Master's thesis
Master's Programme in Data Science

Data Platform for Accelerating Machine Learning Workflows on Fusion Data

Fran Jurinec

June 4, 2023

Supervisor(s): Prof. Jukka K. Nurminen

Examiner(s): Prof. Jukka K. Nurminen
Dr. Aaro Järvinen

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Degree programme
Faculty of Science	Master's Programme in Data Science

Tekijä — Författare — Author

Fran Jurinec

Työn nimi — Arbetets titel — Title

Data Platform for Accelerating Machine Learning Workflows on Fusion Data

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidantal — Number of pages
Master's thesis	June 4, 2023	54

Tiivistelmä — Referat — Abstract

This thesis explores the applicability of open-source tools on addressing the challenges of data-driven fusion research. The issue is explored through a survey of the fusion data ecosystem and exploration of possible data architectures, which were used to derive the goals and requirements of a proof-of-concept data platform. This platform, developed using open-source software, namely InvenioRDM and Apache Airflow, enabled transforming existing machine learning (ML) workloads into reusable data-generating workflows, and the cataloging of resulting clean ML datasets.

Through a survey of the fusion data ecosystem, a set of challenges and goals was established for the development of a fusion data platform. It was identified that many of the challenges for data-driven research stem from a heterogeneous and geographically scattered source data layer combined with a monolithic approach to ML research. These challenges could be alleviated through improved ML infrastructure, for which two approaches were identified: a query-based approach, which offers more data retrieval flexibility but requires improvements in querying functionality and source data access speeds, and a persisted dataset approach, which uses a centralized workflow to collect and clean data, but requires additional storage resources. Additionally, by cataloging metadata in a central location it would be possible to combine data discovery across heterogeneous sources, combining the benefits of various infrastructure developments. Building on these identified goals and the metadata-driven platform architecture, a proof-of-concept data platform was implemented and examined through a case study. This implementation used InvenioRDM as a metadata catalog to index and provide a dashboard for discovering ML-ready datasets, and Apache Airflow as a workflow orchestration platform to manage the data collection workflows. The case study, grounded in real-world fusion ML research, showcased the platform's ability to convert existing ML workloads into reusable data-generating workflows and to publish clean ML datasets without introducing significant complexity into the research workflows.

ACM Computing Classification System (CCS):
Information systems → Data management systems → Information Integration
Software and its engineering → Software creation and management → Designing software → Requirement analysis
Applied computing → Physical sciences and engineering → Physics

Avainsanat — Nyckelord — Keywords

research data management, platform engineering

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — Övriga uppgifter — Additional information

Acknowledgements

This work was done as a part of the project portfolio of the EUROfusion Advanced Computing Hub (ACH-05) at the University of Helsinki.

Contents

1. Introduction

Nuclear fusion holds the promise of a virtually unlimited, greenhouse gas free energy source for mankind. Fusion energy research is entering the era of reactor-scale devices, as ITER[1], is scheduled for its first plasma in 2025. The reactor-scale fusion facilities will encompass unprecedented challenges for faster than real-time predictions to enable active control and avoidance of costly damage due to off-normal events. Machine learning (ML) and artificial intelligence (AI) algorithms are foreseen as enabling tools to develop fast and accurate models by leveraging the information contained in experimental and simulation databases as well as in theoretical fusion reactor physics. However, the current data management practices in fusion energy research are not optimized for large-scale application of ML/AI algorithms, and the data access patterns represent a significant bottleneck in the ML/AI workflows in fusion energy research.

Thus, the goal of this thesis is to answer the question of:

> "How can modern open-source tooling be used to address the challenges of data-driven research in fusion?"

This question is divided into several sub-questions:

- RQ1. What challenges exist in data-driven fusion research today?
- RQ2. How should a data platform be organized in the context of fusion ML research?
- RQ3. What are the appropriate tools for constructing a fusion data platform?
- RQ4. What can be learned from a hands-on proof-of-concept platform built on open-source solutions?

The scope of this thesis is limited to approaches based on open-source tooling, as academic fusion research is typically not reliant on commercial software offerings.

RQ1 is discussed in Section 2 through a literature survey of recent publications on data-driven plasma science and fusion research in combination with direct insights from subject matter experts. The existing landscape of fusion data is explored through the context of data-driven research techniques such as machine learning (ML) and large scale

statistical analyses. Several key challenges are outlined based on this survey, revolving around two central issues: a highly heterogeneous and geographically scattered source data layer, and a commonly ad-hoc monolithic approach to ML research processes.

RQ2 is explored through Chapter 3, where a four component framework is defined to describe the data flow from the source to final ML utilization: source data, data wrangling, clean data, and ML training (or other analysis). Based on this framework, several architectures are described based on the designated hardware for each of these steps. Namely, the current approach is described using these components, and two main approaches of data-on-demand queries and dedicated ML dataset storage are introduced. Finally, the benefits of adopting a central metadata repository are discussed as a potential solution to enabling a heterogeneous ML data architecture.

RQ3 is evaluated in Chapter 4, through a case-study development of a proof-of-concept data platform. This platform is based on the metadata-enabled architecture outlined in Chapter 3. In the case-study, InvenioRDM was used as the metadata platform for dataset publishing and discovery. Additionally, the use of workflow orchestration tools is identified as a way to improve the reusability and longevity of data generating codes. In the proof-of-concept platform Apache Airflow was used as the workflow orchestrator. A selection of alternative open-source tools for both of these components is also presented, and evaluated based on criteria relevant for this proof-of-concept deployment.

Finally, RQ4 is evaluated through the final sections of Chapter 4 which describe the challenges and practical conclusions from the proof-of-concept platform. It is identified that sufficient technologies exist of implementing a functional data platform for use in fusion research, with further evaluation necessary on the topic of HPC-compatible tooling and the feasibility of dedicated storage for ML data. The wider implications, limitations, and applicable insights are presented in Chapter 5, along with cumulative learnings from this thesis and outlined opportunities for future research dedicated towards the development of a live fusion data platform.

2. Survey of Data and ML in Fusion

To better frame the challenges of ML research, it is important to form a clear picture of the current state of data in fusion. This chapters presents an overview of existing data formats, infrastructure, and derived challenges found in modern fusion ML research. Section 2.1 explores the source data characteristics and the limitations of the current data access layer. Section 2.2 further explores how this infrastructure led to monolithic workflows and the challenges which arise from it when performing ML research. In Section 2.3, the challenges of data and ML research workflows are translated into a set of goals which form the basis for the development of a fusion data platform. Finally, Section 2.4 introduces related research efforts from both within the fusion community, and from other domains.

2.1 Challenges of Data in Fusion

In order to better understand the requirements for data infrastructure, it is essential to first understand more about the data itself, what does it represent, where it originates, how is it stored, how is it accessed, and finally how it is consumed.

Fusion data often comes mainly from two sources: measurements of physical experiments or numerical simulations. Experimental data is collected with diagnostics and represent noisy and incomplete information from the full physics fidelity ground-truth information. Due to the high cost of building/running physical experiments, numerical simulations of varying fidelity are also used as an alternate data source, providing a complete state information bound by our current physics understanding. Simulations can also be used to analyze physical relationships, niche conditions, as well as provide additional data points to the limited number of experiments, and are used to suggest new experiment conditions on current devices [2]. High-fidelity simulations (e.g., XGC[3]) require immense computational resources and can take up to months of computation time on supercomputer-scale clusters. Both experiments and high-fidelity simulations have a high monetary cost associated with them, which emphasizes the need to extract the maximum amount of value from the available data.

For magnetic fusion, experimental data is collected during a plasma discharge or

'pulse' within a given device (e.g., the tokamak or stellarator). Physicists perform experiments by varying the device configuration then analyse the resulting effect on the plasma and use insights from the analysis to plan future experiments with the goal of maximizing their knowledge on plasma behavior. Various diagnostic measurements provide the bulk of the information about the plasma, alongside the experimental device configuration parameters.

The majority of experimental magnetic fusion data is in a time-series format, collected over the duration of a pulse. The difference in sampling rates between various diagnostic devices also leads to varying time resolutions in the collected diagnostics, e.g., Hz for Thomson scattering [4] and kHz for reflectometry [5]. The collected data is also multi-modal, spanning different levels of dimensionality: scalar (0D time-series), profile (1D time-series) to video (2D time-series) measurements (Figure 2.1). The IMAS[6] project lead by ITER aims to resolve some of the data heterogeneity issues by providing data standardization in the access layer, and is being adopted by the fusion community, with some present-day devices already providing their data natively in IMAS format(e.g., the WEST[7] and JET[8] tokamaks).

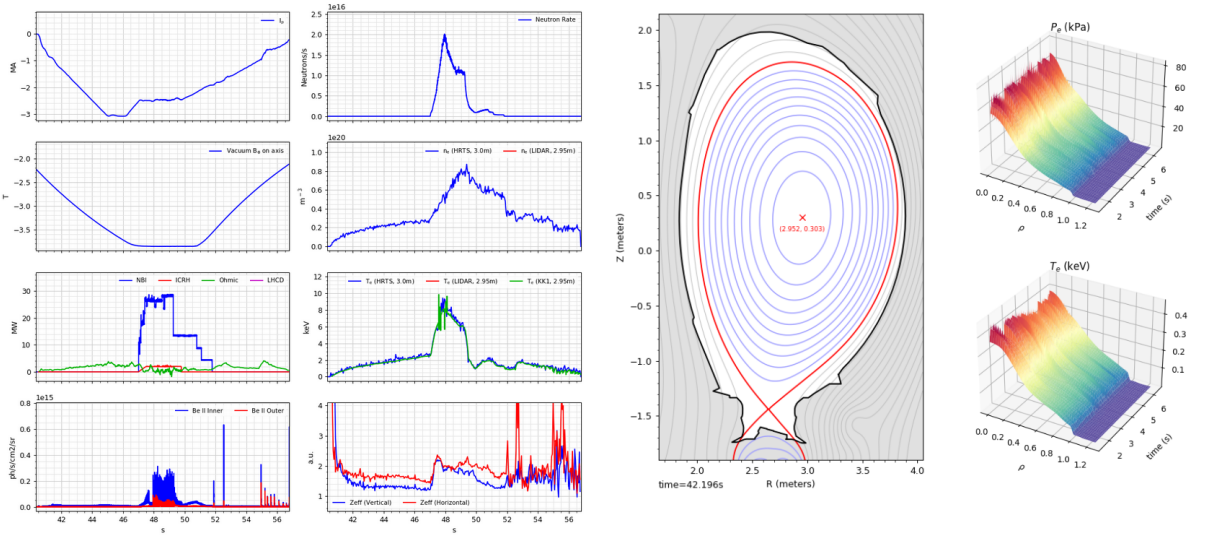

Figure 2.1: Examples of Experimental Fusion Data

After a discharge, the raw experimental data is typically calibrated automatically [9] and sometimes manually by diagnostic operators (RO's) before becoming available to the research community, and this calibrated representation serves as the baseline source data for analysis. In some cases, the available data may be identified as erroneous, requiring a re-calibration. The presence of these re-calibrations, and the consequent lack of guarantee that source data was accurate, present a challenge for ML data curation, and the need for stricter data provenance for tracking the path and changes of source data from source to results [10].

In addition to the already difficult landscape of heterogeneous data, the independent efforts of processing source data by different institutions have led to the adoption of facility-specific meta-structures and taxonomies for the collected data points. The different naming conventions make it additionally difficult to combine data from multiple sources. This issue is also being addressed by IMAS[6].

Experimental data is typically stored at the same facility which operates the device performing the experiment. Accessing this data requires authorization from the facility, and with the current lack of centralized data access, this presents an additional challenge when collecting data across multiple facilities.

Simulation data is also scattered geographically, as simulations are performed across a wide range of HPC environments, and their results can be stored in arbitrary formats [11]. There is currently no adopted standard for simulation data formatting or cataloging, making it difficult to make use of the bulk of existing results. The SimDB[12] project aims to partially address this issue, with the main goal to collect over 2000 existing, and any future ITER-related simulations in a single system.

Both of these sources can also produce very large volumes of data. Each individual pulse contains a a large quantity of multi-model diagnostics, many of which are collected at very high time resolutions. ML datasets containing hundreds or thousands of such pulses can easily reach terabyte scale today, and petabyte scale in the future. Once ITER becomes operational, it is estimated to produce up to 2.2PB of data per day [13], with the majority representing video data.

The challenges of both experimental and simulation data and their current storage layer are summarized in Table 2.1.

Another way to summarize the characteristics of fusion data is through the framework of three Vs of big data[14]; volume, velocity, and variety. In terms of volume, fusion data is projected to reach petabyte-scale data production with ITER[13]. In terms of velocity, fusion data is generated in high throughput batches from experiments or simulations, rather than a constant high velocity stream. Finally, the variety of fusion data is high, both due to the varied types of recorded measurements ranging from scalar observations to camera data, all commonly in time-series format with varying time steps, and due to differences between data formats adopted at different facilities. The high volume, low velocity (but high impulse), and high variety represent the profile of fusion data which is being used for ML research.

Working with such data in the context of ML or big data analysis can be challenging. ML researchers are estimated to spend up to 70% of the research effort on data curation[11]. In addition to the difficult dataset curation process, the lack of ML-specific infrastructure leads to a suite of additional challenges.

Table 2.1: Challenges of Source Data in Fusion

Challenge	Description
Geographically Scattered Data	Both experimental and simulation data is produced across a number of experimental and HPC facilities scattered around the world [11]. This makes collecting data for ML training increasingly difficult, and often requires data to be copied to the local system used for ML training.
Heterogeneous Data Access Systems	To address the research needs, different fusion research facilities independently developed proprietary data access protocols and tools, making it difficult to work with data from multiple sources.
Heterogeneous Data Formats	While there have been recent efforts to standardize fusion data formats under IMAS, a large portion of data still exists in various pre-IMAS formats. Additionally, simulations typically do not have standardized inputs and outputs in regards to each other or experimental data formats.
Varying Sampling Rates	Varying rates of data collections causes combining time-series data for ML applications to require additional pre-processing (e.g., interpolation of lower sampling rate data).
Source Data Re-calibration	Upon human inspection, data can be identified as erroneous, requiring a re-calibration of the data which may have been used in existing big data analysis or ML efforts.
High Volume of Data	When performing ML or other big data analysis over thousands of plasma discharges, the cumulative amount of data can present a significant challenge. ITER is projected to record up to 2.2PB of data per day [13], providing a glimpse of the scale at which fusion data will exist in the future.

