ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH1BCO05

DIAMOND LIGHT SOURCE ATHENA PLATFORM

J. Shannon, C. Forrester, K. Ralphs, Diamond Light Source, Oxfordshire, UK

Abstract

The Athena Platform aims to replace, upgrade and modernise the capabilities of Diamond Light Source's acquisition and controls tools, providing an environment for better integration with information management and analysis functionality. It is a service-based experiment orchestration system built on top of NSLS-II's Python based Bluesky/Ophyd data collection framework, providing a managed and extensible software deployment local to the beamline. By using industry standard infrastructure provision, security and interface technologies we hope to provide a sufficiently flexible and adaptable platform, to meet the wide spectrum of science use cases and beamline operation models in a reliable and maintainable way. In addition to a system design overview, we describe here some initial test deployments of core capabilities to a number of Diamond beamlines, as well as some of the technologies developed to support the overall delivery of the platform.

INTRODUCTION AND MOTIVATION

The current data acquisition platform at Diamond Light Source (Diamond) suffers from several issues including maintenance difficulty, inflexibility and use of obsolete technology. Further motivations for introducing a new platform have been presented previously [1]. The current platform is known as "Generic Data Acquisition" (GDA) which is a large monolithic Java client-server application. It originally began development over twenty years ago.

The high level goals of Athena are to produce a platform that is maintainable, adaptable, testable, and provides beamline scientists with enhanced capabilities. It strives to leverage industry standard technologies and well tested frameworks.

In 2022 Diamond invited a number of external collaborators to review the design. The architecture was presented and the planned transition steps, as well as the research done to justify the decisions. This received a very positive response, with most feedback relating to the management of the programme rather than the technical design.

The COVID-19 pandemic highlighted the increasing need for a system which supports proper remote access for facility users. The current platform supports remote access in a limited capacity. This could be improved using a modern web based front end, significantly improving the user experience.

Cyber security is also at the heart of a new platform. The remote operation use case in particular reinforces the need for properly managed access to the platform and a robust authorisation and authentication scheme.

Diamond II

Diamond is a currently a 3rd generation synchrontron light source with a photon energy of 3.0 GeV. A major upgrade, named Diamond II, has just been approved to enhance the accelerator to an energy of 3.5 GeV [2]. This will allow for a much higher flux to be delivered to beamline end stations enabling a drastic increase in data acquisition rates. The "dark period" for accelerator upgrade is currently scheduled for late 2027.

Increased flux will have consequences for the computing requirements due to the increased data-rates. This is further compounded by new fast detectors which are being introduced to instruments.

Whilst all instruments will benefit and modernise from the upgrade, a key deliverable is the construction of three new "flagship beamlines". These will be the initial target for the full Athena Platform.

BLUESKY

Bluesky [3] is set of Python libraries for experiment control and collection of data. It was developed at National Synchrotron Light Source II (NSLS-II), Brookhaven National Laboratory [4] and is currently used in several facilities around the world.

Its main features include data streaming, rich metadata, decoupling between experiment logic and hardware, and customisability.

Devices are represented as Python objects which may implement various protocols, e.g. "Readable", providing an abstraction from the underlying communication with hardware. This abstraction and interaction with a control system is handled by the Ophyd library [5].

In Bluesky the term "plan" is used for an experimental procedure. This is somewhat analogous to "scan" in other experimental orchestration frameworks. Plans are customisable and composable allowing for complex procedures to be completed. The composable nature encourages the reuse of existing plans. One feature of particular interest is the adaptive plan which allows the running plan itself to change behaviour to respond to data or conditions.

The choice of Python as the implementation language is also significant as Python plays a large role in the field of scientific research. Beamline scientists will have the opportunity to contribute directly to the experimental plan logic which is not possible with Diamond's current acquisition platform. Libraries of common functions and plans will be curated with the aim to prevent duplication and allow review and optimisation.

Fly Scanning Enhancements in Ophyd

One of the requirements of an acquisition and scanning engine is the ability to perform continuous scans (fly scans). In these type of scans, data is collected whilst motors are moving. At Diamond this is implemented by sending a trajectory to a motion controller (PMAC) and using an FPGA

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH1BCO05

based triggering device (PandA) to send triggers to detectors and record real-time motor positions.

The current software platform achieves this very decoupled software components. The coordinating acquisition engine (GDA) and a middle layer application, Malcolm [6], which controls the devices. Whilst this approach provides some useful abstraction, this barrier between control and acquisition software places limitations on how the overall scanning is sequenced and customised.

