# Data Management and I/O Provisioning Across Cloud-Edge Continuum for High-performance Computational Data Pipelines

Bartosz Kryza[1 [0000−0001−5407−4126]], Renata Słota[1 [0000−0001−7424−9317]], Łukasz Dutka[2 [0000−0002−8781−866X]], Jacek Kitowski[1,2 [0000−0003−3902−8310]], Juan Aznar-Poveda[3 [0000−0002−0879−6651]], Thomas Fahringer[3 [0000−0003−4293−1228]], Ines Ortega-Fernandez[4 [0000−0002−8041−6860]], Iago Abad[4 [0009−0005−6092−8331]], Stefan Dlugolinsky[5 [0000−0002−4424−4221]], Martin Bobak[5 [0000−0002−2905−4999]], Ladislav Hluchy[5 [0000−0001−5450−3808]], Igor Škrlec[6], and Peter Kišša[7 [0009−0006−5499−660X]]

[1] AGH University of Krakow, Faculty of Computer Science, Krakow, Poland
{bkryza, rena, kito}@agh.edu.pl
[2] Academic Computer Centre CYFRONET AGH, Krakow, Poland
lukasz.dutka@cyfronet.pl
[3] University of Innsbruck, Innsbruck, Austria
{juan.aznar-poveda, thomas.fahringer}@uibk.ac.at
[4] Galician Research and Development Center in Advanced Telecommunications (GRADIANT), Carretera do Vilar 56-58, 36214, Vigo, Spain
{iortega, iabad}@gradiant.org
[5] Institute of Informatics Slovak Academy of Sciences, Dúbravská cesta 9, 845 07 Bratislava, Slovak Republic
{stefan.dlugolinsky, martin.bobak, ladislav.hluchy}@savba.sk
[6] MicroStep, spol. s r.o., Vajnorská 158, Bratislava, Slovakia
skrlec@microstep.sk
[7] InterWay a. s., Stará Vajnorská 21, 831 04 Bratislava, Slovak Republic
peter.kissa@interway.sk

**Abstract.** The increasing adoption of cloud, edge and Internet of Things (IoT) technologies has led to the emergence of the cloud–edge compute continuum, enabling low-latency, data-intensive applications across highly distributed infrastructures. However, designing and operating high-performance data pipelines in such environments remains challenging due to heterogeneous resources, distributed data sources, stringent security requirements, and the need for efficient data movement and I/O provisioning. This paper presents an architecture for data management and I/O provisioning that supports the execution of high-performance data pipelines across the cloud–edge continuum. The proposed approach combines federated data management, intelligent data placement, and continuum-aware resource orchestration within a unified platform. These capabilities are integrated with pipeline runtime services that enable adaptive execution and optimization based on observed data access patterns and resource availability. The proposed architecture is evaluated using a real industrial data pipeline scenario from fiber laser cutting domain.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42

**Keywords:** data pipelines · data management · data I/O. · distributed filesystems · cloud-edge continuum

## 1 Introduction

The rapid growth in scale and complexity of cloud infrastructures has driven the emergence of Edge and Internet of Things (IoT) computing, giving rise to what is commonly referred to as the compute continuum. This paradigm is particularly relevant for applications that rely on large volumes of sensor data and require low-latency, time-critical, AI-assisted data processing. By distributing computation across IoT devices, edge nodes, and cloud resources, the compute continuum aims to enable intelligent decision-making and to extract deeper insights from data generated at scale. Despite its promise, the design, deployment, and maintenance of robust *data pipelines* across this continuum remain challenging, particularly in light of the need for new programming paradigms and runtime systems for large-scale distributed infrastructures [4]. A data pipeline can be understood as a structured sequence of data-centric tasks responsible for ingesting, transferring, processing, and analyzing large-scale data distributed across IoT, edge, and cloud infrastructures. By effectively leveraging the computational and storage capabilities available throughout the continuum, these pipelines enable efficient and scalable data analysis tailored to the specific requirements of modern, data-intensive applications. However, this often requires addressing a wide range of technical, operational, and organizational issues, including heterogeneity of resources, security concerns, scalability, and performance optimization. These challenges, combined with limited expertise and resource constraints, represent a significant barrier to the widespread adoption of data pipeline technologies in distributed environments.

