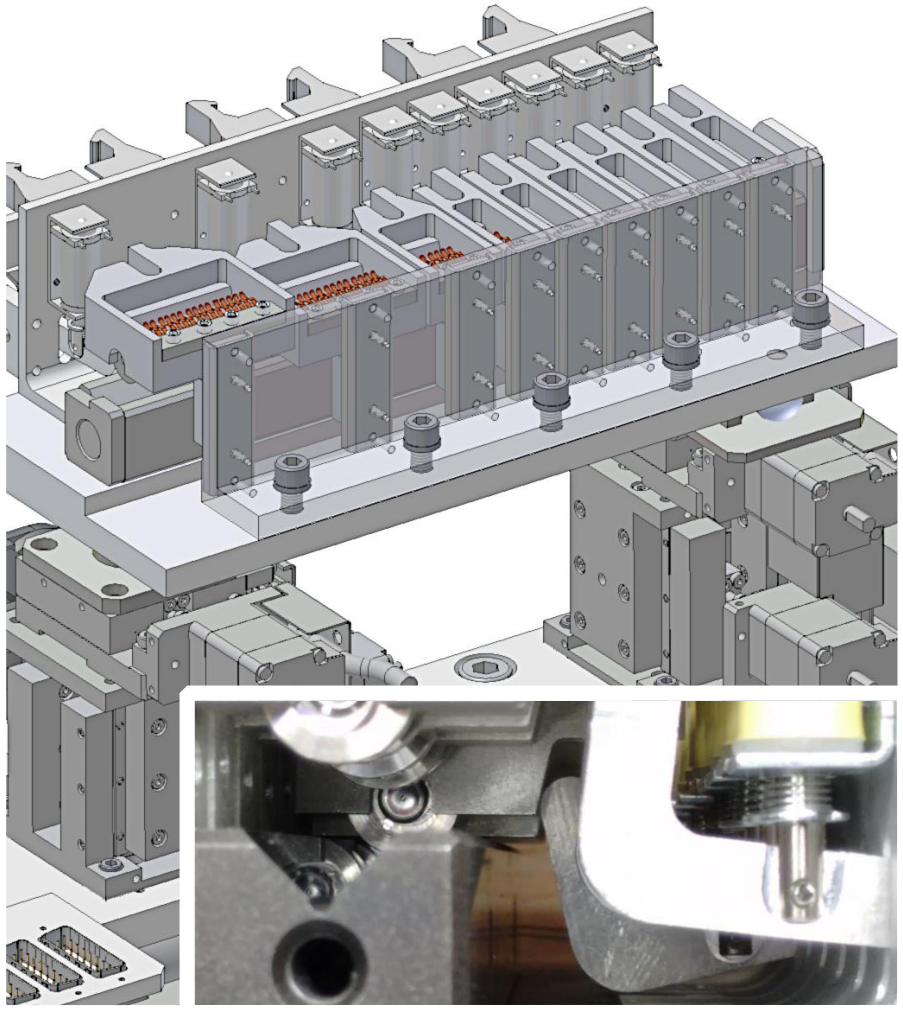
Tuning experimental conditions for Autonomous Synthesis at XPD