2.2 Challenges for Data-Driven Fusion Research

In addition to today's intrinsic challenges of conducting fusion experiments and analysis listed above (Table 2.1), there are also challenges specific to data-driven workflows faced by current ML researchers. Currently there is a lack of standardization for facilitating the growth of ML research. As the presence of data-driven methods in fusion research continues to grow and mature, the challenges of working with data across a heterogeneous landscape become a more urgent problem which requires addressing.

The main goal of applying data-driven methods to fusion is to maximize the insights that can be extracted from existing data. Some of the more common goals of modern data-driven models are: discovering complex non-linear behaviours between variables [15], constructing accurate models of plasma fusion behavior for use in digital twins / control systems [16], or developing ML-based surrogate models that can replace elements of high-fidelity simulations [17], among others. These results can provide valuable improvements to physics insights, device control applications, as well as simulation speeds, and are of high value to the community. Due to the high impact and physical nature of the analyzed data, any ML models aimed to be used in practical experiment control or device design projects need to be reliable. This requires the highest possible degree of interpretability, explainability, and reproducibility, which further highlights the need for stricter and more standardized data workflows.

Many major challenges of data drive-fusion research are related to the typically monolithic ML research process in fusion (researchers perform the bulk of data wrangling and ML work independently), which limits the propagation of source metadata to the results, obfuscates the data wrangling process, and limits the sharing of artifacts such as clean ML-ready datasets. This monolithic approach, further defined in Section 3.2, is a result of complex data access, the need for complex data transformations, and the lack of standardized processes for ML artifact sharing, publishing, and collaboration. These challenges are summarized in Table 2.2.

The available source data access layers today are not focused on presenting bulk data collections for direct ingestion into ML workflows, i.e., magnetic fusion data is typically presented in single pulse format. While extensive metadata exists at the source, the lack of support for bulk datasets, heterogeneous formats across facilities, and the monolithic ML process, often lack the means to fully propagate this metadata to the final ML datasets and derived results, limiting the resulting provenance information. Lacking standardization for bulk metadata, and support for bulk data operations, remain as a challenge for ML research today. The existing user interfaces for data discovery are also focused primarily on individual pulse data, limiting their use for bulk data discovery desired for ML purposes.

The difference between source data formats and ML formats introduce the need for

Table 2.2: Challenges in Fusion Big Data Analysis

Challenge	Description
Monolithic and Siloed ML Research	Due to the siloed ML research efforts, many potentially useful checkpoints such as the dataset preparation code or clean ML-ready data often remain un-shared.
Lack of Data Discoverability	Even for existing data, the lack of major data sharing platforms [11] in the complex landscape of fusion data makes it difficult to identify and procure desired ML research data.
Limited Provenance Metadata	Due to the monolithic ML research process, it is often difficult to trace the results of data-driven research back to source data. This poses a challenge to reproducibility and interpretability.
Lack of Recommended Datasets	While several exist in the fusion community, there is a strong need for the creation of additional curated datasets [18]. Recommended datasets, one such example being the ITPA disruption database [19], create an opportunity of collaborative development of data-driven solutions. They also create a reference for result benchmarking and comparisons. Currently, there is a lack of standard practice for creating these kinds of datasets.
Need for Interpretability and Explainability	The context of physics-based observations, expensive experiments, and high-impact practical applications produces a strong need for model interpretability and explainability.

data transformations. ML workflows typically include a transformation step converting large amounts of source data into ML-ready feature vectors. Extracting ML feature data from source data can be a nuanced process, and, without available access systems for on-demand data transformations and queries, requires researchers to perform these complex data wrangling procedures on their own. The offloading of data processing codes onto ML researchers leads to a monolithic ML process. This monolithic process can obfuscate the path between raw data and final results if not adequately documented. Due to this research approach, various artifacts from this process such as the ML-ready clean feature vectors (ML-ready datasets), or trained models, often do not have standardized publishing or discovery methods (exceptions exist, such as QLKNN [17]).

Creation of recommended datasets, in other words common datasets re-used for specific research problems, has been highlighted as a critical challenge in the wider plasma science domain [18]. These datasets present a possible method of addressing the data curation challenge. By establishing a collection of curated ML-ready datasets, ML researchers can easily source data without the need for ad-hoc monolithic data collection and transformation. Recommended datasets also enable referenceability for ML training data, and easier benchmarking of results.

To address these challenges, the development of a fusion data platform has been outlined as one of the priority research opportunities in fusion [11, 18]. The remainder of this thesis aims to evaluate possible approaches for the development of a fusion data platform.

2.3 Data Platform Goals

The main goals of the platform are to address the challenges related to the heterogeneous data landscape, monolithic ML approaches, and limited ML infrastructure defined in Sections 2.1 and 2.2. Additionally aiming to integrate the platform into the existing landscape of processing and storage tooling in order to maximize the applicability of the platform to both existing and future research efforts. Several goals for a fusion data platform have previously been outlined by previous reports such as enabling a federated data repository, providing provenance information, improving reproducibility, and versioning data products as well as data processing pipelines [11].

The current monolithic process often involves developing ad-hoc solutions to data wrangling challenges. The data platform should provide more standardized methods for working with ML data. Standardizing research practices enables easier collaboration and resource sharing. Another adjacent goal to standardization is to enable data FAIR-ness which has been highlighted as a priority[10].

The historically limited data sharing has resulted in a similarly limited set of com-

mon datasets for establishing model benchmarks. The platform should leverage a standardized approach to ML data workloads and publishing for establishing a larger set of common reusable datasets. Creating common datasets provides the grounds for model comparisons and improved reproducibility of results. One example where a shared dataset would allow model benchmarking and improve reproducibility, is disruption prediction where 4+ independent studies have been performed on proprietary curated datasets without any cross-comparison [20, 21, 22, 23].

The currently limited reporting of provenance metadata is another challenge which needs explicit addressing in the platform development. The data platform should enable a standardized method for recording dataset provenance in terms of utilized source data, as well as referenceability such that resulting publications can easily reference utilized datasets. This likely stems from the monolithic approach to ML fusion research.

Finally, the platform should address the data access needs of ML research, providing simple and sufficiently fast access to ML data.

A summary of the high-level fusion data platform goals is shown in Table 2.3.

Table 2.3: Fusion Data Platform Goals

Goal	Description
Standardizing ML Data Sharing and Discovery	The data platform should enable easier publishing/sharing of data, with a focus on ML-ready datasets. It should also provide an overview of available data and datasets which is easily accessed by ML researchers. This discovery interface should also include sufficient data search and filtering capabilities.
Improving Provenance	Sufficient provenance metadata to accurately link raw data to practical results should exist within the platform, with support for multiple versions of both raw data and results due to re-calibration possibilities.
Aligning with Data FAIR-ness Goals	The platform should aim to make the data within it as FAIR as possible.
Improving Result Reproducibility and Interpretability	The data platform should provide metadata and details about the datasets and data pre-processing to improve reproducibility and interpretability of data-driven research and its results.
ML-appropriate Data Storage and Access	The data layer of the data platform should be scalable with the growing volume of fusion data. The data platform should also improve on the currently available data access layer for ML and big data analysis use cases.

2.4 Related Research

Within the fusion community there have been several reports covering the general state of data-driven research and outlining future development needs. Two major reports are the 2019 workshop report [11] exploring the needs of data-driven and ML fusion research and the 2022 review of data-driven plasma science [18] focusing on issues in the wider scope of plasma physics. These reports outline high-level requirements for enabling ML fusion research, one of which is the development of a fusion data platform.

There have been several recent publications addressing this identified gap in big-data tooling from within the fusion community. An example of a new data access platform has been presented in a recent paper from the EAST tokamak, which aims to address some of the challenges present in current data access systems, with a focus on bulk ML data storage and access [24]. Beyond data storage and access, there also exists an effort

to develop an entire set of big-data workflow tooling by Lawrence Livermore National Laboratory (LLNL) including a custom workflow definition system (Maestro), workflow scheduler (Merlin), and HPC provisioning layer (Flux), focusing on a wide range of ML workflows, not limited to data management [25].

Another major project from the fusion community is Fair4Fusion[10], which emphasises the need for FAIR (findable, accessible, interoperable, reusable) data, and suggests a high-level architecture for a data sharing platform focused on open metadata and cataloging.

Outside of the fusion domain, existing academic efforts have lead to the development of research data platforms, for example in the REANA [26] platform designed in the context of particle physics, and the MONTRA [27] platform developed by the biomedical community. Both of these platforms aim to tackle similar issues in re-use, sharing, and discovery of research data in order to improve research output and quality. Both of these platforms can be described as a custom interface between end-user researchers and lower level open-source technologies such as workflow orchestration, containerization, bulk storage systems, etc. Many other systems are listed, although not explored in-depth, in a recent 2022 report by the European Commission [28]. Many of these systems include a modern browsing/catalog interface similar to Zenodo [29].

Overall, a common approach for enabling data-driven research involves creating or deploying new software solutions. In the context of this thesis, possible architectures for a fusion data platform are explored through a subset of available open-source software components.

3. Data Platform Architecture

This section discusses the goals and possible approaches for new data infrastructure based on the challenges outlined in Sections 2.1, 2.2, and high-level goals from Section 2.3. The goals form a high-level basis for the development of a fusion data platform, and this section considers the potential hardware-level architectures required. This chapter introduces several possible architectures, and defines the basic elements on which the subsequent case-study implementation was designed.

Section 3.1 discusses the abstracted components present in ML workflows in terms of transformations and forms of data. Following this overview, Section 3.2 describes the current state of ML workflow in the context of the identified components and their distribution across physical devices (i.e., where the data is stored vs where the data is transformed). Section 3.3 then describes possible future arrangements for these components based on existing and newly proposed data platform ideas. Finally, Section 3.4 describes how adding a metadata repository can enable a data platform architecture capable of incorporating heterogeneous ML dataset sources into a single platform.

3.1 Data Flow Components

A simplified representation of the data flow present in fusion based ML workflows is defined. Four main components are identified in the end-to-end flow of data from it's source format to ML model training (Figure 3.1):

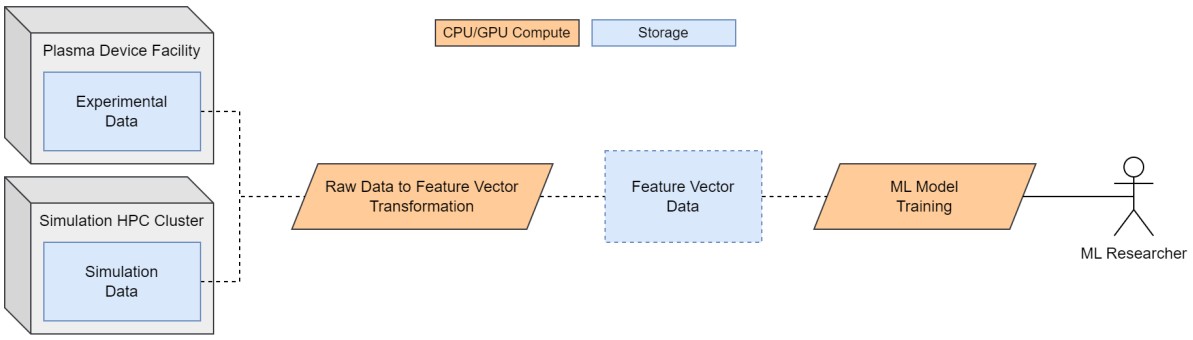

Figure 3.1: ML Data Flow Components

1. Source Data - Source data represents the existing experimental and simulation data which is stored across a number of facilities with heterogeneous formats, databases, and access layers.

2. Source Data to ML Feature Vector Transformation - This transformation is necessary because of the difference between data formats used to store source data and data formats used to train ML models. This transformation is a computational component which is currently commonly performed in the same HPC research environments used for ML training (by the ML researcher). The transformation can also be performed at the source data location (e.g., onsite cluster, if available), or, in the future, a new centralized location.

3. ML Feature Data - This component represents the ML-ready feature vector data format which is ingested by ML training workloads. If the transformation from source data to ML feature vector is performed in-line with the training code, this structured form of the data may only exist in-memory during computation.

4. ML Training - The final component of interest is the ML training workload itself. This workload requires ML feature vector data to be available. This data may be ingested from storage or computed at run-time.

Both the input of this system - the source data, and the output end-users - the researchers, are geographically distributed. Therefore, if we consider the data transformation or the ML feature data as independent components, there are three possible locations for performing data transformations:

- At the source data facilities: Transforming data at the source allows for a distributed ML data access layer and limits unnecessary data transfer. However, this approach would require significant upgrades to the current source data storage systems, and add the need for additional compute power to be available at the facility.

- At a centralized facility: A centralized approach allows data to be collected from existing sources without meaningful changes to the existing data layer, and transformed in a central location where it can be made available to ML researchers. This approach requires additional resources for the deployment of such a centralized ML data platform, and retains the need to transfer bulk data from the source facilities to the centralized facility.

- At researcher's device: The current approach, where researchers collect source data and perform ad-hoc data wrangling in their own environment. This approach does not require any additional shared resources, however it limits the collaboration and

sharing of results, giving rise to the identified challenges associated with current ML practices in fusion (Section 2.2).

A soft constraint present in this system is that at time of ML training, utilized data should be available at sufficient speeds to fully utilize the available CPU/GPU capacity, this typically implies having the data available on a high performance file system available to the ML processing node.

