ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-WE3BC008

EFFICIENT AND AUTOMATED METADATA RECORDING AND VIEWING FOR SCIENTIFIC EXPERIMENTS AT MAX IV

D. van Dijken*, M. Leorato, M. Eguiraun, V. Hardion, M. Klingberg,
M. Lindberg, V. Da Silva, MAX IV Laboratory, Lund, Sweden

Abstract

With the advancements in beamline instrumentation, synchrotron research facilities have seen a significant improvement. The detectors used today can generate thousands of frames within seconds. Consequently, an organized and adaptable framework is essential to facilitate the efficient access and assessment of the enormous volumes of data produced. Our communication presents a metadata management solution recently implemented at MAX IV, which automatically retrieves and records metadata from Tango devices relevant to the current experiment. The solution includes user-selected scientific metadata and predefined defaults related to the beamline setup, which are integrated into the Sardana control system and automatically recorded during each scan via the SciFish library. The metadata recorded is stored in the SciCat database, which can be accessed through a web-based interface called Scanlog. The interface, built on ReactJS, allows users to easily sort, filter, and extract important information from the recorded metadata. The tool also provides real-time access to metadata, enabling users to monitor experiments and export data for post-processing. These new software tools ensure that recorded data is findable, accessible, interoperable and reusable (FAIR) for many years to come. Collaborations are on-going to develop these tools at other particle accelerator research facilities.

INTRODUCTION

Metadata, often referred as data about data, has become more and more important in the scientific field to complete and accompany the results of an experiment. Metadata is often very important to contextualise scientific result and help with the reproducibility of an experiment.

The increasing importance highlights the need to treat metadata with the respect. To collect the metadata in a complete and efficient way, to store it safely and most importantly, in a transparent way for the scientific user, to not increase the burden of work for that already follow the execution of an experiment.

It is also essential to guarantee ways to explore and use such metadata otherwise the whole collection process become meaningless.

Solving this issue has been an important priority to MAX IV with the aim of creating a system that would allow the collection of metadata related to an experiment and provide additional functionalities for the coming scientist during their experiments.

Lastly it is also essential to be able to navigate and use the metadata collected in a efficient way to improve the experience and enable as much as possible scientific operations.

SCIZOO

SciZoo is the collection name of SciCat and the associated services running at MAX IV. SciCat is a collaboration project between several research facilities to support a uniform way of storing metadata. SciCat serves as a web-based graphical user interface designed to showcase scans alongside their accompanying metadata. Its primary aim is to provide a straightforward overview of conducted scans, linking them to specific proposals, detailing relevant parameters, and indicating the data storage location.

The central SciCat webpage presents a comprehensive scan overview, with each scan occupying a dedicated page. Users can log in using their MAX IV user credentials to access scans within their purview. For beamline staff, this encompasses all beamline scans, whereas members of proposal groups can access scans linked to their respective proposals. Publicly shared scans are visible to anyone without the need to log in.

Upon selecting a specific scan, users gain access to all of metadata associated with that scan. This includes details such as scan name, description, a unique PID (also available in the URL for easy reference), data type (raw or processed), and creation timestamp. Additionally, information pertaining to the associated proposal, such as the proposal owner and principal investigator's email, is provided. The storage path of the data is also disclosed. Finally, a list of scientific metadata recorded during the scan is presented.

A schematic overview of SciCat and its associated services is depicted in Fig. 1. At MAX IV, all sub-systems combined is often referred as SciZoo. A detailed description of each of the services included in the SciZoo suite and other relevant services at MAX IV follows below.

DUO

DUO, the Digital User Office at MAX IV, plays a pivotal role in managing user access and data. Before arrival at the facility, each user is assigned a proposal within DUO, and only individuals affiliated with this specific proposal possess the authorization to access the associated data stored on the facility's data storage system. The Tango control system [1] employs a dedicated Tango Device known as "PathFixer" to efficiently generate the requisite file path on the disk with the appropriate permissions for each proposal. Furthermore, PathFixer is responsible for disseminating this path information to the scanning orchestration systems such as Sardana,