At the moment, support for fly scanning in Ophyd is not as comprehensive as that provided by Malcolm. Encouraged by interest from NSLS-II, Diamond has entered the Bluesky collaboration and is actively working on the "ophyd-async" module that provides improved support for fly scanning.

SERVICE BASED ARCHITECTURE

A service based architecture is a well established software design architecture. The key attribute of this involves encapsulation of units of functionality into services. This is in contrast with a monolithic architecture in which a single application contains all the components. Services have their own runtime and are not part of a larger framework.

Advantages of a service based architecture include loose coupling between components, flexibility to change implementations inside services without affecting interaction with the rest of the system, and the ability to test service functionality in isolation.

The Athena platform will scope services to the "functional block" level of granularity; it is not a true *micro*services architecture. A service in this context is an encapsulated block of functionality with an API.

The aim is to use industry standard software and principles where possible. These libraries and frameworks will be widely used and well proven which will aid in recruitment and on-boarding of new staff.

In addition to the services themselves the platform overall will benefit by the provision of templates for standardised logging and monitoring of services.

Providing sufficient support in the communications layer, the use of a service based architecture allows for services to be written in different languages which might be invaluable if one language is better suited to solve a particular problem.

Communications

In a service based architecture there must be communication both internally between services and externally to the outside world. In the age of the web, representational state transfer (REST) has become widely used for the latter. REST allows for a HTTP-based API to be defined. Due to its wide usage there are libraries available in many languages to support development with REST.

The use of HTTP based communication protocols satisfies the required flexibility in both the services themselves and clients who use them. This choice of interface paves the way for future web based clients which is not possible with the current platform.

Athena has requirements for asynchronous communication, both internal and external. To provide internal inter-service communications, the services will be deployed alongside a message bus. There are many choices for the implementation of this, for example ActiveMQ [7] or RabbitMQ [8]. For external asynchronous subscription support, the WebSocket protocol will be used to propagate updates to clients.

Kubernetes

Kubernetes [9] is a platform for deployment and management of containerised applications. Containerisation involves running an application in isolation from the underlying host machine. This has a number of benefits including portability, management of host resources and low resource cost isolation. Kubernetes was originally designed by Google; it is now prolific amongst the technology industry.

Diamond intends to use Kubernetes as the deployment environment for the Athena to make use of it's well proven service management, monitoring and traffic routing facilities.

There is a rich ecosystem of tooling and support for Kubernetes. This includes monitoring, dashboards and alerts. These features will aid maintenance and ensure that any issues can be diagnosed efficiently.

Kubernetes supports the use of a service mesh which is a layer of infrastructure to add capabilities such as routing and security transparently. This means that the services themselves should not have to implement features such as authentication. An example is the Istio service mesh [10].

A schematic of Athena services deployed on Kubernetes is shown in Fig. 1. This illustrates the interaction between services internally in addition to the exposed public API.

At Diamond the intention is that a Kubernetes cluster is deployed per beamline. This will provide a level of isolation, ensuring that any adverse situations (e.g. resource hogging) caused on a particular beamline do not impact the operations of the others. Additionally there is a central cluster where specific services may be deployed if they are deemed central services. This cluster has a dedicated support team and is already provided as an operational platform.

Finally, it is noted that these clusters are all on-premises rather than cloud clusters. This is due to the vast amount of data captured and processed.

Unified Deployment Platform

Diamond has a large group of software developers. The Scientific Software Controls and Computing department (SSCC) consists of approximately 150 people. This is further split into smaller groups focusing on different levels of the software stack; for example Accelerator Controls and Data Analysis.

In the past these groups have been somewhat separated from each other, both in terms of direct leadership and technological aspects such as software deployment approaches.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH1BCO05