3.2 Current ML Research Architecture

In the context of the identified components, the current typical ML research process can be described as a monolithic approach leaving all four components up to the researcher(s). This means that the entirety of the ML research process is distributed across the researcher's devices, with little to no sharing of codes or processed data. Typically this process involves making a copy of source data onto the researcher's available device. Then, an ML training code is developed including both the data transformation and ML model training steps. During this process, the clean form of ML-ready data may be stored local. This flow of ML data can be modeled using the previously identified components from Figure 3.1 by placing the data transformation, ML-ready feature data, and ML training onto researcher's own devices, along with a copy of raw data being made for adequate data access speeds (Figure 3.2). It is worth noting that even for cases where multiple researchers perform work on the same device, there is no standardized infrastructure to reduce the potentially duplicate steps, and their workflows are typically completely independent unless explicitly coordinated.

This monolithic approach is born out of practicality, as little other infrastructure currently exists to ease this workload. However, this approach also gives rise to the issues outlined earlier in Table 2.2 such as siloed ML research efforts and limited data provenance. To address these challenges, one approach is to move the data transformation process into it's own system, to enable greater re-use of ML feature data, and to reduce the amount of "double" work to be done by ML researchers. By extracting the data transformation into a separate unit, it is possible to establish standard practices for recording provenance information, publishing ML-ready data, and automating elements of the ML data curation workload.

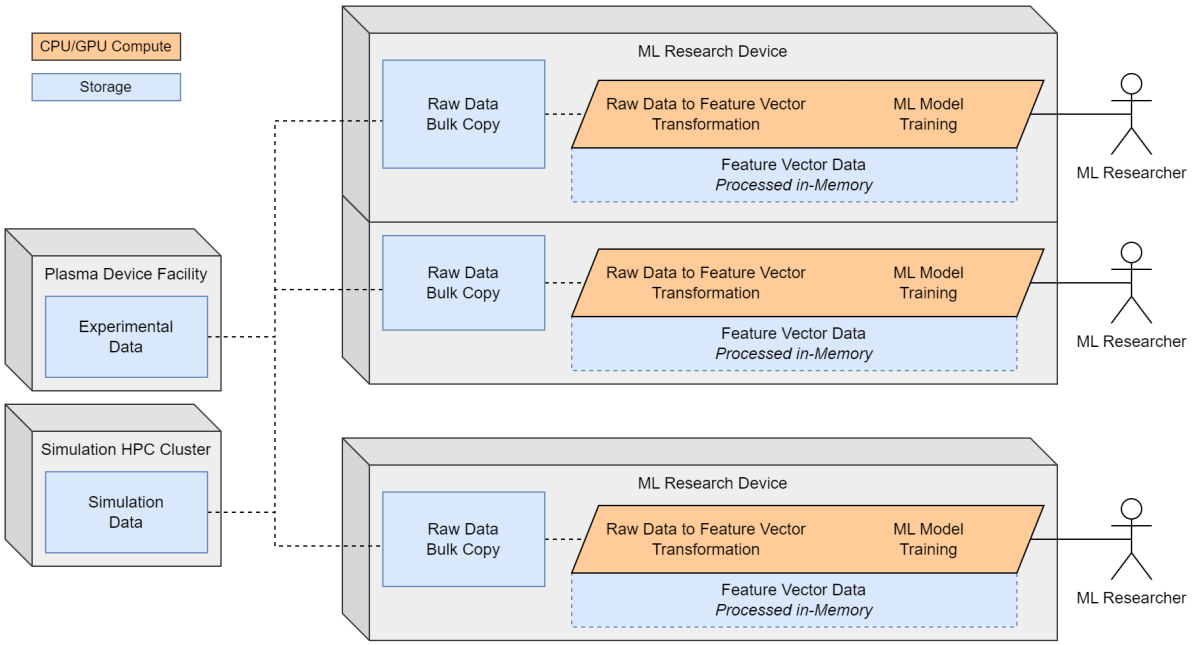

Figure 3.2: In the current ML data workflows, data is typically copied from the source (after an often difficult data discovery process), and the end-to-end ML process is performed in the researcher's siloed environment as a monolith.

3.3 Possible Alternative Architectures

3.3.1 ML Feature Query Approach

As outlined by the fusion needs report [11], a suggested set of features for the fusion data platform would include the ability to define datasets as an on-demand transformation of existing raw data cross a wide range of existing storage locations, formats, and even on-demand simulations. This kind of approach aims to minimize any unnecessary physical transfers of data by distributing the data transformation step across the available existing source data infrastructure, and enable researchers to simply access bulk datasets with fine-grain control over the loaded features and their respective fidelities in order. This kind of query would require complex planning algorithms for execution across hardware and services spanning multiple compute clusters and data centers. This approach, with sufficiently optimized query planning, would allow for ML-ready data to be procured on-demand without unnecessary data transfers, without duplicate data storage, and with near-optimal in-line data preprocessing at source, instead of target locations, provided that the hardware and software capabilities exist at the data source facilities.

This approach places the data transformation onto the source facility hardware, with support for more complex data querying, with a new central query engine component which either works in a distributed manner across the source facilities, or requires

designated centralized hardware (Figure 3.3).

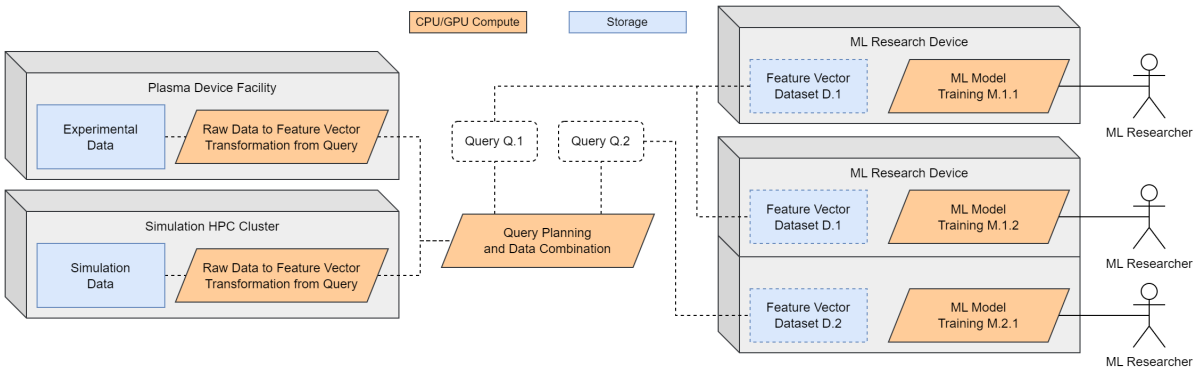

Figure 3.3: In the query-based approach, the data access layer includes the ability to transform data into ML-ready formats. In this case, the transformation computation needs to be located at the source data devices, and an additional system is needed for planning the data query execution.

This approach requires handling data processing at the source data locations, as well as a complex query planning system. Designing such a system would require complex changes to the source data facility infrastructure, thus is beyond the scope of this thesis. The approach explored in this work instead focuses building an architecture around the existing infrastructure.

3.3.2 Persistent Dataset Approach

The alternative to the query system is to rely on discrete transformation workflows. By considering the dataset collection as a consumption of the existing raw data layer, it is possible to create re-usable workflows which can collect data from multiple sources and then perform required data transformation steps at a centralized location. This centralized location will be on a device hereafter referred to as the ML Data Platform. The first major issue that can be observed between this approach and the query-based approach is the need to consume raw data directly. This introduces major overheads for data transfer between the source data facilities and the ML Data Platform facility. Without making changes to the data access layer, there are not many options for reducing this overhead. However, with the low cost of storage, it may be more feasible to create a persistent storage layer for processed datasets resulting from these discrete workflows. This storage layer, being a new component to the system, could be designed in line with the needs of ML research, with a focus on high performance access speeds and richer data queries.

By making the data transformation and ML data storage components independent from individual ML research projects, it becomes possible to create re-usable datasets which can be accessed by different research teams exploring different ML approaches for the same goal (Figure 3.4).

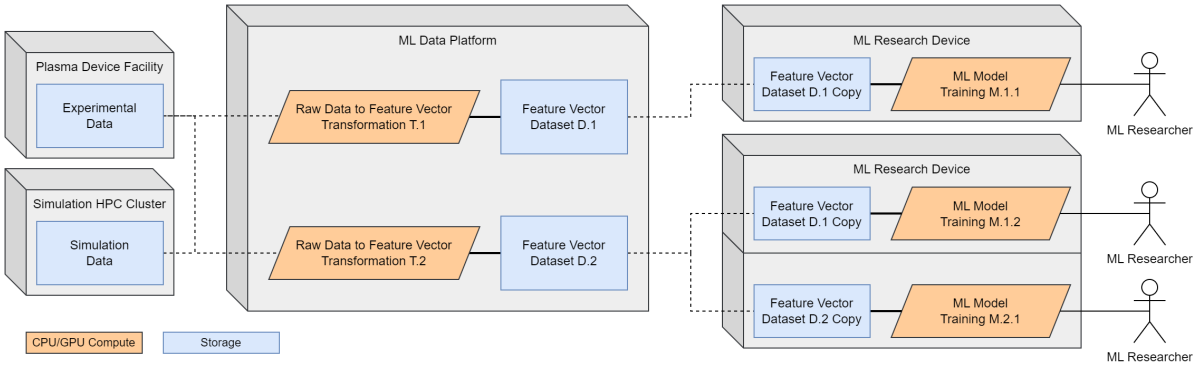

Figure 3.4: In this example, the data transformation is placed in a new centralized device, which enables the collection of data from existing sources and its transformation into discrete ML-ready datasets. The ML-ready datasets are then exposed to the community and copied by ML researchers with ease.

A second challenge arises in this architecture from the previously identified need for high-performance data access at the research locations. In this case, the simplest approach would be to copy the ML dataset to the researcher's device prior to ML training. Researchers can utilize temporary HPC storage resources such as the scratch disk to accommodate this data copy. However, with the expected growth of ML datasets in fusion, the size of an individual dataset might require a very long time to copy. To avoid creating copies of ML datasets, one approach is to centralize the ML research itself at the same location to where the ML dataset is stored (Figure 3.5). This architecture reduces

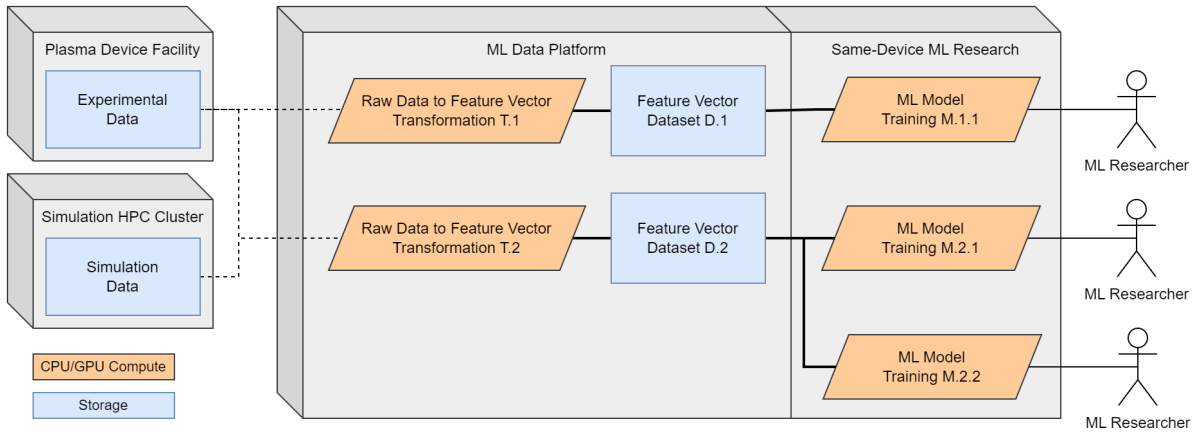

Figure 3.5: By further centralizing the ML research process, the training process can also be placed on the central data platform device, enabling direct high-throughput access to the ML datasets, and removing the need for ML dataset duplication.

the need to transfer and store copies of ML feature data, however it introduces the need for suitable CPU/GPU compute power for ML research to be available as a part of the ML Data Platform. Without any additional components, this approach would also prevent external researchers who do not have access to the ML Data Platform from performing

ML research on the available datasets. One possible solution to this is to introduce a metadata layer, that captures and shares the underlying transformation pipeline that created the datasets.

3.4 Introducing a Metadata Layer

The architectures discussed in Section 3.3 primarily discuss the designation of storage and compute resources across physical devices. However, by extending these primitives through metadata and cataloging, it is possible to leverage a combination of these architectures, and create a more flexible system.

In order to facilitate ML-ready datasets as a widely available and reusable data product, it is important to provide researchers with a centralized portal for data discovery. Such an portal would serve as a catalog of available datasets, and should ideally enable searching and filtering data, as well as allow for data versioning and unique references. A metadata record can represent a centralized persistent dataset, it can represent an external dataset provided by a source data facility, it can even represent a code workflow or query for materializing data on-demand. Since all that is being changed is the addition of a new, lightweight, component to the system, it does not impede existing benefits such as the ability to centralize ML data and compute to the same device, however it enables external producers and consumers of data to be integrated into the same central discovery system.