To address these issues, the SPICE (Smart Data Pipelines for the CognItive ComputE Continuum) research project [16] aims to enable secure, efficient, and intelligent processing of large-scale, distributed, and heterogeneous data, facilitating actionable insights across diverse application domains. The outcomes of this project will be released as an innovation and transformation incubator, fostering reuse and further exploitation by the research and industrial communities. The SPICE project is intended to support a broad range of use cases, from industrial machine data analytics and energy systems management to advanced medical imaging and diagnostics. The proposed work supports diverse data-intensive applications across industrial, energy, and healthcare domains. In each case, it enables the collection, integration, and analysis of large, heterogeneous datasets—ranging from high-frequency sensor streams and geospatial time-series to medical imaging—while ensuring security, data confidentiality, and interoperability of AI-driven insights. The platform facilitates scalable, multi-level data pipelines, supports federated and trustworthy data sharing, and provides robust decision support for domain experts. Across all use cases, a common theme is the need to process large volumes of distributed data in a secure, efficient, and intelligent manner, enabling domain experts to extract actionable insights

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42

with confidence. This paper introduces an architecture for data management and I/O provisioning that supports the execution of high-performance data pipelines across the cloud–edge continuum, leveraging Onedata [12] as the underlying data access and management platform. In particular, the proposed work delivers several key novelties: an Integrated Data Access Layer for the Cloud–Edge Continuum that unifies heterogeneous storage and data access models and provides transparent access across geographically distributed data sets; Dynamic Smart Data Placement mechanisms designed to improve pipeline execution efficiency through proactive data replication, currently implemented at the architectural level and partially realized in the prototype; and the establishment of Distributed Data Access Platform as a foundation for trusted and interoperable analytics.

While the proposed architecture builds upon existing systems such as Onedata and Airflow, its novelty lies in their tight integration within a unified, continuum-aware framework that combines federated data access, intelligent data placement, and pipeline orchestration. In contrast to existing approaches that treat these aspects independently, the proposed system enables coordinated optimization across data management and execution layers, which is essential for geo-distributed edge–cloud environments.

The paper is structured as follows. Section 2 presents the related work covering data management solutions applicable in the area of data pipelines and workflows, as well as relevant technologies to the proposed work. Section 3 briefly introduces the SPICE project. Section 4 presents the architecture of the proposed solution, including all components relevant for data pipeline execution, and Section 5 evaluates the framework prototype on an actual data pipeline. Finally, Section 6 concludes the paper with a brief summary and plans for future work.

## 2 Related work

This section reviews the state of the art in data processing pipeline execution, as well as distributed storage solutions designed to support large-scale computations across heterogeneous environments spanning the compute continuum.

### 2.1 Data processing pipelines

The design and execution of data pipelines have been extensively studied in the context of large-scale data processing and analytics. Early data pipelines were largely static and batch-oriented, limiting their ability to handle dynamic workloads and heterogeneous data sources. Systems such as Hadoop, MapReduce, Hive, and Pig provided scalable processing but lacked flexibility and low-latency support, while Dryad allowed structured DAG-based execution but remained constrained to batch workflows. In contrast, modern frameworks provide more dynamic and unified processing capabilities [14]. For example, Spark combines in-memory computation with support for both batch and streaming workloads [19], Flink provides stateful, low-latency stream processing [3, 14], and Beam decouples pipeline definition from execution, allowing the same workflow to run on

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42

multiple engines. Workflow orchestration and cloud-native solutions tackle the challenges of reproducibility, scalability, and heterogeneity. Tools like Apache Airflow [1], Luigi, Prefect, and Dagster manage complex dependencies, while cloud services such as AWS Step Functions, GCP Workflows, Azure Data Factory, and Kubeflow Pipelines provide elastic, containerized, and serverless execution. Hybrid frameworks (e.g., Delta Lake, Hudi) and domain-specific pipelines (e.g., Nextflow, Snakemake) further integrate batch and streaming semantics, enabling resilient, portable, and flexible workflows. While these frameworks are highly effective for cloud- or cluster-based deployments, they do not natively support the seamless execution of pipelines across geo-distributed environments.

### 2.2 Distributed data management

To overcome these limitations, distributed data management has been explored to enable efficient access, replication, and coordination of data across heterogeneous and geographically dispersed infrastructures [12, 5, 15]. Several frameworks have been developed to provide these capabilities. For example, Onedata [12] provides a unified, virtualized data layer that enables transparent access to distributed storage resources, supporting replication, data locality, and high-throughput I/O across the computing continuum. Eclipse Dataspace Components (EDC) [5] addresses complementary challenges by enabling secure, policy-driven data exchange across organizational boundaries, implementing trust, governance, and technical interoperability for EU Common Data Spaces. Distributed object storage solutions such as MinIO [7] provide scalable, high-performance S3-compatible storage that can be deployed across multiple nodes or sites, supporting erasure coding, data durability, and high-throughput access for geo-distributed workloads. Other tools, including Apache Hudi [2], and lakeFS [17], offer transactional storage, change-data capture, and versioning to support consistent, reproducible, and reliable data workflows. These frameworks provide essential building blocks for distributed data access and management. However, to the best of our knowledge, there is a lack of integrated solutions for orchestrating geo-distributed data pipelines. This highlights the need for architectures that combine advanced data management with intelligent pipeline design and execution. Building on these foundations, this work introduces a unified architectural framework that integrates semantic reasoning, smart data management, and continuum-aware orchestration.