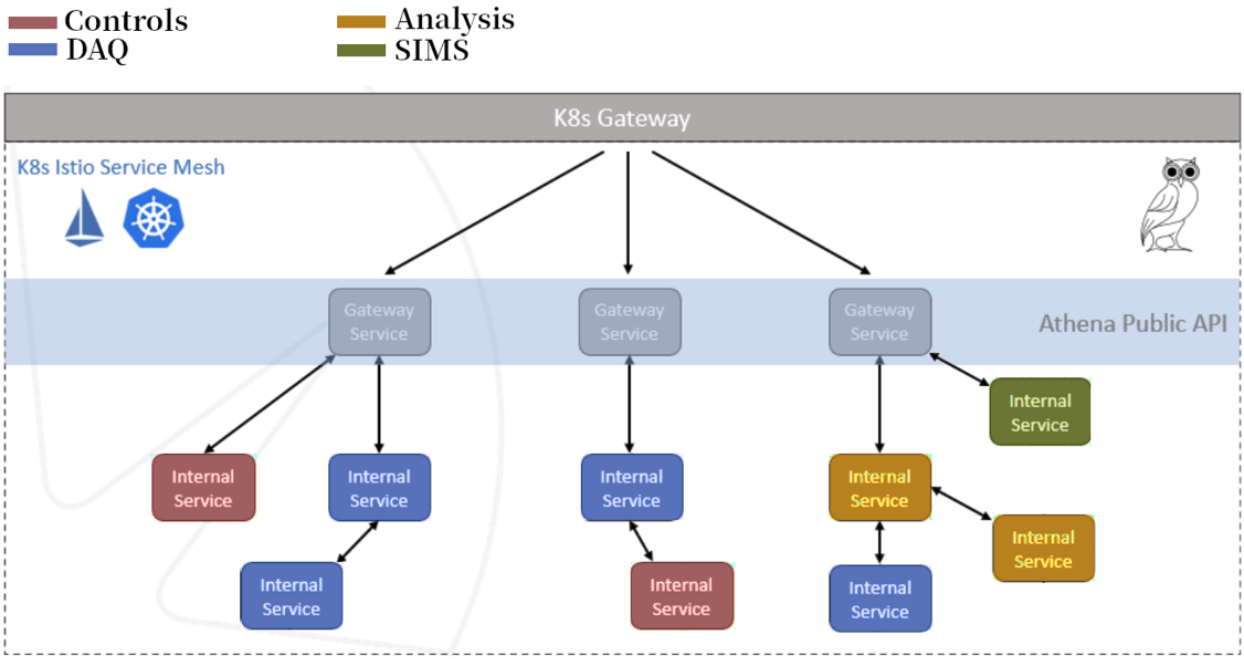

Figure 1: Schematic of Athena services deployed on Kubernetes. The public API is layered behind a service mesh. The coloured key shows different software groups deployed in SSCC.

The proposal for moving towards a service based architecture deployed on Kubernetes will unify many aspects of software deployment and promote increased collaboration between the software groups at Diamond by providing a shared environment to which services from any group can be deployed. Standard approaches will be developed for continuous integration and deployment.

In order to support this cross-group development model, SSCC has recently deployed a developer portal powered by Backstage [11]. This consists of a software catalog with the aim of documenting all software components at Diamond, their owner(s), and their API documentation. Additionally the software groups have begun to write policies for software design, security, and deployment to provide guidance and aid in the on-boarding process for both new staff and those migrating to developing for the Athena platform. These are available to view in Backstage.

EPICS IOCs on Kubernetes

Diamond uses EPICS [12], almost entirely, as its control system. EPICS is widely used within the accelerator and experimental physics sphere and is mature and robust. The input-output controller (IOC) is the fundamental unit for a program controlling a specific component of hardware.

Diamond currently manages IOCs using a set of custom application development environment scripts and mechanisms. This includes deployment, configuration and release of individual IOCs and associated EPICS sub-modules.

Going forward the intention is to develop containerised IOCs [13] which will be deployed in Kubernetes in the same way as the Athena Platform services are deployed.

The primary form of these IOCs with be Python soft IOCs written using the "softioc" module [14] unless there are specific performance requirements. This Python based approach lowers the bar to production and maintenance of IOCs by simplifying the process and tailoring it to a containerised model.

Core Acquisition Services

The service based architecture allows for initial implementations of core functionality to be deployed alongside existing GDA installations, allowing for testing and development of Athena services without impacting current operations. This will provide new acquisition capabilities to beamlines with a minimal set of services to which others can be added, delivering further functionality over time. Currently these core services consist of: BlueAPI and the Nexus file writing service.

BlueAPI This service encapsulates the Bluesky Run Engine and provides a REST API to allow other services to run plans [15]. The Bluesky Documents resulting from the run are emitted onto the shared message bus for consumption by other services. Bluesky was originally designed to be used in an interactive Python environment; putting it behind an API allows other clients to interact with it in a controlled manner.