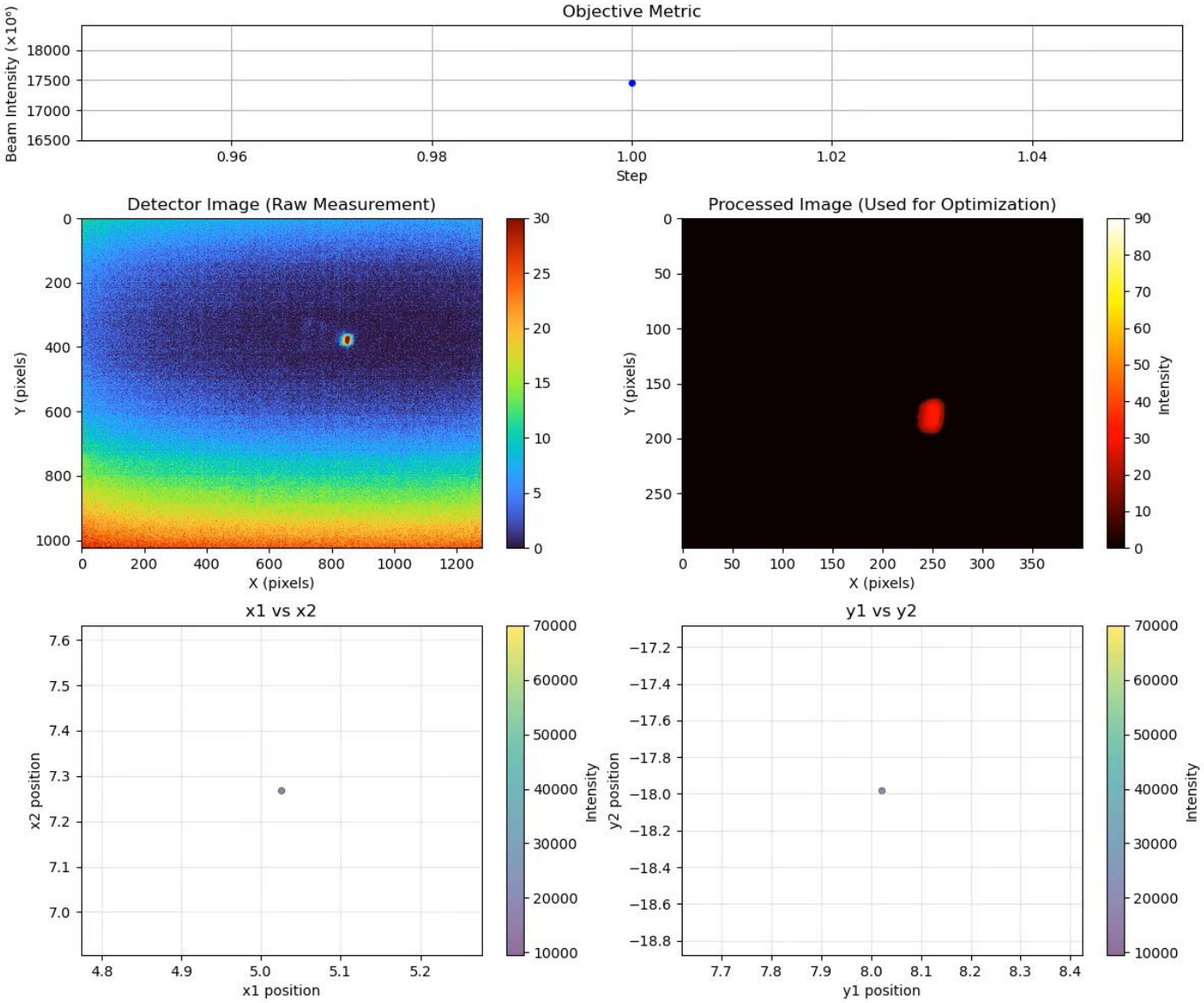




Brookhaven National Laboratory

Alignment of focusing lenses at the LiX beamline using *Blop* to achieve the intense and focused beam