## 3 SPICE project architecture

The principles of operation of the SPICE platform can be conceptually divided into 3 main phases (Fig.1); i.e., abstract data pipeline definition, mapping, and execution.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42

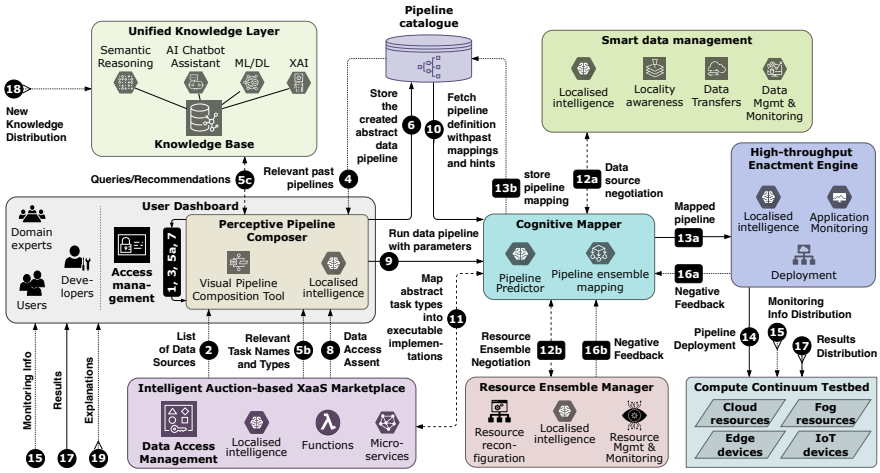



Fig. 1: Principles of operation of the SPICE platform.

### 1) Definition of an Abstract Data Pipeline (steps 1-8)

The user starts in the User Dashboard by opening (1) the Perceptive Pipeline Composer, which provides smart hints to simplify data pipeline composition. An up-to-date list of available data sources is retrieved (2) from the XaaS Marketplace, from which the user selects (3) the data to be processed. Based on the selected data type, the Composer queries (4) the Pipeline Catalogue for existing abstract data pipelines (ADPs), which can be executed directly or adapted to compose a new ADP as a sequence of task types (5b) supplied by the Marketplace. During composition (5a), the user is assisted (5c) by contextual recommendations from the Unified Knowledge Layer and a chatbot assistant. Completed ADPs are stored (6) in the Pipeline Catalogue and can be executed by specifying concrete data sources, user objectives and constraints, after which the Marketplace verifies (7) access rights and grants (8) consent for data access.

### 2) Mapping Abstract Data Pipeline on resources (steps 9-13)

In the mapping phase, the request to execute an ADP together with the selected data sources and user objectives and constraints is passed (9) to the Cognitive Mapper, which retrieves (10) the pipeline definition and relevant mapping history from the Pipeline Catalogue. The Mapper transforms the ADP into a specific data pipeline by selecting (11) suitable task implementations from the Marketplace, obtaining data locations (12a) from Smart Data Management (SDM) and available computing resources (12b) from the Resource Ensemble Manager (REM). Based on this information, it generates an optimized mapping and execution schedule using global optimization and lightweight runtime heuristics, negotiates service level objectives with SDM and REM, and forwards (13) the

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42

resulting pipeline mapping and schedule to the Enactment Engine, while storing the mapping in the Pipeline Catalogue.

**3) Deployment and execution phase (steps 14-19)**

In the deployment and execution phase, the Enactment Engine (EE) deploys and executes (14) the pipeline according to the Cognitive Mapper's instructions and continuously monitors (15) its execution. Monitoring data is distributed to relevant components (i.e., User Dashboard, REM, and SDM), while runtime errors that cannot be fixed either by EE or REM are reported back (16a, 16b) to the Cognitive Mapper for corrective actions such as rescheduling the affected pipeline parts. Execution results are delivered (17) to users or their applications, and collected monitoring data is analyzed using AI/ML methods to enrich system knowledge (18). Users may also request explanations (19) of execution results based on workflow provenance and explainable AI techniques. In this paper, we focus on the Smart Data Management aspect of the SPICE platform.

## 4 Unified data management framework for data pipelines

In order to support the wide variety of data pipelines and workflows from both scientific and industrial domains supported by the SPICE platform, the underlying data management framework must address several requirements, such as providing transparent and secure access to data distributed across multiple infrastructures; handling both POSIX-style as well as object-style data access; supporting rich metadata operations and data anonymization; ensuring sensitive data never leaves specified infrastructure or geographical boundaries and support Data Management Plans and follow FAIR principles.