NeXus File Writing Service Consumes documents from Bluesky and writes NeXus files. The NeXus file format [16] is the preferred file format at Diamond. This service will also support writing data to conform to NeXus application

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH1BCO05

definitions. Using this in conjunction with BlueAPI allows basic acquisitions to be performed resulting in data written in the standard format already used by Diamond.

Future Services Other services will be developed and deployed as there is capacity or a business requirement. Some examples of these are:

- Session Management
- Metadata Catalogue
- Data Streaming
- Experiment Logbook
- Job/Task Manager

Transitional Architecture

In order to migrate from the current software stack there will be a period where a mixture of legacy components and new Athena services are in use concurrently.

The service based architecture lends itself to an iterative approach, building out from the baseline components.

Furthermore, support for calling out to the BlueAPI has been added to the existing acquisition platform allowing the two systems to be deployed side-by-side and operated through the same interface.

ONGOING WORK

The new flagship beamlines built by the Diamond II upgrade are the target for the full Athena platform; operation without use of the legacy existing platform. This remains a long way in the future, however, so Diamond would like to make use of some of this functionality before this time to develop and prove the approach.

In terms of core development, the legacy platform is now effectively in maintenance mode. It is expected to be used for several years but generally there it will not be enhanced to deliver new functionality. New beamline projects, driven by new science requirements, will be implemented by Athena components. Bluesky and Ophyd are the key components here.

These components may be initially deployed, as described above, on current beamlines alongside the existing platform. The current user facing interface will be able to make calls to the new components providing a consistent experience for operation during this time. This will allow developers, scientists and users to start to become familiar with these core frameworks and to inform the design for the eventual form of Athena. Initial projects making use of this approach include: I22 Time Resolved Scanning and MX Hyperion.

Another aspect of the platform which needs to be modified is the user interface. In order to capture the current user experience and requirements, a team of consultants has been brought in to review Diamond's user interfaces across various applications.

I22 Time Resolved Scanning

I22 is a Diamond beamline specialising in multipurpose small and wide angle X-ray scattering (SAXS and WAXS). A project is currently underway to provide flexible hardware triggered scanning to the beamline to enable increased data collection rates and new scanning techniques such as SAXS tensor tomography.

These advances are realised with the flexibility of scan plans in Bluesky and the rich set of configuration parameters afforded by Ophyd. The integration of this with the top-most layers of the existing data acquisition stack allows for a gentle conversion to future technologies, helping transition staff and users to next generation technologies.

MX Hyperion

Macromolecular Crystallography (MX) beamlines form a large segment of Diamond's instruments. Their procedures aim to be highly automated and maximise throughput. Some beamlines employ completely automated data collections. This is currently implemented within GDA, which was not originally designed for maximum efficiency automated control. The procedures cannot be optimised fully under GDA's constraints.

Bluesky, on the other hand, allows the experimental procedure to be optimised by parallelising and reordering the steps. This allows for an increased rate of data collection. Work is already underway with the Hyperion project [17]. Bluesky has been deployed on a number of MX beamlines where it has shown to improve efficiency over GDA and has been well received by beamline staff. Ongoing work will bring the project to use Athena's BlueAPI service.

Simulations

The acquisition platform has historically been difficult to test in an offline environment. Whilst it is possible to write "dummy" implementations of devices there has not always been sufficient resource available for maintaining these. Additionally there is a vast range of different hardware in use across the facility.

Diamond has recently developed the Tickit simulation framework [18]. This framework allows simulation at the device layer and therefore the same IOCs can be run against the simulated devices. This is a big advantage as it allows the higher levels of acquisition software to test the exact same controls interface as the real hardware.

ViSR

Visible Synchrotron Radiation (ViSR) is an attachment to Diamond's storage ring where a spectrum of visible light is taken from one of the ring's dipole magnets. This was originally installed as an engagement project allowing visitors to interact with synchrotron light.

Since ViSR is located within the experimental hall and adjacent to existing beamline computing infrastructure it provides an opportunity for testing Athena in an environment which is close to that of a real beamline.

Beamline in a Box

In order to boost familiarity with Bluesky and Ophyd the concept of a "beamline in a box" has been devised. This

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH1BCO05

consists of a small set of equipment including a Raspberry Pi and a motor which can perform a simple visible light tomography scan.

The main objective of this equipment is to provide a cheap, portable set of devices which allow Bluesky and Ophyd to be installed. This gives a real system that both developers and scientists can use to learn about these frameworks.