* daphne.van_dijken@maxiv.lu.se

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-WE3BC008

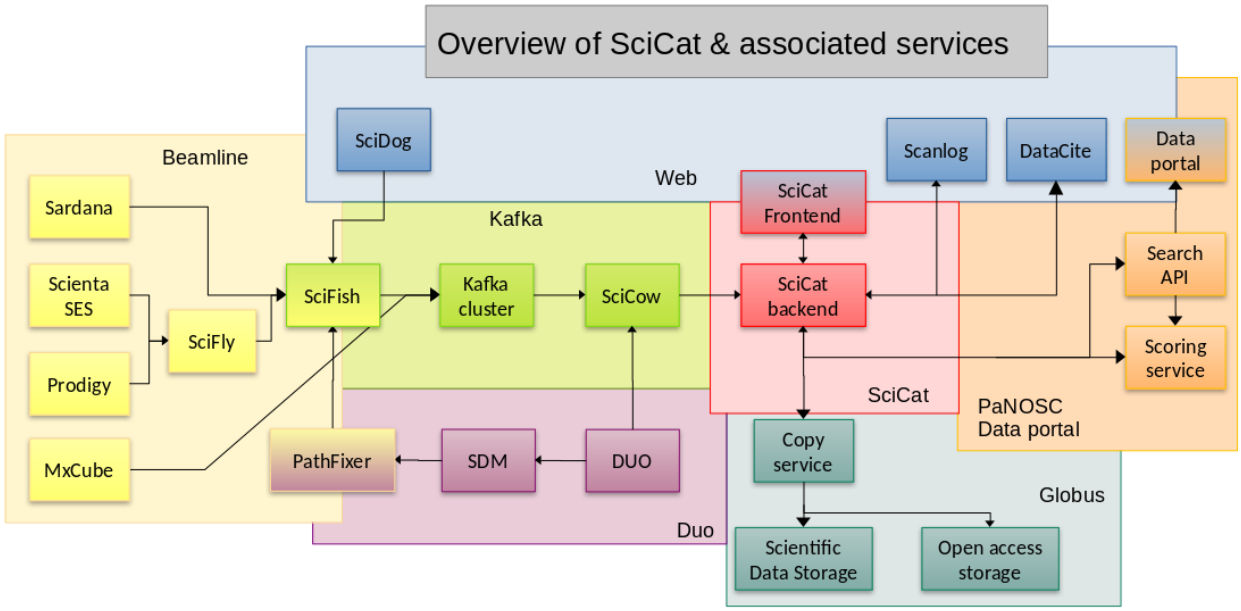

Figure 1: SciCat and its associated services.

ensuring seamless data management. Facilitating communication between the Tango layer and DUO is the Python library "SDM" (Scientific Data Management).

SciDog

Given the diverse requirements and scientific interests associated with each beamline and individual experiment conducted at MAX IV, a unified configuration system, SciDog, has been established. This system is designed to tailor the storage of scientific metadata for specific scans to meet the unique needs of each scenario.

Users have the option to save configurations with their chosen metadata selections. They can utilize all the device/attribute combinations available in the Tango Database and select which are relevant for their current acquisition. To ensure its accuracy and relevance, SciDog is regularly updated by both the beamline staff and the facility visitors. SciDog comprises a frontend built with ReactJS, a backend developed in Typescript, and utilizes a MongoDB database to store these configurations.

The SciDog web interface is divided into two main sections. On the left-hand side, users can view and configure existing metadata, while on the right-hand side, there is an interface for adding new metadata. If needed, metadata can be grouped to enhance its organization within SciCat [2] and Scanlog [3], although grouping is not mandatory. It is advisable to use user-friendly names for metadata to provide a clear understanding of their purpose. The device and attribute values correspond to the Tango database, and when a unit is specified for a device/attribute pair, it will also be displayed; alternatively, users can manually add a unit if necessary.

Mapping is an optional feature that allows users to alter the displayed output values of attributes to enhance readability. For instance, if a device's state is represented by values 4 and 5, but it corresponds to "insert" and "extract", users can define this mapping for better readability in the future.

Additionally, users have the flexibility to define multiple configuration settings. For instance, they can configure a smaller metadata set for a specific scan. This configuration can be designated as the "active configuration", which will be used during an acquisition.

SciFish

SciFish [4] is a Python library that operates as an efficient Apache Kafka [5] producer. SciFish assumes the crucial role of systematically accumulating metadata from diverse sources, encompassing the scanning orchestration system, PathFixer, and the scientific metadata sourced from Tango Devices. This library establishes a dependable and expeditious means of data aggregation while also interfacing with SciDog to retrieve configuration information.

SciFly

SciFly is introduced as a service designed to monitor file activities within the presently active path managed by PathFixer Tango Device. This service operates by utilizing newly written files (or modified files, when the feature is enabled) as triggers, prompting the automatic extraction and writing of associated metadata to SciCat. This functionality is accomplished through the underlying SciFish library.