### 4.1 Onedata as the federated data management middleware for data pipelines

The basis for the data management platform in SPICE is Onedata, an open-source distributed data management system that provides transparent access to data stored on geographically distributed storage resources, making it an ideal candidate as a data provisioning solution for data pipelines spanning the Edge-Cloud continuum. Onedata enables high throughput access to data through standardized interfaces including FUSE-based POSIX, REST, S3 and CDMI, hiding from the users low-level storage interfaces, and allows seamless execution in multi-provider environments. The high performance achieved by Onedata is due to the ability to directly access the underlying storage for access requests that have a direct connection to the storage locally (e.g., within a site). Currently it supports such backend storages as POSIX, NFS, S3, Ceph, GlusterFS, OpenStack Swift, WebDAV and HTTP. Conceptually, the system is built around a concept of a data space, which can be seen as a distributed folder, that can span several data centers and can be shared among several users. Each data center

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42

can deploy its own instance of Onedata Oneprovider service, and support user spaces with specific storage resources and quotas. Multiple Oneprovider services can be federated in a Onedata Onezone service, which provides authentication and provider discovery. Files can be replicated within a user space among the Oneprovider instances either automatically (based on read requests through any of the interfaces), based on QoS rules which can constrain data placement as well as manually. The transfer component features algorithms for dynamic adaptation to network conditions by adjusting the number of connections to a peer node, and the number of requests in the queue of each connection, by analyzing current and past throughput. Furthermore, Onedata is focused on collaboration between users, enabling secure data sharing based on access tokens, which can be used for fine-grained authentication and authorization control, as well as supporting open data access communities by providing interfaces and features for FAIR data publishing and access [9], such as handle-based identifier management and registration (e.g. DOI), as well as the OAI-PMH protocol for metadata harvesting. Onedata has been deployed and tested in several European infrastructures such as EGI DataHub, Galaxy [10] and EOSC [13].

### 4.2 Smart Data Management architecture

To address the requirements outlined in the introduction to Section 4, we introduce a Smart Data Management architecture (Fig. 2) that extends the capabilities of Onedata with additional components for governance, anonymization, and intelligent data movement across the cloud–edge continuum. The architecture provides a unified and decentralized data layer supporting both operational efficiency and compliance requirements. The Decentralized Virtual Data Layer

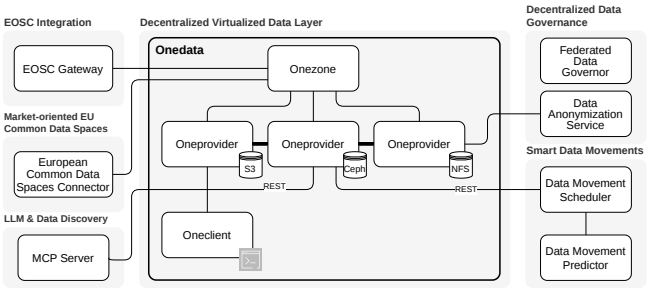



Fig. 2: Overall architecture of SPICE data management framework

encompasses a group of components that together form the foundation of the smart data management layer built on Onedata. It includes all existing capabilities as well as any additional functionality that will be developed and integrated as part of the SPICE project, ensuring a flexible and distributed approach to data access and management while enforcing data sensitivity requirements and

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42

applicable privacy policies. Decentralized Data Governance components focus on introducing higher-level data management capabilities addressing aspects such as the creation and handling of data management plans, as well as data anonymization, with the goal of supporting responsible, compliant, and well-structured data usage across the platform. The Data Anonymization Service aims to optimize the privacy–utility trade-off by minimizing the information loss introduced by the anonymization process while achieving a target level of privacy protection [11]. Because the anonymized datasets will support downstream time-series analytics (forecasting, anomaly detection, or predictive maintenance), the anonymization pipeline must preserve the statistical structure that these tasks rely on (e.g., temporal dependencies, correlations, marginal distributions, and event/rare-pattern characteristics). Moreover, personal and quasi-identifying attributes must be safeguarded through a continuous assessment of residual disclosure risk using quantitative privacy metrics (e.g., re-identification/linkage risk, attribute inference risk, or distributional similarity measures), enabling evidence-based compliance with applicable regulations and robust data-governance practices. Finally, Smart Data Movements optimizes how data is placed and transferred across the infrastructure, based on decisions made by Data Movement Predictor (described in detail in section 4.4).

### 4.3 Data management across data pipeline life-cycle

The Smart Data Management subsystem provides a unified data layer that enables SPICE pipelines to access, share, and govern data across distributed providers and execution environments. By leveraging Onedata, it creates a federated single data namespace that abstracts storage heterogeneity, enforces data-access and locality policies, and ensures that data remains available wherever pipeline components execute. The data pipeline execution from the data management layer perspective is presented in the figure 3. Pipeline input data is first ingested to Onedata, deployed in the Edge infrastructure, either directly from the internal software or as part of the pipeline execution. Every data access (input and output) is handled by Onedata through the custom *airflow-onedata-provider* component, which abstracts the data access operations for Airflow tasks. The tasks can either use object-style REST-based data access (for small files and non-critical latency) as well as POSIX-style access provisioned by the Onedata Oneclient component, which exposes the specified data for Airflow tasks via a FUSE-based mountpoint (either directly if the storage where that data is replicated to is in the same infrastructure as the execution environment or through a high-performance proxy mechanism). After the acquisition and preprocessing tasks, which can be executed in the edge infrastructure, are complete, tasks that should run in external infrastructure (Cloud) can be scheduled by Airflow automatically, and data will be replicated to the relevant Onedata instance on-the-fly as the tasks are executed. In order to improve the performance and minimize data access latency, several data access metrics are monitored for all data access operations and stored in a Prometheus database using the Open Telemetry protocol [6]. The collected metrics can be used after enough data is collected,