Engagement

Efforts to form a community of Bluesky users have begun at Diamond. SSCC have set up a dedicated training room within the containing a set of simplified hardware stacks which are representative of a typical tomography beamline. Figure 2 shows what one of these training rigs looks like.

This resource allows prototyping and experimentation with the Bluesky framework and provides an open forum to discuss ideas and develop good practices. In particular beamline scientists are encouraged to participate.

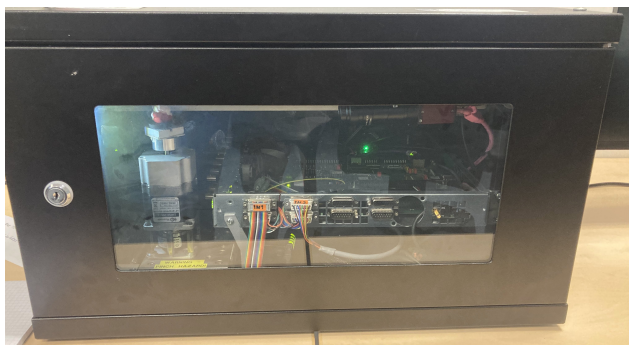

Figure 2: Training rig capable of running hardware triggered tomography scans.

CONCLUSION

The need for a new acquisition platform at Diamond is clear and Athena will provide a flexible, maintainable and testable software framework for the future which can adapt to meet the needs of the beamlines.

Deploying components as discrete services on Kubernetes provides the functionality desired in terms of deployment management, monitoring, and cyber security.

The use of Bluesky as the core of the scanning framework provides a flexible and extensible tool set for experimental logic and the hope of further fruitful collaborations with other facilities.

REFERENCES

[1] K. Ralphs and J. Handford, "Future Acquisition Architecture Investigations at Diamond", in *Proc. ICALEPCS'19*, New York, NY, USA, October 2019, pp. 1240-1243. doi:10.18429/JACoW-ICALEPCS2019-WEPHA060

[2] I. P. S. Martin and R. Bartolini, "Conceptual Design of an Accumulator Ring for the Diamond II Upgrade", in *Proc. IPAC'18*, Vancouver, Canada, Apr.-May 2018, pp. 4046–4049. doi:10.18429/JACoW-IPAC2018-THPMF008

[3] Bluesky homepage, https://nsls-ii.github.io/bluesky

[4] Daniel Allan, Thomas Caswell, Stuart Campbell, and Maksim Rakitin, "Bluesky's Ahead: A Multi-Facility Collaboration for an a la Carte Software Project for Data Acquisition and Management", in *Synchrotron Radiat. News*, 32:3, pp. 19–22. doi:10.1080/08940886.2019.1608121

[5] Ophyd homepage, https://nsls-ii.github.io/ophyd

[6] T.M. Cobb *et al.*, "Malcolm: A Middlelayer Framework for Generic Continuous Scanning", in *Proc. ICALEPCS'17*, Barcelona, Spain, October 2017, pp. 780-784. doi:10.18429/JACoW-ICALEPCS2017-TUPHA159

[7] Apache ActiveMQ, https://activemq.apache.org

[8] RabbitMQ, https://www.rabbitmq.com

[9] Kubernetes, https://kubernetes.io

[10] Istio service mesh, https://istio.io/latest/about/service-mesh

[11] Backstage, https://backstage.io

[12] EPICS, https://epics.anl.gov/

[13] EPICS Containers, https://github.com/epics-containers

[14] Python softioc module, https://dls-controls.github.io/pythonSoftIOC

[15] BlueAPI, https://diamondlightsource.github.io/blueapi

[16] NeXus Data Format, https://www.nexusformat.org

[17] D. Perl, "Ultra-High Throughput Automated Macromolecular Crystallography Data Collection Using the Bluesky Framework", presented at The 19th Biennial International Conference on Accelerator and Large Experimental Physics Control Systems Conf. (ICALEPCS'23), Cape Town, South Africa, October 2023, paper THMBCMO34, this conference.

[18] A. Emery, T. Cobb, C. Forrester, and G. O'Donnell, "Tickit: An Event-Based Multi-Device Simulation Framework", presented at The 19th Biennial International Conference on Accelerator and Large Experimental Physics Control Systems Conf. (ICALEPCS'23), Cape Town, South Africa, October 2023, paper TUPDP109, this conference.