SciFly is currently enabled to listen to files orchestrated by Scienta SES and Prodigy SPECS acquisitions and is easily configurable for the beamlines needs.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-WE3BC008

SciCow

SciCow is the Kafka Consumer responsible for ingesting datasets into SciCat. It is a python-based service running at MAX IV which is responsible with reading the messages from the Apache Kafka queue. There is one topic per beamline, and a single consumer that reads all the topics. SciCow receives the JSON dict from the producer, SciFish, and ingests the data into SciCat. It has a connection to DUO to validate the proposal and check basic information like principal investigator, title and abstract.

SciToad/Scanlog

SciToad, or Scanlog, is a web-based interface to explore information stored in SciCat, which offers a concise representation of the metadata. In Scanlog, users have the capability to rate their data and add comments, facilitating a quick assessment of the success of a given scan.

Scanlog, written in ReactJS, relies entirely on the SciCat database via the NestJS API; it does not operate with its own independent backend. User authentication is conducted through SciCat, utilizing the Keycloak [6] authentication system.

Within Scanlog, users can create customized tables based on specific metadata parameters they wish to visualize. Moreover, users can select and export scans to Elogy [7] by either generating a new entry in their preferred logbook or appending to an existing entry.

To filter the scans displayed in the table, two options are available. On the top left, users can select a time frame for the scans to be shown, adjustable by toggling to the left and selecting a different time range from the ensuing window. On the top right, users can choose a proposal ID to display only scans associated with that particular ID.

Directly beneath the table, users can customize the displayed columns. By clicking the ”x” next to a header, the corresponding metadata column will be hidden. Conversely, columns can be added by clicking in the same row and selecting an item from the drop-down list. Users can also reorder columns by simply dragging the headers. These personalized settings are saved within the browser on the user's computer.

In the actual table, metadata is presented row by row for each scan. A link to the respective SciCat page for each scan can be accessed by clicking the green ”Link” button. Users can also leave comments specific to each scan, upload images, and assign Data Quality Metrics ratings. The absence of stars indicates an unrated scan, while ratings can range from half a star to a maximum of three stars, reflecting the perceived quality of the scan.

Data can be exported from the table by clicking the selectable button on the left of each row. Export options include sending data to Elogy, which allows users to create a new entry or append to an existing one in their chosen logbook. When creating a new entry, users can specify a custom title or utilize the default title, ”Exported from Scanlog.” Only the columns currently visible in the table will be exported to Elogy. Additionally, data can be exported to a CSV file, which will be automatically downloaded.

Data Portals

The SciCat web interface facilitates the publication of datasets, enabling the assignment of a stable DOI (Digital Object Identifier). Users can select multiple scans and add them to a cart. Subsequently, datasets can be published via the actions page, resulting in the allocation of a static DOI hosted at https://doi.maxiv.lu.se/. This DOI can serve as a reference in academic publications. By following this DOI link, readers gain access to public dataset pages, obviating the need for authentication. Data is available for download through the Globus system and can be discovered through search engines such as https://data.panosc.eu/ and more forthcoming platforms.

Given the stringent data access permissions and the incentive to preserve the integrity of raw data at MAX IV, a dedicated service operates to facilitate data file access. When a user seeks access to specific data, this service initiates the secure copying of data files to a publicly accessible storage location. Public datasets are made readily available for access by all users, who can conveniently request download links for the Globus platform through the SciCat web interface.

DEPLOYMENT

Following the requirement for proper metadata management, special attention has been given to the deployment of the critical components of the metadata flow to ensure that no metadata will be lost.

The deployment of the required software can be split in two categories: ”cloud” and local. The former is used for all the system that don't work directly with the metadata acquisition. The service of choice was to rely on a modern Kubernetes [8] version from Red Hat, OKD [9], to ensure proper redundancy and availability of the essential system. Even though it can be considered a cloud solution the physical infrastructure for it is still internal to MAX IV to provide for more control over the system. Kubernetes was also chosen as it synergizes very well with the use of an Apache Kafka queue that ensure that messages are properly managed and a distributed MongoDB [10] deployment. The latter instead is for those system that are closer to the metadata acquisition in which case they need to be deployed, locally, on each beamline as they need to talk directly to the acquisition software used at MAX IV.