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42

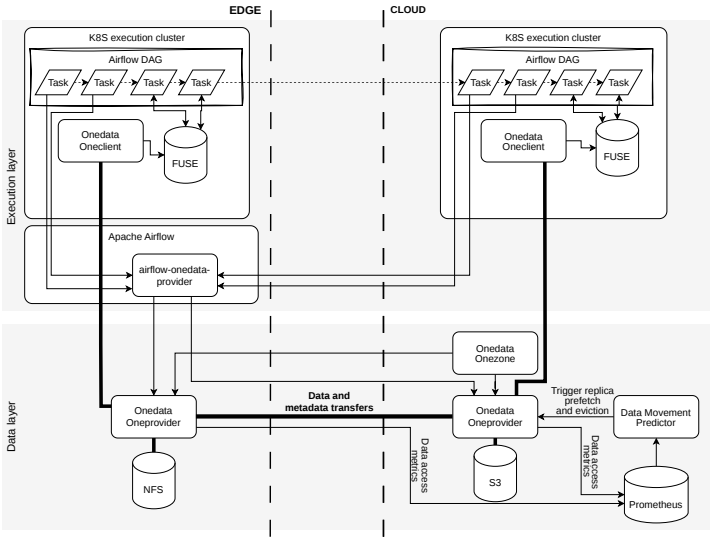



Fig. 3: Data I/O in Edge-Cloud data pipeline

and the Data Movement Predictor can automatically prefetch the data to the appropriate infrastructure before the task's execution starts.

### 4.4 Data movement prediction

The *Data Movement Predictor* (DMP) is a core component of the Smart Data Movements subsystem in Onedata, responsible for optimizing data placement and transfer decisions across Edge and Cloud infrastructures in support of data pipeline execution. Its primary objective is to proactively adapt data placement to observed and predicted access patterns, ensuring that workflow tasks meet their required Service Level Objectives (SLOs) while minimizing access latency and unnecessary data transfers. During execution, fine-grained data access metrics are continuously collected at provider-side clients, including Oneclient instances and object access endpoints. These metrics are exported via the OpenTelemetry protocol and stored in a Prometheus monitoring infrastructure. For each accessed data object, the collected metrics include a unique object identifier, the current physical location(s) of the object, and the observed response time. In addition, the DMP considers required SLOs, which may be explicitly specified by users or derived from other components of the SPICE project. Based on these inputs, the predictor computes a priority score, reflecting the deviation between observed access performance and the required SLO, following established approaches where performance prediction is used to guide runtime optimization decisions [8]. Objects with higher priority scores indicate suboptimal placement with respect to upcoming or ongoing task execution. The Data Movement Predictor periodically aggregates and analyzes metrics collected from

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42

multiple providers and execution environments, encompassing both Edge and Cloud infrastructures. The aggregated view enables the DMP to identify data objects whose current placement is likely to introduce performance bottlenecks for subsequent pipeline stages. Based on the computed priority scores, objects are ranked and proposed as candidates for data movement actions. Beyond reactive analysis, the DMP incorporates predictive modeling to estimate future data accesses associated with pipeline execution. Using historical access traces, workflow execution metadata from Airflow, and cross-cluster observations, the predictor derives a prediction signal for each object, indicating the likelihood and expected locality of upcoming accesses. This prediction enables proactive data prefetching to the infrastructure where tasks are scheduled to execute, prior to task start. Using the combined information from priority scoring and predicted access signals, the Data Movement Predictor recommends and coordinates 3 types of placement actions: *replication*, *migration*, and *eviction*. Replication is favored when predicted access patterns indicate concurrent or repeated access from multiple infrastructures; migration is applied when access demand persistently shifts toward a specific execution site; and eviction is triggered when predicted future accesses fall below defined thresholds and storage resources must be reclaimed. Movement actions are scheduled in descending order of priority score, ensuring that data with the highest performance impact is considered first. However, final execution decisions are conditioned on predicted access behavior, preventing unnecessary movements caused by transient access spikes. Through this integration of monitoring-driven analysis and prediction-aware decision making, the Data Movement Predictor enables Onedata to transparently prefetch and place data in advance of task execution, thereby reducing latency and improving the overall efficiency of data pipeline execution across Edge and Cloud environments.