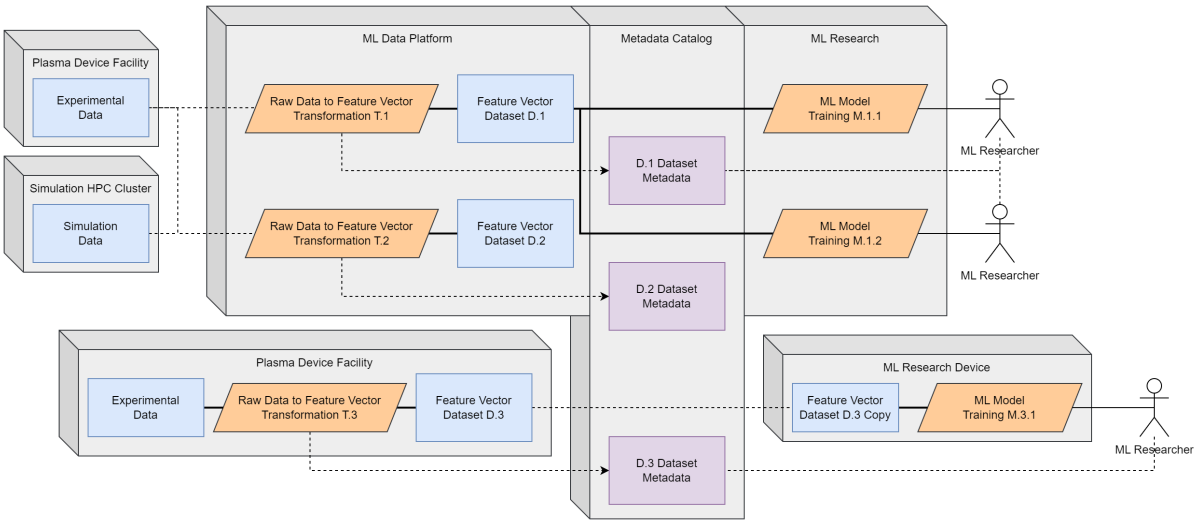

Figure 3.6: Through the use of a metadata layer, data from various sources can be cataloged and discovered in a central location. In this example it is showcased how local and remote persisted datasets can be accessed by both local and remote researchers. Additionally, the catalog could include a variety of sources including the previously discussed query endpoints.

A central device would house the metadata repository, which would enable the cataloging of both locally produced datasets, external datasets, or any other sources of data compatible with the chosen metadata record format. These ML-ready datasets could then be analyzed locally (for local datasets), or downloaded as needed (Figure 3.6).

The ability to create a ML-ready data catalog introduces the additional need to reliably generate and record metadata, as well as the need for interfaces to enable metadata record browsing and discovery.

This section introduced two approaches to creating a data platform: the query-based approach, and the persisted dataset approach, along with the introduction of a central metadata catalog for aggregating heterogeneous ML data sources. To further evaluate the feasibility of the metadata-enabled architecture, a case-study was performed to create a proof-of-concept data platform. This platform is based on a limited implementation of the metadata-enable architecture, demonstrating primarily its usability as a centralized data platform. The creation and evaluation of this platform is described in the following section.

4. Case Study: Proof-of-Concept Data Platform

In this case-study section, a proof-of-concept implementation of a data platform specifically designed for the fusion research community is presented. The proof-of-concept platform is based on the metadata-enabled approach outlined in Section 3.4. The development of this platform aims to address the unique challenges and requirements inherent to the field of fusion research, which demands a robust, scalable, and efficient data management infrastructure which can easily integrate with existing domain-specific tooling.

This objective is pursued through an evaluation of available open-source tooling, combined with the practical deployment of selected solutions. The deployed system is then evaluated in collaboration with domain-expert fusion researchers. This approach ensures that the developed platform remains not only theoretically robust, but also practically viable and relevant to the needs of the end-users within the fusion research community. The proof-of-concept data platform presented in this case-study is designed to serve as a blueprint for future development.

In the subsequent sections, the methodology employed in the development of the proof-of-concept data platform will be outlined, detailing the open-source tooling utilized, the software deployment process, and the evaluation strategy. Additionally, the insights and findings will be discussed, highlighting the ways in which the platform addresses the unique challenges and requirements of the fusion research community. Section 4.1 describes the functional requirements for the proof-of-concept platform and Section 4.2 outlines the resources available for this case study. Together these two sections outline the basis on which the platform is designed. Section 4.3 describes the main components of the platform and Section 4.4 discusses selected open-source tools for the implementation. Section 4.5 goes into the details of the deployment of the platform and the development of custom tool extensions. Finally, Section 4.6 presents the evaluation of the platform, the end-to-end trial, encountered challenges, and platform resource requirements.

4.1 Establishing MVP Criteria

Based on the challenges defined in Sections 2.1, 2.2, and derived high-level goals defined in Section 2.3, a list of feature requirements is defined for the Minimum Viable Product (MVP) of a proof-of-concept data platform. The platform is also designed to emulate the metadata-enabled architecture outlined in Section 3.4.

The goal of this platform is to enable the publishing and sharing of ML-ready datasets in a centralized location. To satisfy the need for referenceability, the platform should include unique references for individual datasets. As the datasets may be subject to changes over time due to source data re-calibration or changes in the data transformation script, it is necessary to accommodate the possibility of different dataset versions. Different versions should also be uniquely referenceable to enable accurate provenance down the line.

To record provenance of the datasets, information about features and source data used in the dataset should be recorded alongside the dataset records. This metadata currently does not have a standard format so the constraint is somewhat flexible, however there should exist a method of recording relevant information about the utilized source data.

As the chosen approach is based on a centralized data platform, data collection and transformation scripts form the basis of the available datasets. There is an effective 1:1 mapping from each data transformation code to a task-specific dataset. We can label this approach as a datasets-as-workflows approach. The data platform should therefore include a standardized method for workflow creation, such that workflows become a maintainable and reusable asset.

The platform should also aim to handle source data changes by notifying researchers and/or auto-generating new versions of the dataset based on updated source data. To enable this, the platform needs to support either push based (e.g., via. API, event-driven) or pull-based (periodic checks) triggers for handling such events.

The platform should be easy to adopt, as research experts should not be expected to spend large amounts of effort on adopting the platform. There is no simple way to measure the difficulty of using a platform, however the goal should be to minimize the amount of changes to the researchers' existing workloads.

The platform should be possible to deploy on HPC or vendor-agnostic cloud resources. Tools designed for use on commercial cloud resources such as AWS, GCP, or Azure are not suitable in this context. Additionally, the chosen open-source tools should be mature and consistently maintained, such that their use does not pose a risk of near-term obsolescence.

The refined list of requirements for the data platform is shown in Table 4.1.

Table 4.1: Fusion ML Data Platform Minimum Viable Produce Requirements

Requirement	Description
Unique Data References	Datasets present in the platform should have unique reference identifiers (URIs) which can be used in ML research and publications to easily trace the original data
Data Versioning	Source data, data processing workflows, and dataset scope are all subject to change. Therefore, it is imperative that multiple versions of a single dataset can exist within the platform, such that each version can be individually referenced, and individual references remain immutable.
Provenance Metadata	Exact details of what raw data is being processed in the creation of the dataset should be embedded in an accessible and interoperable format alongside the dataset records.
Reusable Data Collection Workflows	The centralized platform should have a standard way of defining data collection and transformation workflows so that they may be re-used (e.g., in the event of data re-calibration).
Automated Handling of Source Data Re-calibration	In the event of source data changes, provenance information should be used to determine which datasets are affected. The platform should be able to trigger tasks based on these events, such as notifying researchers or generating new versions of the affected datasets.
Independence from Public Cloud	The chosen tools should not force the user to rely on commercial cloud services. It must be possible to deploy the tools on common infrastructure such as Docker, Kubernetes/OpenShift, HPC resources, or bare-metal.
Easy to Adopt	Since the platform is intended as a proof-of-concept, it should be easy for users to understand and quickly integrate with existing workflows for testing.
Tools with Strong and Active Development	The chosen tooling should be under active development with a rich ecosystem of users. This makes adoption and debugging a lot easier through existing community resources.

4.2 Available Resources for Platform Deployment

This section introduces the computational resources available for the deployment and evaluation of the proof-of-concept fusion data platform. Both computational and storage resources were necessary to facilitate the evaluation of chosen tools. In addition, example data was also needed to evaluate the end-to-end workflows of the platform.

The primary set of resources available for this research was provided by CSC*. The following resources were primarily considered for the data platform:

- **CSC Puhti/Mahti [30]** - HPC compute clusters running SLURM
- **CSC Rahti [31]** - A container orchestration platform based on OKD
- **CSC Allas [32]** - A CEPH-based object storage / data lake service

For this case study, the full scale deployment of the platform was performed using CSC Rahti, the OKD Kubernetes service. This allowed for more accurate evaluation of how end users would work with the platform.

The data used for evaluating the platform was a dataset used in recent ML-driven research paper[15]. This dataset allowed for a full end-to-end trial of the platform, as well as practical feedback from the original author. The example consisted of a data-wrangling Python script, and a resulting data set containing clean NumPy[33] array data for a large number of plasma pulses sourced from the JET tokamak[34].

4.3 Case-Study Data Platform Components

In this section, the metadata-enabled architecture defined in Section 3.4 is further defined in terms of available software tooling categories while adhering to the MVP requirements outlined in the prior Section 4.1.

4.3.1 Metadata Platform

The primary tool that has been identified as a key enabler for collaboration and acceleration of dataset creation and usage is a comprehensive and feature-rich metadata catalog, hereby referred to as a data discovery portal. Unlike existing current interfaces designed to preview calibrated data of individual pulses, the goal is to create a single interface for discovering existing ML-ready datasets. This interface would also serve as the ideal location for publishing reference datasets. This component would include the

*CSC Project #2005083

storage of metadata records associated with datasets, providing provenance information, as well as unique resource identifiers (URIs) that can be used to reference the dataset in later results. The goal of the metadata platform is to enable unique data references and versioning, while extending it with an intuitive user interface for dataset search and discovery.

Metadata cataloging tools are not a clear cut category. Depending on the context, different tools can be employed to manage the indexing and discoverability of resources (e.g., data). The main interest in this case study are tools which are used to document data, datasets, and workflows. Additionally, tools used to document provenance are also an important consideration. From a general perspective, many of these tool are designed to integrate into the public cloud computing ecosystem. However, that is not a compatible approach to the desired system which is designed to fit into the existing fusion landscape of HPC and proprietary tooling.

A sub-category of data cataloging tools are data management platforms[35] (DMPs), also sometimes referred to as research data management[36] (RDM) tools. These are platforms which aim to address the need for data cataloging and publishing in academic environments, and could provide a better fit to the needs of the case study platform. Zenodo [29], a popular open platform used for dataset sharing, is an example from this category. The challenge remains to identify suitable general purpose metadata cataloging tools which remain under active development and meet the defined needs of the case study.

4.3.2 Workflow Orchestration

In addition to a data discovery portal, the platform should enable the re-use of data processing codes used to create the published datasets. A tooling category which enables the creation and execution of workflow resources are workflow orchestration managers. By enabling workflow re-usability it becomes possible to automate the propagation of source data changes by generating new dataset versions. Workflow orchestration has been identified as a potential solution for tackling data provenance, as standardized workflows and their execution can automatically produce valuable metadata for establishing data provenance. The standardization of data processing workflows can provide additional benefits in re-usability and longevity of previously one-off data processing codes. Workflow orchestration platforms often require additional configuration in addition to the regular code, however they can provide a rich set of features for managing the codes execution and monitoring, as well as provide the necessary interfaces for event-driven execution of workflows for example to automatically re-generate datasets in the event of source data changes.

4.3.3 Component Architecture

A mapping can be made from the architecture defined in Section 3.4 to the proposed metadata catalog and workflow orchestration components (Figure 4.1).

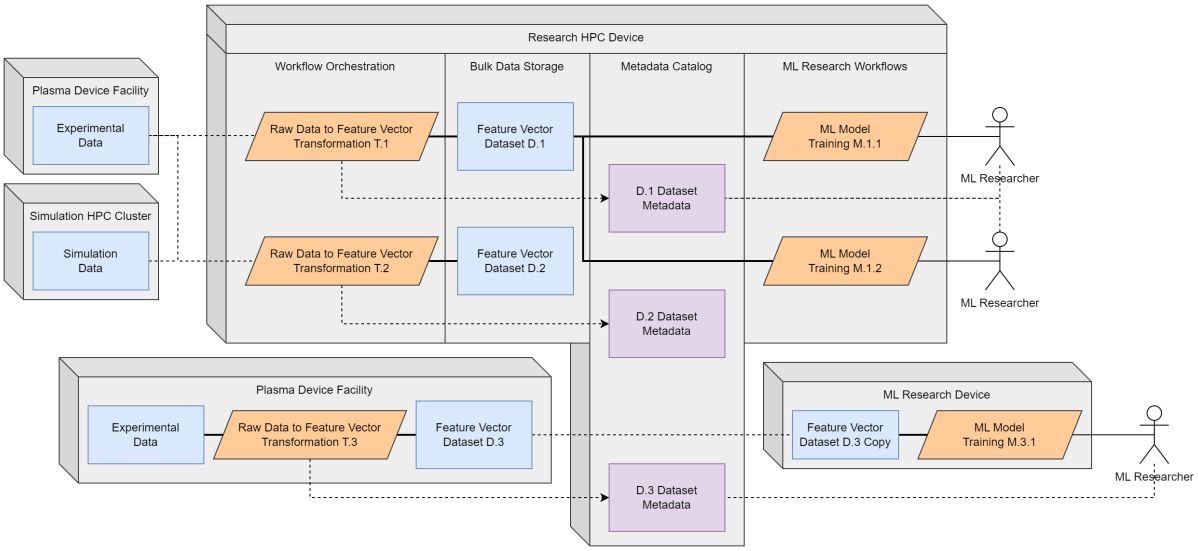

Figure 4.1: The figure demonstrates how the identified components are mapped to the architecture outlined in Section 3.4. The workflow orchestrator manages the data transformations and publishes resulting metadata and clean ML data to the metadata cataloging and bulk data storage components respectively.

The belief behind this platform is that these challenges can be addressed through two major improvements. Firstly, the data wrangling workflows currently performed by ML researchers in fusion are a valuable asset which can be re-used to generate data long after the individual research project has concluded. And secondly, the clean data used to train ML models should be published and catalogued in a central (or per-facility) location coupled with a rich interface to enable discovery of existing data. This catalog would also serve as the ideal location of recommended datasets, as well as a source of unique references for citing source datasets in derived results.

4.4 Evaluating Existing Open-Source Data Tooling

In this section, a set of open-source tooling for metadata cataloging and workflow orchestration are identified and evaluated in the context of the MVP requirements identified in Section 4.1. These components were previously introduced in Section 4.3.