Pipelines

The new system was set up with an easy and quick way of deploying to improve the stability and the security of the system itself. This has been done taking advantage of the GitLab [11] CI feature to improve the quickness of deployment and reduce the possibility of a human mistake to a minimum. The pipelines for all the projects needed for the metadata collection have a similar structure but there is a

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-WE3BC008

small difference on the last part depending if the service in question is a web application that will be deployed on Kubernetes or is a python package which will need to be installed on the beamline machines to interact with the acquisition system. In both cases the code is kept in a GitLab repository where MAX IV developer can make update and improvements to it. When a new feature is ready for testing it is merged into the main branch of the repository which trigger an automatic deployment. This process start by building a new docker image that will be used for the deploy, then has the ability to run some automated test on it (when present), then run the deploy of the new images to the test Kubernetes environment. After the added feature has been validated and approved a git tag will be created to finalise the version. This is where a slight difference in the production deployment can happen. The deployment to Kubernetes allow for almost no downtime during updates which enable us to deploy new updates as soon as they are ready and tested. Deployment to beamlines instead can be more impacftul so those are held and done together with our weekly updates for all the beamlines.

A slight difference in this process is present only for the deployment of SciCat. Since SciCat is a collaboration between multiple facilities the codes lives in GitHub meaning that updates and new features are added there. For this reason the MAX IV repository only contains some extra customisation files and the Helm [12] chart needed for the deploy. During the build of the docker images in this case the pipeline will pull the SciCat files from the external repository, add our customisation on top of it and then build the final images to be used. The rest of the deployment instead proceed as the others.

Kubernetes OKD

For the Kubernetes infrastructure at MAX IV the community project OKD is used. OKD is the upstream project for the Red Hat OpenShift project. This is used in conjunction of Helm to allow for easier and quick deployment of our software while having resilient system that enable the needed stability to safely collect metadata during experiments. OKD also allow for an easier developer approach with an easy to use UI while keeping all the more complex functionalities of the Kubernetes environment through the use of the Kubernetes CLI.

Apache Kafka

Apache Kafka is an open-source event streaming platform that is currently maintained by the Apache Software Foundation. The reason it was chosen for the metadata workflow in MAX IV is due to it's high reliability, very high performances in event streaming, compatibility and it being considered the state of the art in event streaming queues. All this without needing compromise on the open-source focus in the facility.

Apache Kafka was also deployed on our Kubernetes OKD infrastructure to allow for a proper setup to make full use of the high availability and reliability features of Apache Kafka. This ensure that even in case of slight malfunctions in the ingestion software the system is still able to properly collect the metadata that will then be stored in SciCat as soon as the system is back online. The Apache Kafka configuration at MAX IV was also made by creating different queue for each beamline allowing for a clear separation in the source of the metadata.

Database, MongoDB

In SciCat the database of used is MongoDB, this was a choice based on the nature of metadata since that, often, tend to represent more a document with different field and can be difficult to fit into more strict SQL tables. This make this kind of metadata work quite well with the document oriented feature of MongoDB.

To ensure major data safety the database has been deployed on the MAX IV Kubernetes infrastructure by taking advantage of MongoDB already being built toward a high availability and data redundancy. Having two server room available a structure with two replicas, one per room, and an arbiter to guarantee the system availability and the safety of the metadata collected as seen in Fig 2.

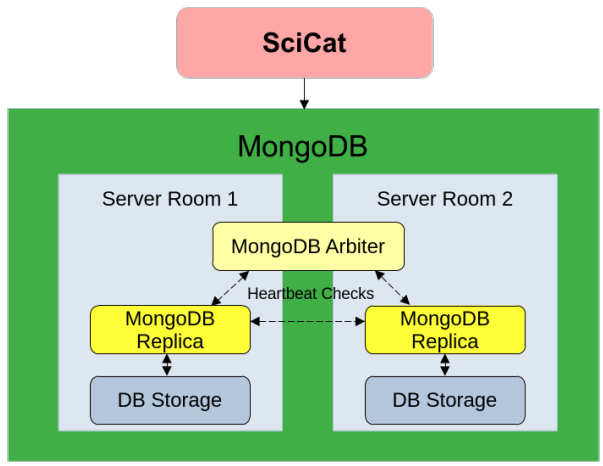

Figure 2: MongoDB replicas structure. The Arbiter can be moved in between the servers but lives on a single one at the time.

To guarantee the proper functionality some extra configuration was needed for the MongoDB *writeConcern* value as by default it is set to *w:majority* but with a 2+1 this translate into always requiring both replica to be reachable. This allow for having data redundancy but does not allow for high availability. For this reason the *writeConcern* was set at *w:1* to allow also for the high availability functionalities.

USAGE

At present, MAX IV operates 16 beamlines, with SciCat integration established in 14 of them. The predominant orchestration system for scan and acquisition management is Sardana. Consequently, a custom Sardana recorder has been developed, utilizing the SciFish library as its foundation. During the acquisition process, metadata is automatically

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-WE3BC008

extracted from the control system and ingested into SciCat once the acquisition is complete.

In total, this results in 244.781 metadata of scans uploaded to SciCat since 2019 (see Table 1).