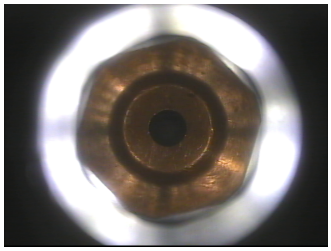
## 5 Industrial use case evaluation results

### 5.1 Use case description: Machine Data Analysis for Proactive Customer Support in Fiber Laser Cutting Centers

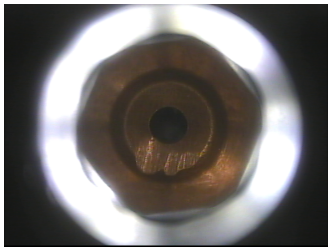
The evaluated use case focuses on machine data analysis for proactive customer support in fiber laser cutting centers, aiming to implement an automated framework for detecting consumable wear and optimizing technological parameters in CNC laser cutting systems. As laser cutting performance depends heavily on process parameters, material properties, and the condition of consumables such as the cutting nozzle and protection glass, progressive wear and material variability can lead to quality degradation and require frequent adjustments. Current practices rely on highly trained operators and trial-and-error parameter tuning, which is time-consuming, machine-specific, and often results in inconsistent quality and increased service interventions. To address these challenges, the proposed solution introduces automated monitoring of nozzle damage and protection glass contamination through an inspection workflow in which the cutting head is positioned at a camera station, grayscale images of the nozzle are captured periodically (approximately every 10 minutes), and transmitted

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42

for analysis, enabling more reliable, adaptive, and scalable process control with reduced dependence on manual expertise.



(a) Nozzle in nominal state



(b) Nozzle in damaged state

Fig. 4: Examples of cutting nozzle conditions. *Images courtesy of MicroStep, spol. s r.o.*

The nozzle damage classification training pipeline (5) consists of 3 sequential phases, performed in parallel on 2 training data sets - nominal nozzle images (4a) and damaged nozzle images (4b):

1. **Anonymization (Edge):** Since the data used for processing and model training is treated as a competitive advantage, it has to be anonymized before it can leave the edge infrastructure.
2. **Pre-processing (Cloud):** Image preparation including cropping, scaling, normalization, and grayscale adjustments.
3. **Classification (Cloud):** Training of a Convolutional Neural Network (CNN) for damage recognition and classification.

Each step in the data pipeline (except for the final training task) is performed on a directory containing a set of training images. Before the anonymization phase, the input images and their metadata are only stored in the Onedata Oneprovider on the Edge side of the deployment. As the preprocessing phase starts, when a task opens an anonymized file in the Cloud part of the infrastructure, the file blocks are automatically transferred by Onedata from Edge to Cloud.

Similarly, the nozzle damage condition classification pipeline (6) consists of 4 sequential phases:

1. **Anonymization (Edge):** Again, the data has to be anonymized before it can leave the Edge infrastructure.
2. **Pre-processing (Cloud):** Image preparation including cropping, scaling, normalization, and grayscale adjustments.
3. **Classification (Cloud):** Application of a Convolutional Neural Network (CNN) for damage recognition and classification.
4. **Post-processing (Edge):** Refinement of classification results and generation of machine-readable output (e.g., JSON format).

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42

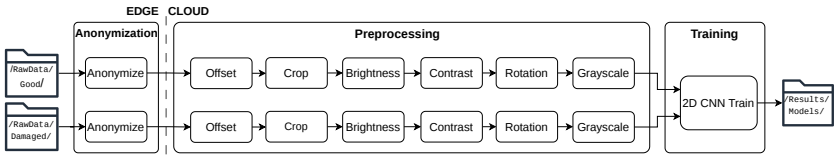



Fig. 5: Nozzle condition model training data pipeline.

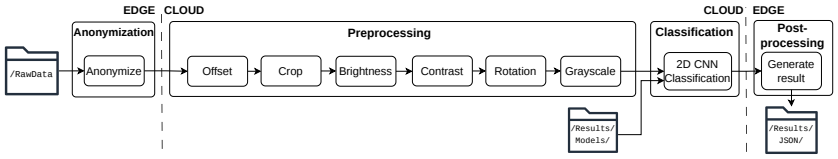


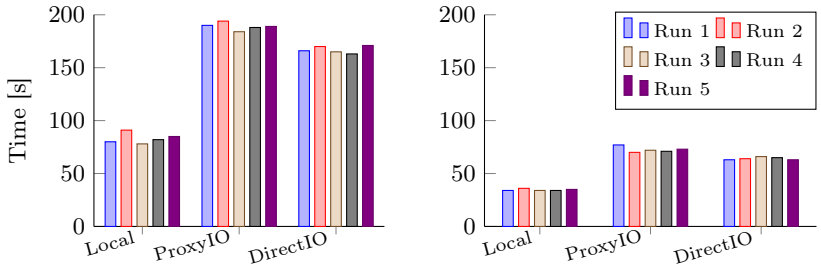

Fig. 6: Nozzle condition model classification data pipeline.

In this case, each task works on a single image, and the resulting classification result is stored in a Onedata space and attached as metadata to the original image in JSON format.