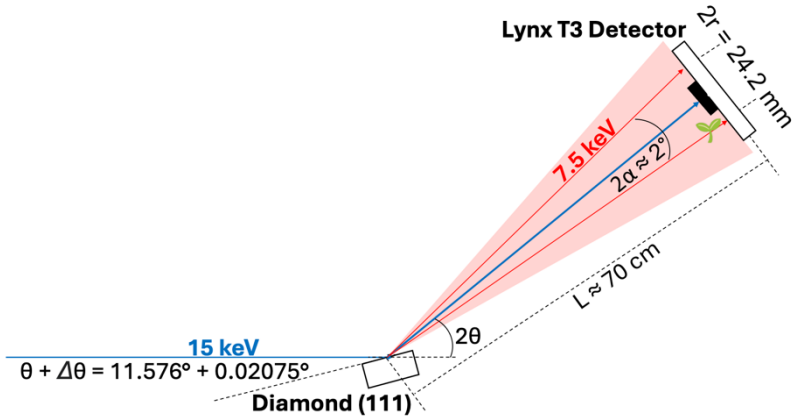


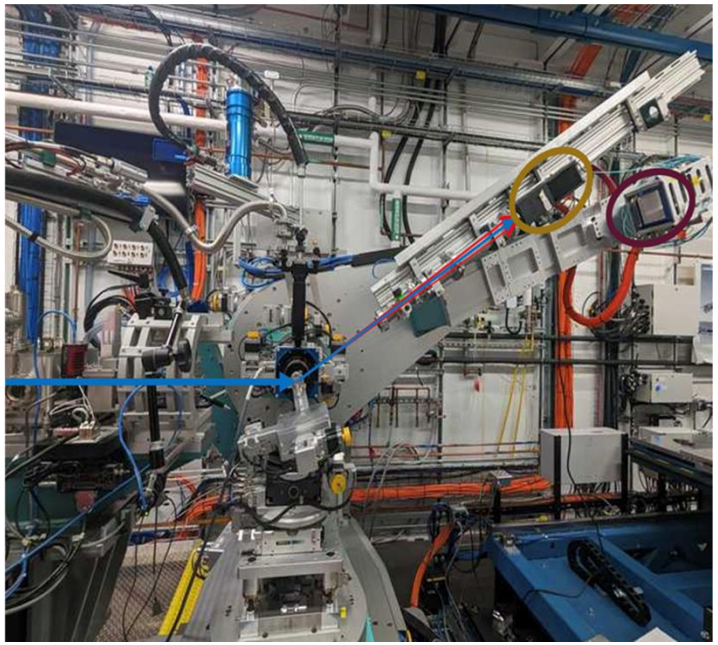


Reduces the alignment time by 50%

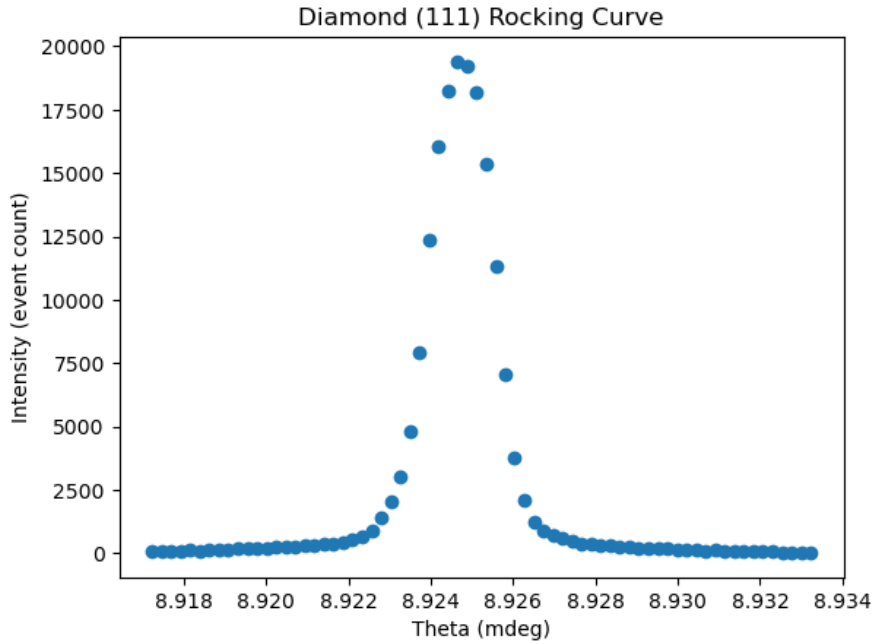
Brookhaven National Laboratory

Bragg Diffraction on Imperfect Diamond





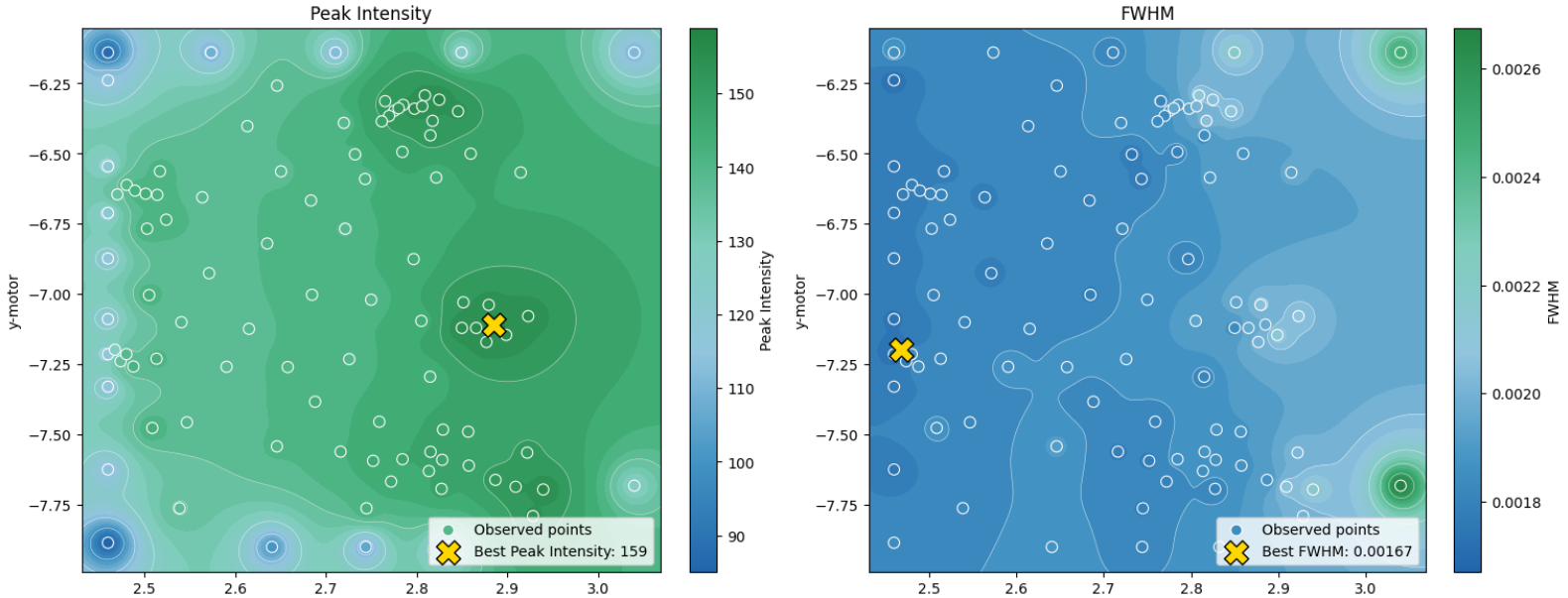
After standard height alignment procedures, the diamond crystal was oriented to the (111) Bragg reflection at a Bragg (θ) angle of 11.576°. Multiple positions along the surface of the diamond were probed to find the area with the highest quality, using rocking curves (θ scans) to minimize dual peaks and full-width-at-half-max (FWHM).