4.4.1 Metadata Platform

A set of core characteristics is defined for evaluating potential metadata in regards to their ability to address the needs of fusion ML research (Table 4.2). A graphical user interface (GUI) is required to enable a clear overview of available ML datasets allowing fusion researchers to quickly identify relevant data and potential recommended datasets. Due to the possibility of data re-calibration or changes to the data transformation scripts, resulting datasets may evolve over time. To address the support for both clear data provenance as well as evolving datasets, versioning is necessary along with uniquely referenceable dataset versions. To adhere to FAIR and Open Data goals, records should adopt common metadata formats. One of the simplest of these formats is Dublin-core[37], a basic set of metadata fields for describing digital or physical resources. To enable faster discovery, and a more scalable platform, the system should include full search capabilities, ideally through the use of a dedicated search engine. To support integration between tools, it is important to note what approaches to metadata collection (push vs. pull) are supported by the platform. Finally, some of these tools may also include their own dedicated bulk data storage layer, however the scalability limits and supported data formats can limit their usability.

Table 4.2: Data Catalog Features

Feature	Description
F.1 Browsing Interface	A complete overview of available data is available through a GUI.
F.2 Versioned Records	Multiple versions of a record are supported and are uniquely referenceable.
F.3 Dublin-core Metadata	Record metadata adopts the Dublin-core standard.
F.4 Search Engine	Records can be searched via a dedicated search engine.
F.5 Push-based Ingestion	Record information can be submitted from external sources to the platform.
F.6 Pull-based Ingestion	Record information can be queried from external sources by the platform.
F.7 Bulk Data Storage Layer	Bulk storage is included as a part of the platform and can be used to store data alongside metadata records.

CKAN[38], InvenioRDM[39], Dspace[40], and Dataverse[41] DMP/RDM solutions been selected as possible candidates from existing surveys[35, 36]. Additionally, more generic open-source solutions OpenLineage[42]/Marquez[43] (Marquez is a reference meta-

data platform implementation on the OpenLineage standard) and DataHub[44] have been selected from the Linux Foundation for AI & Data Landscape[45].

Table 4.3: Data Catalog Comparison

Software	F.1	F.2	F.3	F.4	F.5	F.6	F.7
OpenLienage/Marquez	✓				✓		
DataHub	✓	✓		✓	✓	✓	
InvenioRDM	✓	✓	✓	✓	✓		✓
CKAN	✓	✓	✓	✓	✓		✓
Dspace	✓		✓	✓	✓		✓
Dataverse	✓	✓	✓	✓	✓		✓

In addition to these tools, the Fair4Fusion[10] central metadata services can also be considered as a viable candidate for ML dataset metadata recording, as the goals of the platforms are closely aligned on data discovery, search, and recording better metadata (e.g., provenance). For this case study however, the selection was limited to the readily available open source candidates.

InvenioRDM was selected among the available DMP/RDM tools for it's modern UI, extensibility, native support for record versioning, and focus on academic records. Another favorable factor is that InvenioRDM is intended to be the future backbone for Zenodo[29], one of the most prominent academic data repositories, providing a case for long term maintenance and a sustained development effort. It is also used as the base for several other institutional repositories (e.g., CaltechDATA [46], TU Wien Research Data[47]) showcasing its usability in production environments.

4.4.2 Workflow Orchestration

Likewise, a set of characteristics is defined for evaluating workflow orchestration platform (Table 4.4). Script-based workflow definition allows domain experts to more easily convert their existing codes into reusable workflows. Presenting a GUI for workflow management lowers the technical complexity of monitoring workflow execution and managing available workflows. Scalability is an added concern for fusion workloads which may require processing high volumes of complex data. The project-based nature of academic work, including fusion research, benefits from multi-tenancy as projects can receive independent scope without interfering with other's work. Finally, HPC-native execution is an added bonus, however not many workflow orchestrators exist for that purpose today [48].

Several open-source workflow orchestration tools have been selected as potential

Table 4.4: Workflow Orchestrator Features

Feature	Description
F.1 Script-based Workflows	Workflows can be defined in-line with the data processing code (via classes or function decorators).
F.2 Workflow Management GUI	A GUI is available for monitoring and managing available workflows.
F.3 Scalability	Workflow execution is horizontally scalable across cloud or HPC resources.
F.4 Multi-Tenancy	The platform supports independently scoped workflow projects.
F.5 HPC-native Execution	Workflows are executed using HPC resources (e.g., via SLURM).

candidates for the case study data platform and evaluated against the criteria (Table 4.5). An existing comprehensive review of available workflow orchestration tools[49] was used to select Apache Airflow[50] and Flyte[51] as potential candidates. Both of these tools are also highly adopted in the industry, however they have limited usability on HPC resources. StreamFlow[52] was also considered as a candidate as it is based on the CWL[53] workflow standard, an academic vendor-agnostic workflow definition language, and supports execution across HPC and cloud resources.

Table 4.5: Workflow Orchestrator Comparison

Software	F.1	F.2	F.3	F.4	F.5
Apache Airflow	✓	✓	✓		
Flyte	✓	✓	✓	✓	
StreamFlow			✓		✓

Apache Airflow was selected as the proof-of-concept platform workflow orchestrator despite the lack of out-of-the-box multi-tenancy and HPC support. Several factors played a role in this decision. Airflow is a Python-first workflow orchestrator, aligning well with the existing Python-heavy ML workloads in fusion, especially for the end-to-end trial which is based on existing Python code. It is a very mature tool, released in 2015, with large adoption in the industry providing a large base of available knowledge, community support, and guides. It is also a very extensible platform, with support for implementing custom Operators and Hooks, explained in more details later, via simple drop-in Python

files. The robust architecture, extensibility, and mature ecosystem make Airflow the tool of choice for evaluating the core principles of workflow orchestration in this study.

4.5 Development and Deployment of the Data Platform

This section details the hands-on process of developing the proof-of-concept data platform. It covers the deployment process, encountered challenges, and development of custom tooling.

4.5.1 Selected Tooling Details

InvenioRDM

The InvenioRDM architecture (Figure 4.2) consists of several key components in order to be deployed. The core of InvenioRDM is deployed as a web app consisting of an independent backend API and static frontend component served via nginx[54]. The API backend is dependent on an array of supporting services: an HAProxy[55] instance for load balancing, a filesystem or S3 backend for file storage, an SQL database for structured metadata records, a Redis[56] cache, and a RabbitMQ[57] and Celery[58] based task queue (used for indexing, setup, and other asynchronous tasks such as sending out account activation emails).

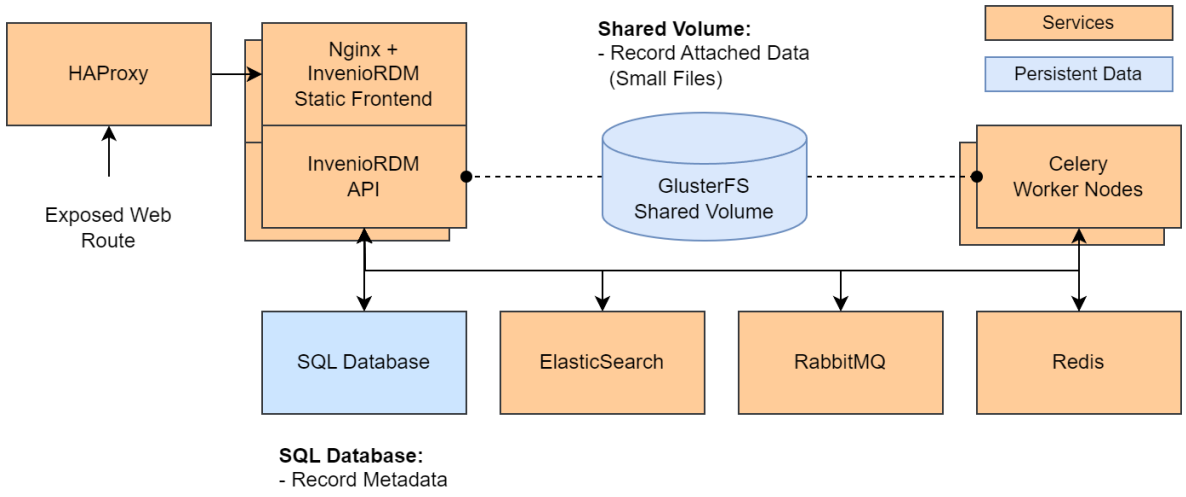

Figure 4.2: Core InvenioRDM Services Deployed on CSC Rahti

All required services are open-sourced, and the core web API and frontend can be compiled and deployed freely in any desired configuration. To assist with the setup, a

Kubernetes Helm[59] chart, utilized in this deployment, and a Docker Compose CLI setup tool are available. The Helm chart provides the ability to quickly provision resources and deploy all necessary services on top of Kubernetes. The InvenioRDM frontend and API are contained within the same Kubernetes pod, and are horizontally scalable (enabled via the load balancer). Likewise, the number of Celery worker nodes is also scalable. In this deployment, all scalable services exist in two instances.

Apache Airflow

The architecture of Apache Airflow (Figure 4.3) is based on three main components. A frontend webserver, a scheduler, and and a set of worker nodes make up the core of the system. The webserver provides a Web GUI for managing workflows, the scheduler provides an API for scheduling and monitoring workflow execution, and the worker nodes are used to execute workflow tasks. These worker nodes are implemented as a Celery cluster with two worker nodes in the proof-of-concept deployment. Additionally, an SQL database used for storing operational data, and a shared filesystem containing the workflow codes, plugins, and logs have to be available from all of the core nodes.

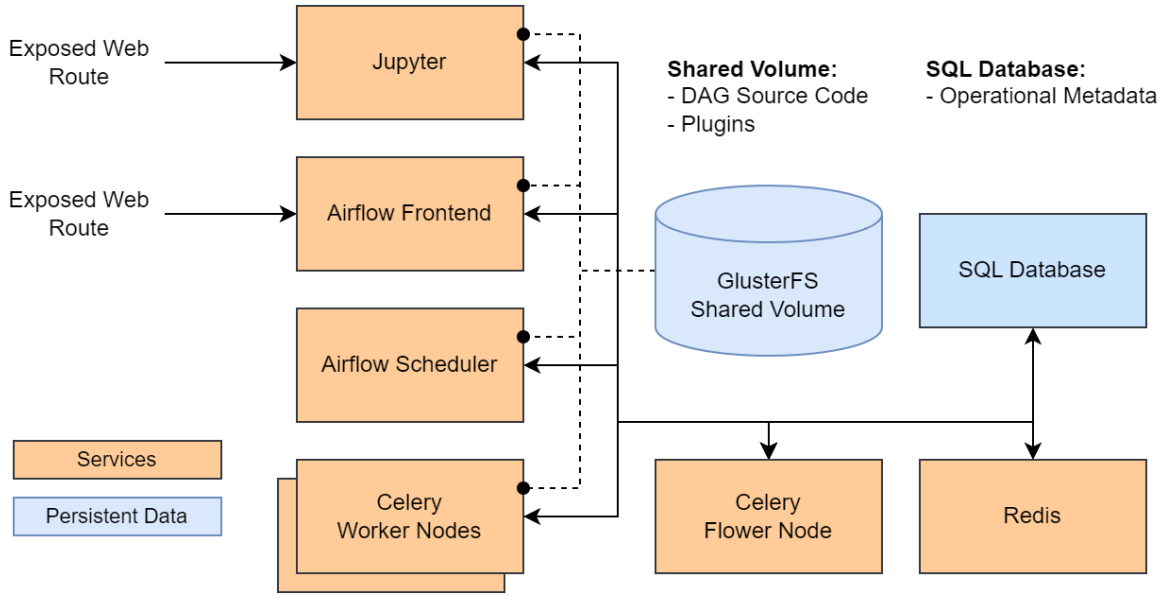

Figure 4.3: Core Apache Airflow Services Deployed on CSC Rahti

Airflow is centered around the directed acyclic graph (DAG) structure for it's workflows and task planning. Each Airflow DAG can be comprised of one or more tasks, with a defined order of execution. DAGs and Tasks can be defined through Python code using Airflow's classes or through a newer simpler method called TaskFlow, which allows using decorators to turn existing functions into Tasks. Airflow is also designed to be highly extensible, with a wide range of integrations with other tools through or re-usable APIs

(Hooks) that are typically used to interact with external services (e.g., SQL connection), and predefined input-driven tasks (Operators) which can use available Hooks (e.g., executing an SQL query). There is strong user support for developing custom Hooks and Operators, allowing for simple creation of support tools to aid fusion workloads. These concepts are relevant for the later platform modifications (Subsection 4.5.4)

CSC Allas

The final component of the proof-of-concept platform is a managed object storage service provided by CSC, Allas. CSC Allas is based on CEPH [60], an open-source and highly scalable[61] object storage system which enables the storage of arbitrary binary data objects. Relying on an object store for bulk data storage allows for great flexibility in the stored data formats, however it limits the ability to query data based on structured features. Modern solutions such as the Lakehouse[62] architecture can be deployed in hand with object stores to enable structured data and querying features if desired. In the case of the proof-of-concept implementation, Allas is used to store clean ML-ready data and to provide access to researchers aiming to use this data.

4.5.2 Deployment Process

The infrastructure layer in this implementation is based on available services from CSC outlined in Section 4.2. An OKD Kubernetes cluster "CSC Rahti" is the main backbone on which this proof-of-concept platform is deployed. The OpenShift instance was used to deploy the InvenioRDM and Apache Airlow applications along with their supporting services. Additionally, the bulk storage layer for processed datasets is an S3-compatible data lake service "Allas".

Following the initial architecture planning and tool selection, work began on deploying the desired tools onto CSC Rahti / OKD cluster. The deployment for Airflow and InvenioRDM were handled as independent projects, with public routes used for connections between the two systems.

For Apache Airflow, a deployment template [63] was available from the CSC Rahti service. This template was used in combination with a custom Docker image containing any extensions and custom tooling developed for this case study. The template allows for a simple user configuration to be converted into a deployment of the specified Airflow image for the frontend, scheduler, and worker containers, along with a supporting SQL database, and a Redis + Celery task queue. Additional configuration is available for configuring credentials, environment variables, and other minor properties.