Table 1: Current Status, Uploads Per Beamline

Beamline	Control System	Datasets
CoSAXS	Sardana	77.965
HIPPIE	Prodigy, Scienta	53.352
ForMAX	Sardana	46.255
BioMAX	Sardana, MxCube	30.130
FlexPES	Sardana, Scienta	21.413
MAXPEEM	Sardana, Uview	12.289
Balder	Sardana	7.178
NanoMAX	Contrast	4.066
SPECIES	Sardana, Prodigy	1.907
Veritas	Sardana	1.301
FinEstBeAMS	Sardana	971
DanMAX	Sardana	328
FemtoMAX	Sardana	14
MicroMAX	Sardana, MxCube	1
BLOCH	Scienta	0
SoftiMAX	STXM, Contrast	0
		244.781

Following the upload of a scan, SciCat serves various purposes. The most common utilization is through Scanlog web interface during acquisition procedures. Scientists are provided with a real-time overview of their acquisitions and can construct customized tables containing metadata of immediate interest. Additionally, besides data visualization, SciCat facilitates the addition of comments and data export to formats such as CSV or MAX IV's internal electronic logbook system, Elogy.

Another increasingly prevalent use-case at MAX IV is the publication of datasets in scientific papers, with SciCat serving as the platform to publish corresponding metadata and raw data files. Figure 3 offers an illustrative example of this process.

FUTURE WORK

In the realm of scientific research at MAX IV, SciCat has gained increasing traction among scientists and users alike. The volume of datasets being incorporated into the system has shown gradual growth over time, yet there remains untapped potential for heightened activity. Some beamlines, for instance, are not currently set up for automatic data ingestion, presenting an opportunity for enhancement.

Continuous efforts are underway to streamline and enhance the user experience, making it more seamless to work with the SciCat system. One significant avenue for improvement involves expanding the wealth of information contained within SciCat. Presently, MAX IV has not fully harnessed the full spectrum of capabilities that SciCat offers. By incorporating comprehensive details about samples and instruments, the categorization and organization of the data sources can be enhanced.

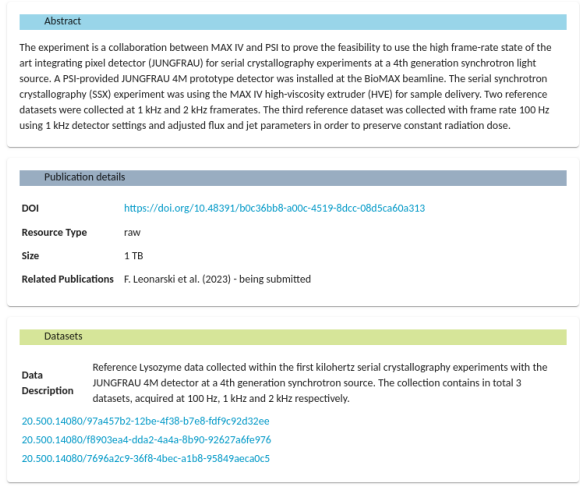

Figure 3: Example of published dataset.

Furthermore, an exciting prospect on the horizon is the implementation of data processing and analysis pipelines, potentially leveraging tools such as JupyterHub. This development has the potential to revolutionize the workflow for the users. With all data accessible in a centralized location, the process of data analysis becomes significantly more efficient and user-friendly.

In summary, SciCat at MAX IV is on a trajectory of continuous improvement, and its adoption among our scientific community is steadily on the rise. We anticipate that ongoing efforts to enhance data ingestion, expand data information, and implement advanced data processing pipelines will further elevate the utility and impact of SciCat for our researchers.

ACKNOWLEDGEMENTS

MAX IV appreciatively acknowledge Knut and Alice Wallenberg Foundation (KAW) for the financial support of the DataSTaMP project and all SciCat collaborators from the different facilities.

REFERENCES

[1] Tango controls, https://www.tango-controls.org/
[2] SciCat, https://scicatproject.github.io/
[3] Scanlog, https://gitlab.com/MaxIV/svc-maxiv-scanlog
[4] SciFish, https://gitlab.com/MaxIV/lib-maxiv-scifish
[5] Apache Kafka, https://kafka.apache.org/
[6] Keycloak, https://www.keycloak.org/
[7] Elogy https://gitlab.com/MaxIV/Elogy/
[8] Kubernetes, https://kubernetes.io/
[9] OKD, https://www.okd.io/
[10] MongoDB, https://www.mongodb.com/
[11] GitLab, https://www.gitlab.com/
[12] Helm, https://helm.sh/