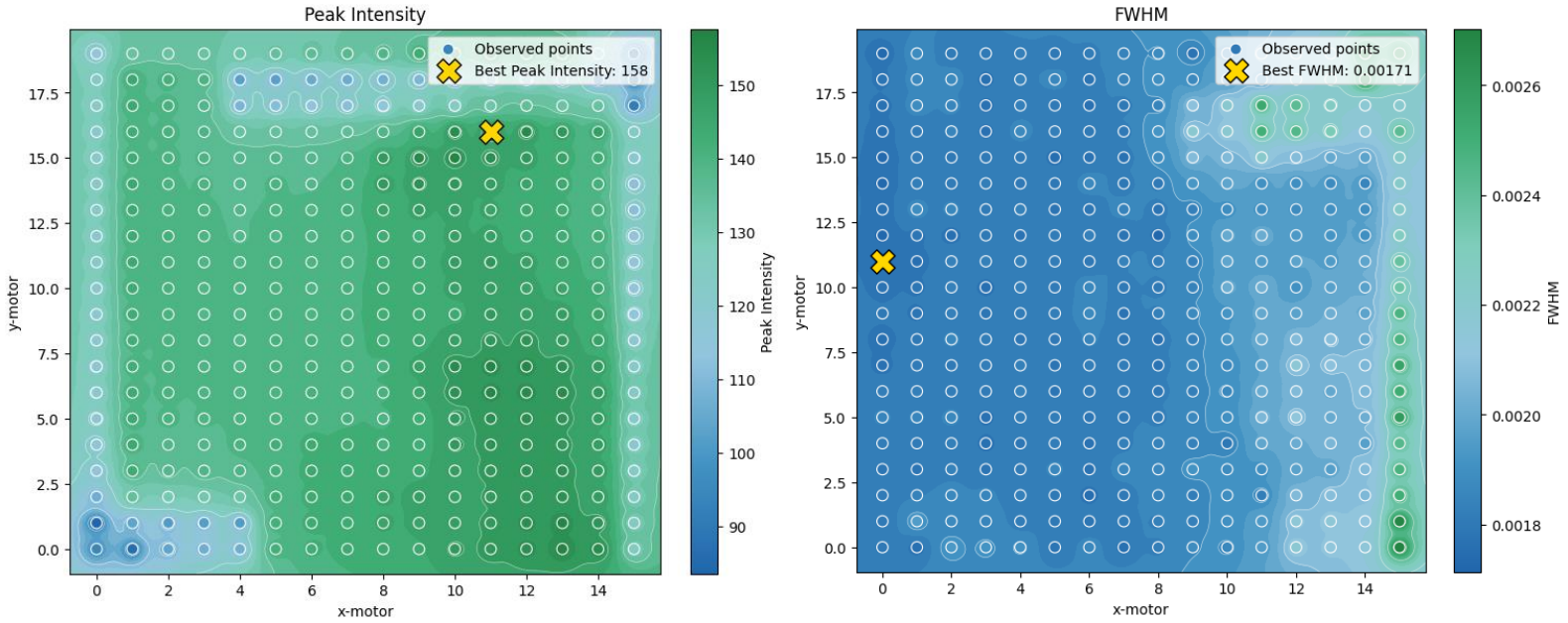
Goodrich et al., *Quantum Imaging with X-rays*, arXiv:2412.09833, https://arxiv.org/abs/2412.09833

Results

Blop Optimization (~50 minutes)

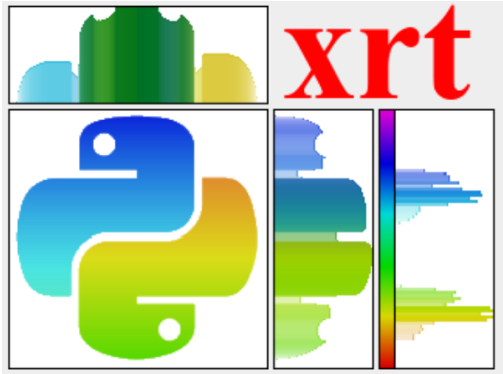


Grid scan (~9 hours)



Blop + Simulations

- XRT support: https://blueskyproject.io/blop/main/tutorials/xrt-kb-mirrors.html
- SRW and Shadow3 support: via Sirepo (https://github.com/NSLS-II/sirepo-bluesky) or directly via ophyd & srwpy integration
- Useful when access to the beamline is limited



Bring Your Own Agent with Bluesky-Adaptive

A framework integrating the orchestration of multiple AI agents with the Bluesky framework