InvenioRDM was more challenging to deploy, as there was no readily available deployment method compatible with CSC Rahti's OKD instance. What was hoped to be a

simple solution, a readily available Kubernetes/OpenShift Helm chart, did not work with the available cluster. The following changes to the available Helm chart were necessary:

- The differences between schema strictness between the available OKD Kubernetes version on CSC Rahti, and the Kubernetes and OpenShift versions for which the Helm chart was designed made the Helm chart incompatible for deployment on CSC Rahti. With minor changes to the schema, the Helm chart could be successfully used to deploy the bulk of the services.
- By default, the Helm chart deploys an OpenSearch instance to handle full search capabilities of the InvenioRDM platform. However, the available OpenSearch container image is not designed to work with non-privileged containers available on CSC Rahti. Due to this issue, OpenSearch was replaced by ElasticSearch, which is also natively supported by InvenioRDM v11.0

After these changes there was one remaining missing feature for automated initial setup, so those commands had to be executed manually through the CLI of a worker node. The InvenioRDM Helm chart also supports the use of a custom Docker image for the core services, which was utilized to enable platform customization. After these modifications, InvenioRDM was successfully deployed along with all of it's supporting services.

Both the custom Airflow and InvenioRDM Docker images are generated directly from the project's git repository through a CI/CD pipeline triggered on each new code commit, to enable faster iteration when developing custom tooling and platform extensions.

4.5.3 Deployment Architecture

The architecture for the case study deployment closely follows the metadata-enabled architecture outlined in Section 3.4 and developed for this proof-of-concept implementation in Section 4.3. In the final deployment Apache Airflow is used as the workflow orchestration platform, and InvenioRDM is used as the Research Data Platform. A clear mapping can be made from the metadata-enabled architecture to the selected tools (Figure 4.4).

At this stage, most of the MVP criteria are satisfied, with the exception of ease-of-use, and ad-hoc provenance metadata collection. InvenioRDM provides unique and references for the records. Provenance metadata can be attached in any desired file format to the InvenioRDM records, however the lack of standardization in this aspect does not provide a significant improvement over the current approach. Airflow introduces a common format for data workflows and the ability to handle data re-calibrations both via API triggers or through a scheduled checkup routine. Finally, both InvenioRDM and Airflow satisfy the need for deployment independent from public cloud vendors, and are

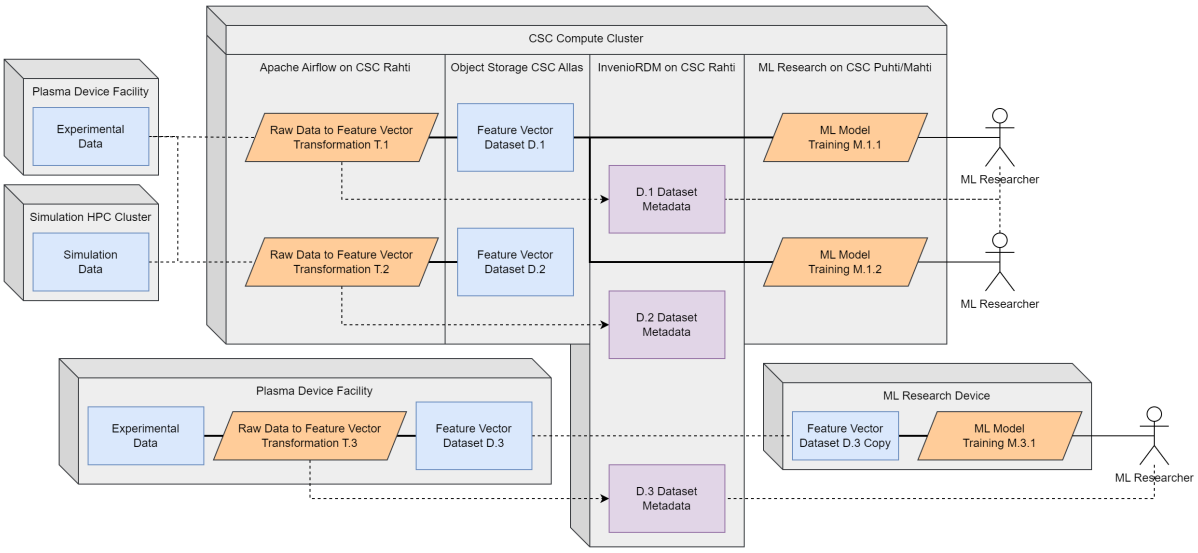

Figure 4.4: From the component architecture shown in Section 4.3, the workflow component is now represented by Apache Airflow, and the metadata catalog is represented by InvenioRDM, both hosted on CSC Rahti. Additionally, the CSC Allas object store is used as the bulk storage component. The resulting data can be efficiently accessed from an HPC environment using CSC Puhti/Mahti.

under active development for the foreseeable future. To address the need for improved ease-of-use, and to establish more standardized provenance and metadata propagation, additional tooling was developed to support the fusion data workflows.

4.5.4 Extending Selected Tools for Fusion Workloads

Both InvenioRDM and Apache Airflow are highly customizable platforms, which is one of the key reasons for their selection. The focus on enabling ease of use for end users has led to the development of custom Apache Airflow Hooks to enable simple integration with fusion data ingestion, the automation of metadata generation, and a one-line publishing function for submitting results to InvenioRDM.

The goal of these extensions is to greatly reduce the effort required by the researcher to utilize the platform in their existing workflows, and to establish a standardized method for provenance metadata to be propagated to the InvenioRDM platform. The provided tools enable simple Python functions to be imported into the existing code for the following tasks:

- Ingesting source data - Adding a custom function for source data access in the workflow allows for automating the metadata collection process.
- Outputting resulting data - Since the platform introduces it's own bulk data layer, a simplified abstraction can be provided to the researchers for storing results. In

this case the abstraction also provides a way to organize the resulting data based on the unique InvenioRDM record version.

- Publishing a metadata record - To avoid the need for manual submission or using generic Python API interactions, a custom functions allows metadata records to be automatically published based on metadata collected from the data ingestion hook, along with additional generic information (Dataset Author, Title, and Description) provided by the researcher.

In-situ Metadata Generation

One of the core features of the trialed system is the generation of metadata. Providing high-quality metadata both for data querying, filtering, and analysis, as well as provenance, is a repeatedly identified challenge [11, 18, 10]. The different push and pull approaches to metadata collection have already been discussed in the Fair4Fusion report [10], noting that push-based metadata ingestion is likely the better choice in situations where it is possible as it allows the data producers to have greater control over how data and metadata is submitted into the system. This is especially relevant in the administratively complex landscape of fusion research facilities.

For this purpose, custom tooling has been developed to extend the capabilities of Apache Airflow. Namely, an Airflow Hook has been defined which collects metadata throughout the data transformation workflow. The tool is structured as a middleware between the data transformation code and the source data access layer. By including custom metadata-collecting code in this segment, researchers automatically have reliable metadata recorded during the source data fetching stage. This collected metadata can be used to annotate the provenance of the output result and to provide additional context to the output dataset.

Dataset Publishing

To facilitate the easier output of clean ML-ready data, another middleware-like Airflow Hook was developed around the S3 protocol used to upload data to CSC Allas, the object storage platform used for bulk data storage in this proof-of-concept platform. The Hook serves two purposes, primarily it enables a simplified data output process for the researcher, and secondly, it allows data to be stored according to the InvenioRDM record ID, which is automatically incorporated into the data path. This approach simplifies the data organization within the object store by mapping dedicated paths to each InvenioRDM record (and version).

By default InvenioRDM records are designed to be versioned, with individual versions being immutable once recorded. To enable a more streamlined metadata publishing

process, a custom integration was developed once again as an Airflow Hook. This hook allows researchers to easily upload metadata for a new version of the dataset, which includes metadata collected from the previously outlined metadata generation hook used for source data access in combination with a title, description, and author details provided by the researcher. By using this hook, researchers can rely on a simple Python function to handle the metadata submission process, as opposed to manually interacting with the InvenioRDM API.

4.6 Proof-of-Concept Data Platform Evaluation

4.6.1 End-to-end Evaluation

The user experience of the platform was evaluated through a real-world fusion research example [15]. In this example raw data from the JET tokamak was processed into an ML-ready format. Originally, this process was performed in isolation from any existing fusion research infrastructure, and the middle-stage artifacts such as the data transformation and cleaned dataset remained unshared. By adopting the proof-of-concept platform into the workflow, the hope is to demonstrate how artifacts which already existed within the workflow, namely the data processing script and resulting clean data, can be easily standardized, made re-usable, published, and discovered by other researchers.

Data Production Flow

The following steps are needed to convert an existing ML data transformation script into a dataset registered in the proof-of-concept data platform:

1. The researcher creates a draft record using the InvenioRDM web platform. The draft provides the researcher with a unique ID for referencing the root record from the data generating workflow code.

2. The existing Python data transformation code is converted to an Airflow DAG via the following modifications:

    - The existing data loading mechanism is replaced with the provided data ingestion hook which allows the user to retrieve desired diagnostics from a list of specific JET pulses as a collection.
    - The data transformation code is wrapped with functions representing the Airflow DAG, and a single Airflow Task using the TaskFlow decorators
    - The existing data output mechanism is replaced with the provided data output hook

- An additional call is added to the metadata publishing function with specified metadata for the dataset title, description, and author information

3. The root record ID is passed to the Airflow Hook used for publishing metadata. Once specified, this ID is used to populate the InvenioRDM record, and to publish additional versions if the workflow is re-run.

4. The code is uploaded to the platform using the Jupyter Web GUI.

5. The workflow is executed using the Airflow GUI or API (Figure 4.5).

6. The workflow automatically outputs the final data into a unique path in CSC Allas, and publishes a respective record to InvenioRDM (Figure 4.6) embedded with provenance metadata and data access instructions. The published dataset can immediately be discovered and used by other researchers with sufficient access.

Data Discovery Flow

The following steps make up the process of discovering existing datasets and using them for new ML research efforts:

1. Researchers use the browsing and search GUI of InvenioRDM to locate existing datasets.

2. The chosen dataset is inspected in the GUI where instructions are provided on how to access the bulk data.

3. The complete dataset is loaded directly from CSC Allas into the ML research environment.

4. When publishing results, the record of the used dataset version, which includes provenance information describing exact details of contained source data, can be uniquely referenced.

The presented workflow has been successfully demonstrated by onboarding the provided ML workflow example [15].

Figure 4.5 displays how Apache Airflow and InvenioRDM interact on a higher level and includes screenshots of the live deployed instances. A final preview of the resulting record is show in Figure 4.6, taken from the live proof-of-concept deployment.

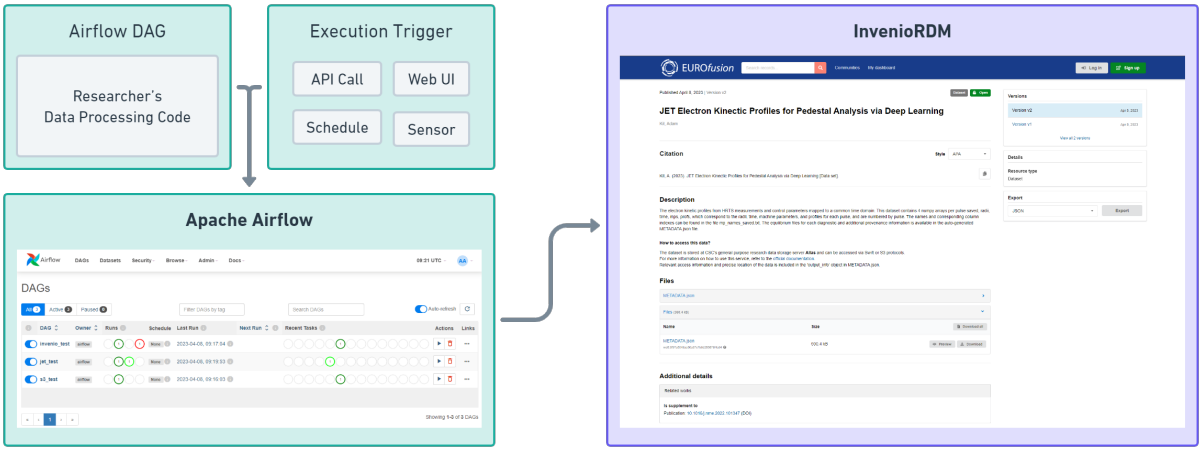

Figure 4.5: Application Diagram

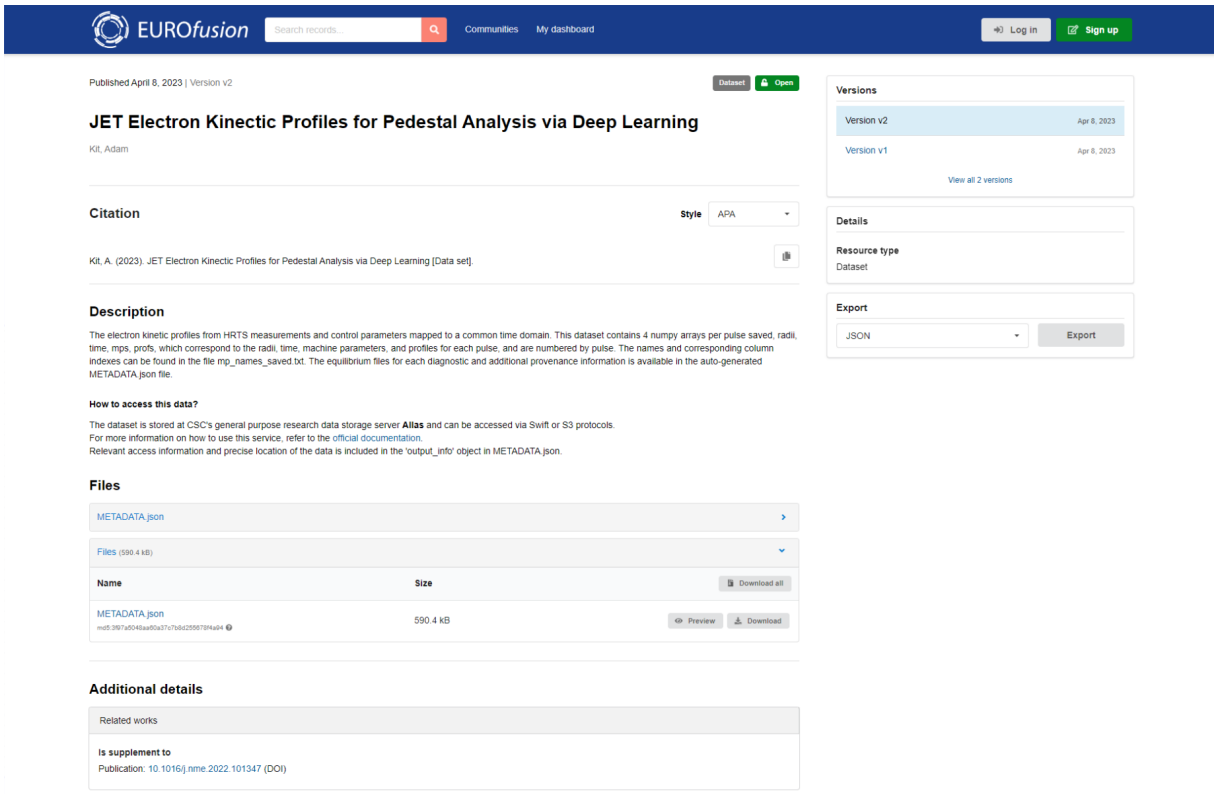

Figure 4.6: InvenioRDM Record Screenshot

4.6.2 Challenges

The following challenges have been encountered during the development, deployment, and end-to-end evaluation of the proof-of-concept platform.