### 5.2 Evaluation results

In order to evaluate the applicability of the proposed solution for the use case, in particular the overhead of the network filesystem such as Onedata, we decided to use the model training data pipeline. We deployed a standalone Kubernetes cluster on 1 physical machine with the following specification: AMD Ryzen 9 7950X 5.8Ghz (32 vCPU), 64GB RAM, NVIDIA GeForce RTX 3090 GPU and Samsung SSD 980 PRO NVMe disk drive. We tested 3 data access modes: local filesystem, Onedata Oneclient Proxy IO, and Onedata Oneclient Direct IO. In the local file system mode, we created a shared Docker volume, which was used by all Airflow DAG tasks to read and write files (equivalent to native local filesystem performance). The Onedata Oneclient Proxy IO and Direct IO access modes were based on the Onedata Oneclient, which creates a FUSE mountpoint exposing the Onedata virtual filesystem across any network, while the local applications can treat it as a regular POSIX filesystem. This was necessary as current implementations of workflows only support POSIX filesystem interface, and cannot be evaluated on other types of distributed storages such as S3 or Ceph without refactoring. The Proxy IO and Direct IO modes in Oneclient differ in how Oneclient performs `read()` and `write()` operations. In case of Proxy IO, these operations are sent over a binary protocol to the Onedata Oneprovider service, which then performs these operations on the actual storage and returns results back to Oneclient. This introduces some latency, however enables to run Oneclient in locations from which the actual storage is not available. In case the storage can be made available to a particular instance of Oneclient, either as POSIX mountpoint or using some networked storage as NFS, Ceph, S3 and oth-

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42




(a) Single-threaded (b) Multi-threaded

Fig. 7: Training pipeline execution time comparison across access modes.

ers, Oneclient can execute the `read()` and `write()` operations directly on that storage, while only performing the metadata operations (e.g. `open()`, `close()`, `stat()`) through Oneprovider in order to ensure filesystem consistency.

The training data set consisted of 187 operational (good) nozzle images and 72 damaged nozzle images. The image sizes ranged from 300 to 600 KiB. The evaluation results were collected as follows. For each data access mode, we have executed the classification pipeline 5 times on the same data set and measured the overall pipeline execution time. We repeated this process for the pipeline in which tasks run in single-thread mode (each image in the directory is processed sequentially) as well as multi-thread mode (images are processed in a thread pool of size equal to the number of vCPUs on the test machine, i.e. 32). The single-thread results are presented in Figure 7a and multi-thread results are presented in Figure 7b. The results show that this particular pipeline first of all significantly benefits from being executed with multiple threads per task, and that the Onedata network filesystem introduces overhead necessary to map the POSIX operations to a distributed filesystem. The improvement, when using Direct IO instead of Proxy IO (i.e. performing the reads and writes directly on storage) is measurable, however in the case of a workload composed of many small files, the majority of overhead is caused by filesystem metadata operations (i.e. creating, opening files and listing directories), part of which is inherent to FUSE itself [18]. Another important aspect of the additional overhead is related to the replication of data between Edge and Cloud, necessary when tasks in the Cloud start reading files stored in the Edge part of the infrastructure. Currently, predictive and proactive replication is not yet enabled in the evaluated prototype, and files are replicated only upon access by tasks. As a result, the evaluation does not yet capture the potential performance benefits of prediction-driven data placement.

From a broader perspective, the proposed approach differs from classical centralized or cloud-only data pipeline solutions in several key aspects. Traditional approaches typically assume a single administrative domain and rely on centralized storage systems, which simplifies data access but introduces limitations in geo-distributed scenarios due to increased latency and bandwidth constraints. In

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42

contrast, the proposed architecture leverages federated data management and locality-aware data access, enabling data to remain closer to its source while still supporting distributed processing. While the current evaluation focuses on a single-site deployment, the architectural design directly targets challenges such as cross-site data movement, heterogeneous resources, and decentralized control, which are not addressed by conventional approaches.

## 6 Conclusions

This paper presented an architecture for data management and I/O provisioning that enables efficient execution of high-performance data pipelines across the cloud–edge compute continuum. By combining federated data management, intelligent data placement, and continuum-aware orchestration within a unified platform built on Onedata, the proposed approach addresses key challenges related to heterogeneity, distributed data sources, and performance-sensitive data movement. The integration of runtime monitoring and predictive data placement through the Data Movement Predictor further enhances the system's ability to adapt to dynamic workload characteristics and evolving access patterns. The evaluation using a representative industrial machine data analysis data pipeline demonstrates that the proposed solution provides a practical and scalable foundation for distributed pipeline execution.

Future work will focus on extending predictive capabilities for data movement, incorporating tighter integration between data pipeline schedulers and data placement decisions, and evaluating the architecture in fully distributed, multi-site deployments under realistic network conditions. Additional efforts will address the optimization of metadata-intensive workloads and the optimization of metadata operations latency in Onedata.