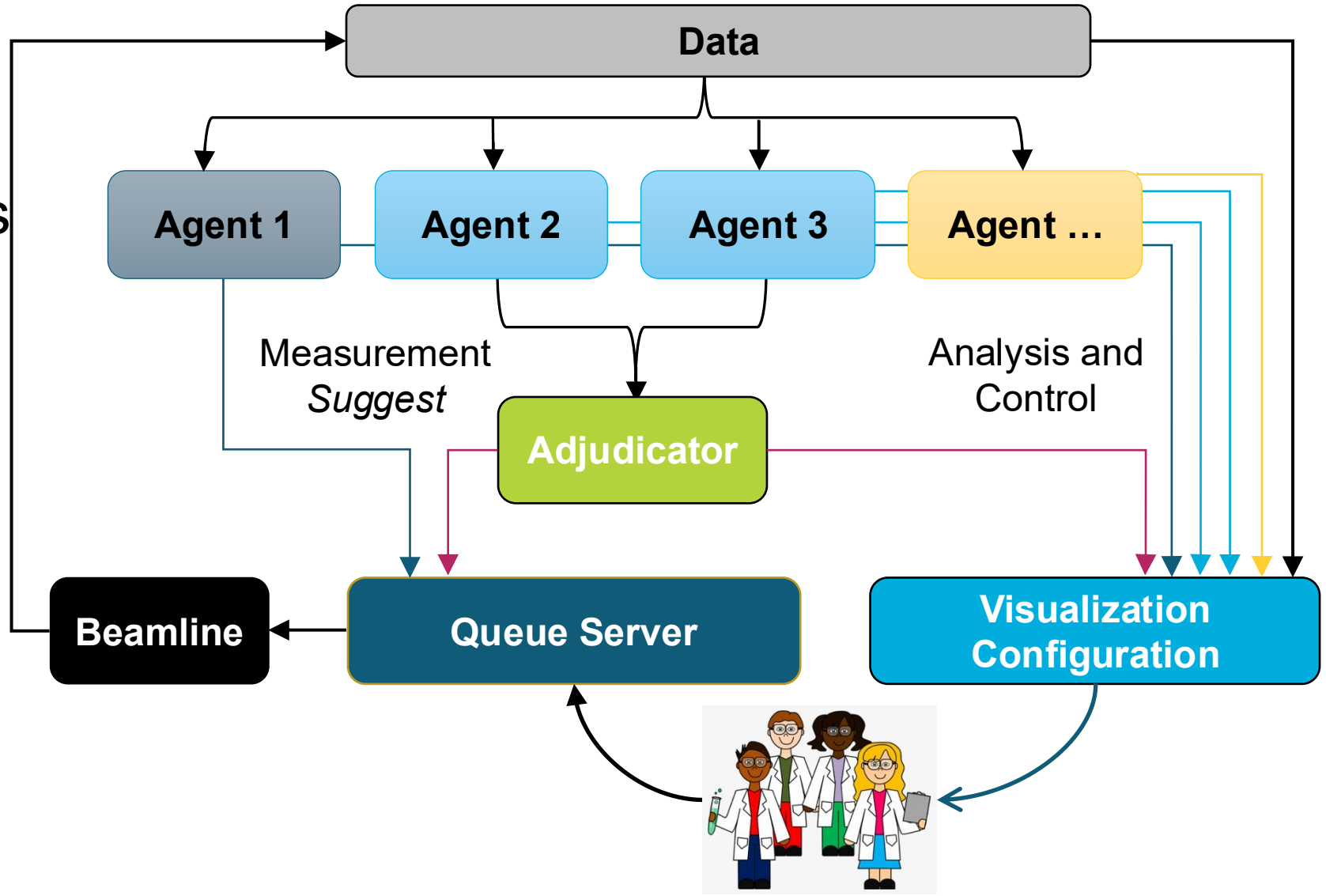
Development of the Bluesky-Adaptive framework: https://github.com/bluesky/bluesky-adaptive

- Each agent exposes a simple interface (*ingest*, *suggest*, *report*).
- Agents can be written in any language (Python, C++, Rust, …) and can be AI-enabled or can perform conventional analysis.
- Multiple agents can work in parallel, asynchronously “suggesting” next actions to take.
- Adjudicator to pick the best suggestion (more in the docs).
- Requires an ecosystem of tools/services for the distributed data acquisition/access/processing:
  - Message bus (Kafka, Redis Stream, ZMQ, …)
  - Bluesky-Queueserver (data acquisition)
  - Tiled (data access)
  - Agents wrapped into REST API services (e.g., via VMs/containers hosted in the cloud/k8s infra)

The NSLS-II team will focus on the development of Bluesky-Adaptive after Blop 1.0 release (planned soon), where Blop will be made as an agent in the Bluesky-Adaptive framework.

Empowering the collaboration of many agents and scientists

- Queue Server acts as control hub
- Users query agents and data
- Adjudicators mediate agent requests
- Users maintain authority over experiment



Brookhaven National Laboratory

Summary

- Bluesky provides experiment orchestration, rich metadata, and scalable data access.
- Blop integrates optimization algorithms directly into experiment control.
- Bluesky-Adaptive is aimed to orchestrate multiple agents and provide a pluggable ecosystem to “**Bring Your Own Agent**”.

Thank you!