The design of the case study platform has evolved over the course of this project due to issues with selected tools, or the discovery of more suitable solutions later in the process. This was a result of the limited project resources, time frame, and the volume of available tools. Initially, trials were performed with DataHub, and then with Marquez, as the metadata platform. Neither of these application was a great fit for this project due to the lack of a public-facing discovery portal, as well as limited support for open data standards. Finally, InvenioRDM was selected and integrated as the core metadata component, satisfying the core metadata cataloging needs.

Another encountered challenge with the chosen tools has been the difficulty to perform local testing of Airflow DAGs without introducing significant additional work by the researcher. Airflow does not offer a comprehensive remote development solution, nor an easily testable workflow code. This presented a challenge to the development pace, as it required constant manual updates to the Airflow DAGs, and led to a trial-and-error approach to debugging. Suggestions for addressing this challenge in future efforts are outlined in Section 5.1.

4.6.3 Resource Estimation and Scalability

The proof-of-concept platform was deployed on top of OKD Kubernetes, and the consumed resources are spread across the various Kubernetes pods (smallest Kubernetes units with defined compute and memory resources). In the case of this case study the provisioned resources are based on the provided Apache Airflow template provided by CSC, and the Helm chart provided by InvenioRDM developers. There has not been significant effort dedicated to optimizing the resource use, so the following estimates (Table 4.6 and 4.7) are considered conservative.

The case study solution minimizes resource usage by abstracting away from the bulk computation and storage layer. The core of the platform consists of services which can run on very limited resources. In a production setting however, many of these services such as the SQL databases, cache, or worker nodes would be scaled up, according to the operational demands. Despite the scaling changes to a production setting, the majority of these services remain light on resource usage compared to the bulk data processing and storage elements.

The main data processing cluster in the proof-of-concept deployment is made up from Celery worker nodes (as seen in Table 4.7, and previously introduced in Figure 4.3). The number and resources of these nodes can be scaled up for processing production grade

Table 4.6: InvenioRDM Deployment Resource Utilization

Service	CPU (in 2.4 GHz Cores)	RAM (in GB)
nginx* (InvenioRDM Frontend + Proxy)	0.25	0.5
InvenioRDM API*	1	1
HAProxy	0.25	0.5
PostgreSQL	2	8
Redis	2	8
ElasticSearch	2	1
RabbitMQ	1	2
Worker*	1	1
Worker Beat	2	0.5
Total	11.5	22.5
Persistent Storage	10 GB	

*These services are horizontally scalable, in the proof-of-concept deployment there are two instances of each.

Table 4.7: Apache Airflow Deployment Resource Utilization

Service	CPU (in 2.4 GHz Cores)	RAM (in GB)
Web UI	1	2
Scheduler	1	2
Worker*	2	8
Celery Flower	2	8
Redis	2	1
PostgreSQL	2	1
Jupyter	2	8
Total	12	30
Persistent Storage	14 GB	

*These services are horizontally scalable, in the proof-of-concept deployment there are two instances of each.

data transformation workflows. Since the Celery worker nodes are provisioned from the CSC Rahti Kubernetes layer, the proof-of-concept system in this deployment is relying on cloud, rather than HPC resources, to execute the data processing workloads. As the resulting datasets are persisted, these workflows do not need to run often, which lowers the need for fast execution. Despite the low impact of data workflow execution speed in the context of persisted datasets, it is recommended to consider alternative deployment approaches which better utilize HPC resources to execute the data processing workloads (discussed in more details later in Section 5.4).

In addition to the Apache Airflow and InvenioRDM instances, the CSC Allas object store was used for persisting the resulting datasets. For persisting the trial dataset, which primarily consists of low dimensional data, only 0.5GB of CSC Allas storage was used. Despite very limited resources used for the trial evaluation, the scalable bulk storage layer CSC Allas, and scalable Celery worker cluster utilized by Apache Airflow allow the system to easily scale up to meet the growing demands of fusion data workloads.

5. Discussion

The following sections compile and interpret the findings from the survey of fusion ML research and the case study ML data platform implementation. The chapter is divided into three sections, the first section discusses the choices and insights from the case study, the second section discusses possible actions that can be taken to improve ML workflows in fusion today, and finally the last section discusses the opportunities for future work on the topic of ML infrastructure in fusion.

5.1 Reflection on Case-Study Decisions

The challenges encountered with the selected tools, outlined in Section 4.6.2, suggest further evaluation of available tools. Several alternatives can be suggested for further evaluation.

To address the challenge of the development experience, and the ability to quickly test and iterate on workflow design, there is a need to expand on the proof-of-concept implementation. It is possible to deploy a local instance of Airflow for running DAGs locally. There is also a supported method to synchronize DAGs from a git[64] repository. Both of these Airflow-specific approaches could assist in addressing this issue in the future. Beyond Airflow, one of the other evaluated workflow orchestrators, Flyte, promises better code portability, which can in theory enable easier re-use of the same code across different infrastructure ranging from local Python environments for testing to big data-scale cloud resources in production. Evaluating the development experience and code testing capabilities should be included as a significant consideration for similar future studies.

Another option is to rely on the Common Workflow Language (CWL)[53], an academic open-source workflow standard. CWL establishes a vendor-agnostic specification for defining computational workflows. CWL workflows can be executed by several open-source implementations, the most comprehensive being StreamFlow[52], which enables execution of CWL workflows across both cloud and HPC resources. Alongside the LLNL stack of workflow tooling [25], they present promising solutions in the very limited tooling for HPC-native workflow management.

Finally, a feature which was not explored in the context of the proof-of-concept

implementation is the management of workflow source code. For this purpose the recommendation is to rely on git[64]-based code repositories such as GitLab or GitHub, as they provide the necessary versioning and referenceability of unique code versions.

5.2 Applicability Limitations of the Case-Study Results

The content of this thesis does not address the administrative challenges of fusion research. Data access policy, privacy, and security pose significant factors which can change the constraints of possible data architectures. Currently, various fusion research facilities require authentication and authorization of researchers in order to provide access to their data. The additional regulatory constraints posed by the risk associated with nuclear research limit the openness of the fusion data and research ecosystem to the general public.

Addressing these limitations in detail is beyond the scope of this thesis. Thus, with the assumption that sufficient data sharing regulations can be achieved, a number of applicable suggestions and proposed topics for future research are derived from this case-study in the subsequent sections.

5.3 Derived Suggestions for Fusion ML Infrastructure

A central research dataset repository is a common solution in different academic settings and could be integrated into a fusion research community. Allowing duplicate data storage for ML-ready data (e.g. recommended datasets) should be considered, as it provides a simple solution to avoid the previously lacking data access layer in ML applications. A baseline one-size-fits-all solution can be achieved by using a data lake along with a comprehensive data catalog. The flexibility of relying primarily on metadata for dataset records, agnostic of the bulk storage layer, allows future datasets to be stored in more optimized storage systems, once such storage systems are developed. Before more distributed approaches such as the dataset-as-query architecture are developed, a dataset-as-workflow approach, which standardizes the data-generating code as a re-usable workflow, could provide a significant improvement over static one-off datasets. Research should be done on which code standardization approaches are best suited for existing fusion research infrastructure.

Developing and deploying a full scale ML data platform for fusion research requires

a significant investment into manpower, especially experts in platform engineering, HPC systems, and fusion ML approaches. Only through a tight collaboration between infrastructure experts and end users can it be possible to develop a system which is practical, robust, and resource efficient.

In addition to developing a data platform, it is necessary to address the challenges of adoption. The initially empty data platform requires effort to populate with existing datasets, and support for onboarding future ML research projects to the platform. For this purpose, it would be highly beneficial to appoint an onboarding officer for the purpose of migrating existing and future datasets onto the platform until a critical mass is reached. A similar approach as been shown effective in stimulating the adoption of Duqtools[65], a tool for dynamic uncertainty quantification in fusion simulations.

Furthermore, this research should not be done in isolation from other topically adjacent efforts within the fusion community. Projects such as Fair4Fusion[10], which focuses on wider-scope FAIR data and encompasses metadata cataloging, could easily result in a platform which can be suitable as the ML data cataloging component of the considered ML data platform architecture, which was implemented through InvenioRDM in this study. Other projects focusing on ML data processing such as the disruption database tooling[66] should also be considered as possible integrated components of the platform.

In addition to these suggestions, there remain several open questions for the development of a data platform based on this centralized metadata-enabled approach. These questions are presented as future research opportunities in the following section.

5.4 Future Work

Additional work is required to explore the cost and benefit of persisting ML-ready datasets in a central repository (Subsection 3.3.2) versus developing complex ML data query systems (Subsection 3.3.1), in the context of available infrastructure for fusion research.

In the proof-of-concept platform, the data transformation workflows were defined using Apache Airflow and executed via a Celery cluster. This approach relied on Kubernetes-native cloud resources. More commonly, fusion research relies on HPC resources. Therefore, further work is also necessary to identify satisfactory HPC-native counterparts for workflow processing. While Apache Airflow, Flyte, and many other mature workflow orchestration tools exist for running workflows on top of cloud resources, there is a lack of support for HPC execution in mainstream solutions. Several HPC-compatible possibilities have been suggested in previous sections such as CWL[53] via StreamFlow, and LLNL's own workflow tooling [25]. By offloading the bulk computation to HPC resources, the cloud resources necessary for maintaining a scalable data platform would be significantly

lowered.

In terms of platform architecture, the explore approach focuses on a centralized data platform, which enables collecting data across the heterogeneous sources, and processing it into ML-ready workflows which are persisted in a new location for easy access by ML researchers. This approach has scalability limitations associated with the volume of datasets. While low-dimensional data can produce manageable scale datasets, projected video data volume such as the 2.2PB/day estimate for ITER[13] may require pass-through access directly to source storage facilities. Thus, consideration should be given in future research on how to best to accommodate heterogeneous datasets which combine processed ML-ready low-dimensional data and more heavy source data such as video into a single dataset.

In addition to providing infrastructure for ML workloads, there is a need for standardization of ML tooling within the fusion community due to the extent of the available options. Deciding on a recommended set of ML tools can provide ground for easier standardization and interoperability of research codes and results. Having a standardized set of tools also simplifies the decision making process for later infrastructure expansion. This suggestion is also based on existing efforts from other domains such as the Pangeo[67] ecosystem which introduces a recommended set of tools for data-driven research in geosciences.

6. Conclusion

In this thesis, it was established that categories of data tooling such as metadata catalogs and workflow orchestrators can potentially be used to address the challenges of standardizing the source to ML data transformations, and the publishing of resulting datasets. These improvements can be used to break up the monolithic ML research process, and reduce the duplicate work currently performed by researchers. By adopting these tools into the ML research workflow, a common base of ML-ready datasets can also be established, providing grounds for improved provenance, reproducibility, and model benchmarking.

Challenges in data-driven research stem from the highly heterogeneous and geographically scattered source data landscape and the monolithic ML research process. Experimental source data is organized in complex tree-like structures which are useful for recording detailed information about individual pulses, however these structures pose a challenge to bulk data processing and require further transformation to clean ML-ready data. Curating data for ML research is currently estimated to take up to 70% of the research effort [11]. In addition to the difficult data curation, ML processes are often performed as a monolithic effort, leading to additional challenges such as duplicate work, lack of collaboration, and often unshared data transformations and cleaned data artifacts which are present in every ML research workflow.

To address these challenges, several possible architectures for the organization of a fusion ML data platform have been explored. The research process was separated into a framework of four components representing the source data, source -> ML data transformation, ML feature data, and the ML training process. These four high-level components are present in every fusion ML workflow, and their distribution across real-world hardware outlines the possible high-level architectures for fusion ML data. Two different approaches can be identified: a query-based approach where ML data can be queried on-demand, with data cleaning and preprocessing performed at the source location, and a persisted dataset approach where data is collected and cleaned through a centralized workflow, with the resulting ML feature data stored as a discrete readily-available dataset. The query-based approach offers higher flexibility for what data can be retrieved, however it requires adding considerable improvements in querying functionality as well as source data access speeds to the current storage systems. On the other hand, the persisted dataset approach, evalu-

ated in the case study, is limited to select curated datasets and requires additional storage resources for storing the clean datasets. However, it does not require significant changes to existing infrastructure, and it enables the creation of reference datasets. Finally, through a comprehensive metadata catalog, data from both of these approaches could be indexed and made discoverable and referenceable.