**Acknowledgments:** Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I02-03-V01-00012. BK, RS and JK were partially supported by the Ministry of Education and Science funds assigned to AGH University. SD, MB and LH were partially supported by the Slovak Research and Development Agency under contract No. APVV-23-0430, and by VEGA grant no. 2/0081/26.

## References

1. Apache Software Foundation: Apache Airflow. https://airflow.apache.org (2023), [Accessed 16-01-2026]
2. Apache Software Foundation: Apache Hudi. https://hudi.apache.org/ (2025), [Accessed 16-01-2026]
3. Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S., Tzoumas, K.: Apache Flink: Stream and batch processing in a single engine. The Bulletin of the Technical Committee on Data Engineering **38**(4) (2015)
4. Costa, G.D., Fahringer, T., Rico-Gallego, J., Grasso, I., Hristov, A., Karatza, H.D., Lastovetsky, A.L., Marozzo, F., Petcu, D., Stavrinides, G.L., Talia,

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42

D., Trunfio, P., Astsatryan, H.V.: Exascale machines require new programming paradigms and runtimes. Supercomput. Front. Innov. **2**(2), 6–27 (2015). https://doi.org/10.14529/JSFI150201
5. Eclipse Foundation: Eclipse Dataspace Components. https://eclipse-edc.github.io/ (2021), [Accessed 16-01-2026]
6. Koniaris, B., Sinclair, D., Mitchell, K.: Dancemark: An open telemetry framework for latency-sensitive real-time networked immersive experiences. In: VR Workshops. pp. 462–463. IEEE (2024)
7. MinIO, Inc.: Minio: High performance object storage. https://min.io (2024), [Accessed 16-01-2026]
8. Nadeem, F., Yousaf, M.M., Prodan, R., Fahringer, T.: Soft benchmarks-based application performance prediction using a minimum training set. In: 2006 Second IEEE International Conference on e-Science and Grid Computing (e-Science'06). pp. 71–71 (2006). https://doi.org/10.1109/E-SCIENCE.2006.261155
9. Opioła, Ł., Jarosz, K., Dutka, Ł., Słota, R.G., Kitowski, J.: Group membership management framework for decentralized collaborative systems. Computer Science **23**(4) (Nov 2022). https://doi.org/10.7494/csci.2022.23.4.4642
10. Opioła, Ł., Walkowicz, B.: Getting started with Onedata distributed storage (Galaxy Training Materials), https://training.galaxyproject.org/training-material/topics/galaxy-interface/tutorials/onedata-getting-started/tutorial.html, [Accessed 16-01-2026]
11. Ortega-Fernandez, I., Martinez, S.E.K., Orellana, L.A., Soldatos, J., Kyriazis, D.: Large Scale Data Anonymisation for GDPR Compliance. Big Data and Artificial Intelligence in Digital Finance p. 325 (2022)
12. Orzechowski, M., Wrzeszcz, M., Kryza, B., Dutka, Ł., Słota, R.G., Kitowski, J.: Global access to legacy data-sets in multi-cloud applications with Onedata. In: Wyrzykowski, R., Dongarra, J., Deelman, E., Karczewski, K. (eds.) PPAM Intern. Conf. pp. 305–317. Springer International Publishing, Cham (2023)
13. Orzechowski, M., Opioła, Ł., Martínez, I.L., Ioannides, M., Panayiotou, P.N., Dutka, Ł., Słota, R.G., Kitowski, J.: Integrated data, metadata, and paradata management system for 3d digital cultural heritage objects: Workflow automation, federated authentication, and publication. Future Generation Computer Systems **174**, 107964 (2026). https://doi.org/10.1016/j.future.2025.107964
14. Pedratscher, S., Samani, Z.N., Poveda, J.A., Fahringer, T., Etheredge, M., Younesi, A., Barrionuevo, J.J.D., Thoman, P.: Streamline: Dynamic and resource-efficient auto-tuning of stream processing data pipeline ensembles. Internet of Things p. 101731 (2025)
15. Poveda, J.A., Ebner, M.F., Fahringer, T., Samani, Z.N., Etheredge, M., Pedratscher, S., Saurabh, N.: Smartkv: A cost-effective and low-latency geo-distributed key-value store for the computing continuum. Future Generation Computer Systems p. 107857 (2025)
16. SPICE Platform: SPICE Platform. https://spice-platform.eu/ (2026), [Accessed 16-01-2026]
17. Treeverse, Inc.: lakeFS. https://lakefs.io/ (2025), [Accessed 16-01-2026]
18. Vangoor, B.K.R., Tarasov, V., Zadok, E.: To FUSE or not to FUSE: Performance of User-Space file systems. In: 15th USENIX Conference on File and Storage Technologies (FAST 17). pp. 59–72. USENIX Association, Santa Clara, CA (Feb 2017)
19. Zaharia, M., Xin, R.S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M.J., et al.: Apache Spark: a unified engine for big data processing. Communications of the ACM **59**(11), 56–65 (2016)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_42