The metadata-enabled architecture (Section 3.4) was the basis of a proof-of-concept case-study data platform. Two principal components of this architecture are identified: a metadata catalog to index and provide a dashboard for discovering ML-ready datasets, and a workflow orchestration platform to manage the data collection workflows. The case-study proof-of-concept data platform based on these two components was implemented using available cloud resources and readily-available open-source tools. The proof-of-concept platform consisting of an InvenioRDM instance as the metadata catalog, and an Apache Airflow instance as the workflow orchestrator, was deployed using an OKD Kubernetes cluster (CSC Rahti), while a CEPH-based object store (CSC Allas) was used as the data lake for storing processed datasets. Additionally, extending the chosen tools with deeper integrations and automation of standard tasks, such as metadata collection, significantly improved ease-of-use.

The evaluation of the proof-of-concept platform was based on a real fusion ML research example [15], which was onboarded onto the platform. The end-to-end process allowed for the identification of practical challenges such as enabling a practical development environment, workflow testing, and the need for simplified layer for user interactions with the infrastructure through custom tooling. Overall, the platform demonstrates that it is possible to transform elements of existing ML workloads into reusable data-generating workflows, and that the clean ML data from existing research can be published and cataloged with ease when researchers are supplied with the relevant infrastructure and tooling.

For future efforts on developing a fusion data platform, open questions such as the feasibility of dedicated ML-data storage systems, and the pursuit for HPC-compatible workflow orchestrators, remain to be addressed. Additional focus should also be directed towards ensuring a user-friendly platform interface, such that domain experts do not need to spend considerable efforts on adopting the platform. Finally, a general recommendation can be made for introducing of a metadata catalog capable of indexing datasets across a heterogeneous set of sources, while providing a central interface for ML data publishing and discovery.

Bibliography

[1] *ITER Technical Basis*. Number 24 in ITER EDA Documentation Series. INTERNATIONAL ATOMIC ENERGY AGENCY, Vienna, 2002. URL https://www.iaea.org/publications/6492/iter-technical-basis.

[2] Aaron Ho et al. Predictive JET current ramp-up modelling using QuaLiKiz-neural-network. *Nuclear Fusion*, 63(6):066014, 4 2023. ISSN 0029-5515. doi: 10.1088/1741-4326/ACC083. URL https://iopscience.iop.org/article/10.1088/1741-4326/acc083.

[3] S. Ku et al. Full-f gyrokinetic particle simulation of centrally heated global ITG turbulence from magnetic axis to edge pedestal top in a realistic tokamak geometry. *Nuclear Fusion*, 49(11):115021, 9 2009. ISSN 0029-5515. doi: 10.1088/0029-5515/49/11/115021. URL https://iopscience.iop.org/article/10.1088/0029-5515/49/11/115021.

[4] R. Pasqualotto et al. High resolution Thomson scattering for Joint European Torus (JET). *Review of Scientific Instruments*, 75(10):3891–3893, 10 2004. ISSN 0034-6748. doi: 10.1063/1.1787922.

[5] L. Cupido et al. High resolution fast wave reflectometry: JET design and implications for ITER. *Review of Scientific Instruments*, 79(10), 10 2008. ISSN 00346748. doi: 10.1063/1.2953576/1071038. URL /aip/rsi/article/79/10/10F106/1071038/High-resolution-fast-wave-reflectometry-JET-design.

[6] F. Imbeaux et al. Design and first applications of the ITER integrated modelling & analysis suite. *Nuclear Fusion*, 55(12):123006, 10 2015. ISSN 0029-5515. doi: 10.1088/0029-5515/55/12/123006. URL https://iopscience.iop.org/article/10.1088/0029-5515/55/12/123006.

[7] L Fleury et al. WEST plasma reconstruction chain and IMAS related tools, 2020. URL https://hal.science/hal-03117771/document.

[8] C. Bourdelle et al. Separatrix parameters and core performances across the WEST L-mode database. *Nuclear Fusion*, 63(5):056021, 5 2023. ISSN 0029-5515. doi: 10.1088/1741-4326/acbfcf.

[9] R. Layne et al. The JET Intershot Analysis: Current infrastructure and future plans. *Fusion Engineering and Design*, 85(3-4):403–409, 7 2010. ISSN 0920-3796. doi: 10.1016/J.FUSENGDES.2009.12.012.

[10] Marcin Plociennik et al. Blueprint architecture for a Fusion Open Data Framework, 5 2022. URL https://doi.org/10.5281/zenodo.6759119.

[11] David Humphreys et al. Advancing Fusion with Machine Learning Research Needs Workshop Report. *Journal of Fusion Energy*, 39(4):123–155, 8 2020. ISSN 0164-0313. doi: 10.1007/s10894-020-00258-1. URL https://link.springer.com/article/10.1007/s10894-020-00258-1.

[12] Jonathan Hollocombe et al. Cataloguing of ITER simulations using SimDB, 2021. URL https://conferences.iaea.org/event/251/contributions/20655/.

[13] S. D. Pinches. ITER's Strategy for Diagnostic Data Analysis, 5 2019. URL https://nucleus.iaea.org/sites/fusionportal/Pages/DPWS-6/TM%20Fusion%20Data%20Processing%20Validation%20and%20Analysis/2_Tuesday/Session%20II%20Data%20Analysis%20Lessons%20Learnt,%20Best%20Practices%20and%20Proposals%20for%20ITER/1120%20Pinches%20S.pdf.

[14] Sachchidanand Singh and Nirmala Singh. Big Data analytics. *Proceedings - 2012 International Conference on Communication, Information and Computing Technology, ICCICT 2012*, 2012. doi: 10.1109/ICCICT.2012.6398180.

[15] A. Kit et al. Developing deep learning algorithms for inferring upstream separatrix density at JET. *Nuclear Materials and Energy*, 34:101347, 3 2023. ISSN 2352-1791. doi: 10.1016/J.NME.2022.101347.

[16] Jonas Degrave et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature 2022 602:7897*, 602(7897):414–419, 2 2022. ISSN 1476-4687. doi: 10.1038/s41586-021-04301-9. URL https://www.nature.com/articles/s41586-021-04301-9.

[17] Karel Lucas van de Plassche et al. Fast modeling of turbulent transport in fusion plasmas using neural networks. *Physics of Plasmas*, 27(2), 11 2019. doi: 10.1063/1.5134126. URL http://dx.doi.org/10.1063/1.5134126.

[18] Rushil Anirudh et al. 2022 Review of Data-Driven Plasma Science. 5 2022. URL http://arxiv.org/abs/2205.15832.

[19] N.W. Eidietis et al. The ITPA disruption database. *Nuclear Fusion*, 55(6):063030, 6 2015. ISSN 0029-5515. doi: 10.1088/0029-5515/55/6/063030. URL https://iopscience.iop.org/article/10.1088/0029-5515/55/6/063030.

[20] E. Aymerich et al. Disruption prediction at jet through deep convolutional neural networks using spatiotemporal information from plasma profiles. *Nuclear Fusion*, 62: 066005, 6 2022. ISSN 0029-5515. doi: 10.1088/1741-4326/ac525e.

[21] G. Sias et al. Disruption prediction approaches using machine learning tools in tokamaks. pages 2880–2890. IEEE, 6 2019. ISBN 978-1-7281-3403-1. doi: 10.1109/PIERS-Spring46901.2019.9017280.

[22] Joost Croonen et al. Investigation of machine learning techniques for disruption prediction using jet data. *Plasma*, 6:89–102, 2 2023. ISSN 2571-6182. doi: 10.3390/plasma6010008.

[23] Julian Kates-Harbeck et al. Predicting disruptive instabilities in controlled fusion plasmas through deep learning. *Nature*, 568:526–531, 4 2019. ISSN 0028-0836. doi: 10.1038/s41586-019-1116-4.

[24] Chenguang Wan et al. A Robust and Fast Data Management System for Machine-Learning Research of Tokamaks. *IEEE Transactions on Plasma Science*, 50(12): 4980–4986, 12 2022. ISSN 0093-3813. doi: 10.1109/TPS.2022.3223732. URL https://ieeexplore.ieee.org/document/9972905/.

[25] Dong H. Ahn et al. Flux: Overcoming scheduling challenges for exascale workflows. *Future Generation Computer Systems*, 110:202–213, 9 2020. ISSN 0167739X. doi: 10.1016/J.FUTURE.2020.04.006.

[26] Tibor Šimko et al. REANA: A System for Reusable Research Data Analyses. *EPJ Web of Conferences*, 214:06034, 9 2019. ISSN 2100-014X. doi: 10.1051/epjconf/201921406034.

[27] Luis Bastiao Silva et al. MONTRA: An agile architecture for data publishing and discovery. *Computer Methods and Programs in Biomedicine*, 160:33–42, 7 2018. ISSN 01692607. doi: 10.1016/j.cmpb.2018.03.024. URL https://linkinghub.elsevier.com/retrieve/pii/S0169260717312725.

[28] Open Research Data and Data Management Plans Information for ERC grantees by the ERC Scientific Council. 2022.

[29] European Organization For Nuclear Research and OpenAIRE. Zenodo, 2013. URL https://www.zenodo.org/.

[30] Computing - Docs CSC, 2023. URL https://docs.csc.fi/computing/.

[31] Rahti - Docs CSC, 2023. URL https://docs.csc.fi/cloud/rahti/.

[32] Allas object storage - Docs CSC, 2023. URL https://docs.csc.fi/data/Allas/.

[33] Charles R. Harris et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 9 2020. ISSN 14764687. doi: 10.1038/S41586-020-2649-2.

[34] J. Mailloux et al. Overview of JET results for optimising ITER operation. *Nuclear Fusion*, 62(4):042026, 4 2022. ISSN 0029-5515. doi: 10.1088/1741-4326/ac47b4.

[35] Michael Boch et al. A Systematic Review of Data Management Platforms. pages 15–24. 2022. doi: 10.1007/978-3-031-04819-7{_}2. URL https://link.springer.com/10.1007/978-3-031-04819-7_2.

[36] Ricardo Carvalho Amorim et al. A comparison of research data management platforms: architecture, flexible metadata and interoperability. *Universal Access in the Information Society*, 16(4):851–862, 11 2017. ISSN 16155297. doi: 10.1007/S10209-016-0475-Y.

[37] S. Weibel et al. Dublin Core Metadata for Resource Discovery. Technical report, 9 1998. URL https://www.rfc-editor.org/info/rfc2413.

[38] CKAN - The open source data management system, 2023. URL https://ckan.org/.

[39] InvenioRDM, 2023. URL https://inveniosoftware.org/products/rdm/.

[40] DSpace, 2023. URL https://dspace.lyrasis.org/.

[41] The Dataverse Project - Dataverse.org, 2023. URL https://dataverse.org/.

[42] OpenLineage Docs, 2023. URL https://openlineage.io/.

[43] Marquez, 2023. URL https://marquezproject.ai/.

[44] A Metadata Platform for the Modern Data Stack | DataHub, 2023. URL https://datahubproject.io/.

[45] LF AI & Data Landscape, 2023. URL https://landscape.lfai.foundation/.

[46] CaltechDATA, 2023. URL https://data.caltech.edu/.

[47] TU Wien Research Data, 2023. URL https://researchdata.tuwien.ac.at/.

[48] Jorge Ejarque et al. Enabling dynamic and intelligent workflows for HPC, data analytics, and AI convergence. *Future Generation Computer Systems*, 134:414–429, 9 2022. ISSN 0167-739X. doi: 10.1016/J.FUTURE.2022.04.014.

[49] Mihhail Matskin et al. A Survey of Big Data Pipeline Orchestration Tools from the Perspective of the DataCloud Project. 5 2021.

[50] Apache Airflow, 2023. URL https://airflow.apache.org/.

[51] Build production-grade data and ML workflows, hassle-free with Flyte. URL https://flyte.org/.

[52] Iacopo Colonnelli et al. StreamFlow: Cross-Breeding Cloud with HPC. *IEEE Transactions on Emerging Topics in Computing*, 9(4):1723–1737, 2021. ISSN 21686750. doi: 10.1109/TETC.2020.3019202.

[53] P Amstutz et al. Common Workflow Language. page 1, 2016. doi: 10.6084/m9.figshare.3115156.v2. URL https://w3id.org/cwl/.

[54] Will Reese. Nginx. *Linux Journal*, 9 2008. doi: 10.5555/1412202.1412204. URL https://dl.acm.org/doi/10.5555/1412202.1412204.

[55] HAProxy - The Reliable, High Perf. TCP/HTTP Load Balancer, 2023. URL https://www.haproxy.org/.

[56] Redis, 2023. URL https://redis.io/.

[57] RabbitMQ - Messaging that just works, 2023. URL https://www.rabbitmq.com/.

[58] Celery - Distributed Task Queue, 2023. URL https://docs.celeryq.dev/en/stable/.

[59] Helm - The package manager for Kubernetes, 2023. URL https://helm.sh/.

[60] Sage A. Weil et al. Ceph: a scalable, high-performance distributed file system. 2006. doi: 10.5555/1298455.1298485. URL https://www.scinapse.io/papers/1999984505.

[61] Diana Gudu et al. Evaluating the performance and scalability of the Ceph distributed storage system. *Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014*, pages 177–182, 1 2015. doi: 10.1109/BIGDATA.2014.7004229.

[62] Michael Armbrust et al. Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics. In *Proceedings of CIDR*, page 8, 2021.

[63] Rahti Templates - Docs CSC, 2023. URL https://docs.csc.fi/cloud/rahti/template-docs/.

[64] Diomidis Spinellis. Git. *IEEE Software*, 29(3):100–101, 5 2012. ISSN 0740-7459. doi: 10.1109/MS.2012.61. URL http://ieeexplore.ieee.org/document/6188603/.

[65] Stef Smeets et al. duqtools. 5 2023. doi: 10.5281/ZENODO.7912818. URL https://zenodo.org/record/7912818.

[66] Alessandro Pau et al. Disruption prediction tool. Manuscrpit in progress., 2023.

[67] Pangeo - A community platform for Big Data geoscience, 2023. URL https://pangeo.io/.