Check for updates

SOFTWARE TOOL ARTICLE

ReProv: Extending REANA with WaaS capabilities and data provenance on the European AI-on-Demand Platform

[version 1; peer review: awaiting peer review]

Antonis Troumpoukis[1], Antonis Gkanios[1], Iraklis Klampanos[2]

[1]National Centre for Scientific Research Demokritos, Athens, Greece
[2]University of Glasgow, Glasgow, UK

v1 **First published:** 15 Jul 2026, **6**:246
https://doi.org/10.12688/openreseurope.24424.1
Latest published: 15 Jul 2026, **6**:246
https://doi.org/10.12688/openreseurope.24424.1

Abstract

Reproducibility remains a major challenge in computational research, particularly for workflow-based analyses executed across heterogeneous data, software, and computing environments. While workflow management systems such as REANA support reproducible execution through containerisation and workflow orchestration, they provide limited support for workflow registration, service-oriented reuse, and structured provenance management. This paper presents ReProv, a provenance capture and management layer that extends REANA with Workflows-as-a-Service (WaaS) capabilities and automated provenance capture based on the W3C PROV-O standard. ReProv enables researchers to register, discover, execute, and reuse Common Workflow Language (CWL) workflows while automatically recording the relationships between workflows, executions, data artefacts, and responsible users. Provenance information is stored in a structured form and can be explored through APIs, a Python SDK, and graph-based visualisations. The system has been integrated with the European AI-on-Demand (AIoD) Platform, enabling workflows and datasets to be linked with a broader ecosystem of AI resources. Two use cases demonstrate the approach: a machine-learning workflow based on the MNIST benchmark and a larger-scale climate-data workflow for next-day Mediterranean heatwave prediction. Results show that ReProv captures detailed execution lineage, including failed workflow executions, while introducing only negligible runtime overhead. By combining workflow lifecycle management, provenance-aware execution, and service-oriented workflow reuse, ReProv provides a practical framework for transparent, reproducible, and FAIR-aligned computational research.

Open Peer Review

Approval Status *AWAITING PEER REVIEW*

Any reports and responses or comments on the article can be found at the end of the article.

Keywords
Workflow Management System, Data Provenance, CWL, REANA, FAIR, AI-on-Demand Platform, Scientific Workflows

This article is included in the Horizon Europe gateway.

Corresponding author: Antonis Troumpoukis (antru@iit.demokritos.gr)

Author roles: Troumpoukis A: Conceptualization, Investigation, Methodology, Software, Writing – Original Draft Preparation, Writing – Review & Editing; **Gkanios A**: Software; **Klampanos I**: Conceptualization, Investigation, Methodology, Software, Supervision, Validation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This project has received funding from the Horizon Europe Framework Programme under grant agreement No [101070000](AN AI ON-DEMAND PLATFORM TO SUPPORT RESEARCH EXCELLENCE IN EUROPE [AI4EUROPE]).
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2026 Troumpoukis A *et al*. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Troumpoukis A, Gkanios A and Klampanos I. **ReProv: Extending REANA with WaaS capabilities and data provenance on the European AI-on-Demand Platform [version 1; peer review: awaiting peer review]** Open Research Europe 2026, **6**:246 https://doi.org/10.12688/openreseurope.24424.1

First published: 15 Jul 2026, **6**:246 https://doi.org/10.12688/openreseurope.24424.1

Introduction

Reproducibility and interoperability are foundational to credible scientific research, yet remain difficult to achieve in computational workflows that span diverse data formats, software environments, and execution platforms. Scientific analyses are increasingly composed of complex, multi-step workflows that depend on containerized software, heterogeneous computing resources, and collaborative data sharing. Despite the growing adoption of workflow management systems, researchers still face challenges in registering, discovering, and reusing workflows, as well as in systematically capturing provenance information needed to understand how results were produced.

The REANA platform [1], developed by CERN, has addressed some of these challenges by enabling users to describe, execute, and share reproducible analyses within containerized environments. REANA supports multiple workflow languages and can execute analyses across Cloud and HPC resources. However, while REANA effectively supports reproducible workflow execution, it does not natively provide workflow registry capabilities, Workflows-as-a-Service (WaaS) functionality, or structured provenance management aligned with semantic provenance standards.

To address these limitations, we developed ReProv, a provenance capture and management layer that extends REANA with Workflows-as-a-Service (WaaS) capabilities, workflow registration, provenance capture, and workflow lifecycle management. Our approach is inspired by the principles of the DARE platform,[1,2] which emphasizes workflow sharing, metadata-driven discoverability, FAIR-aligned reuse, and traceability of computational research artifacts. The new capabilities allow users to register workflows by name or identifier, discover and invoke them through APIs, and visualize complete data lineage across computational steps and resources.

The resulting system broadens REANA's role from a workflow execution engine into a service-oriented research infrastructure that supports transparency, reuse, and accountability. By integrating WaaS and provenance capabilities, it addresses critical gaps in workflow registry and provenance-management capabilities, providing a foundation for reproducible, FAIR-compliant computational science. These developments are currently being leveraged in the European AI-on-Demand Platform, enabling AI innovators to manage and execute large-scale, reproducible, AI-driven workflows. Through this integration, workflows, datasets, provenance records, and execution metadata can be connected to the broader European AI ecosystem, facilitating discovery, reuse, and collaboration.

This article is positioned as a software/methodological contribution: it describes the design, implementation, and evaluation of ReProv, an open provenance-aware WaaS system for reproducible scientific workflows built for REANA. More specifically this article contributes:

1. The design and implementation of ReProv, a provenance and capture management layer, targeting REANA.
2. A PROV-O–based provenance model[3] capturing workflows, executions, and agents.
3. Ongoing integration with the European AI-on-Demand Platform for metadata-driven reuse.
4. Visualization, REST API, and Python SDK support for workflow and provenance exploration.

ReProv does not replace REANA's execution capabilities but it rather complements them by adding persistent workflow registration, service-level invocation, execution-level and step-level provenance capture and visualisation, and AIoD resource linking. This separation of concerns allows REANA to remain the execution layer while ReProv provides the metadata, provenance, and service interfaces needed for workflow capturing and auditability.

This article is organized as follows. The following section, "Background", provides useful background information on the AI-on-Demand Platform, scientific workflows, the Workflows-as-a-Service paradigm, data provenance, and REANA. Section "Methods" presents the design and implementation of ReProv, including its architecture, provenance model, FAIR alignment, and operational aspects. Section "Use Cases" illustrates the capabilities of the system through representative use cases and experimental evaluation. Finally, Section "Conclusions/Discussion" concludes the paper and discusses future directions.

[1] https://reanahub.io

Background

The European AI-on-Demand platform

The European AI-on-Demand (AIoD) platform [2] is a collaborative, community-driven digital space designed to support the sharing, discovery, and reuse of AI-related resources, including datasets, tools, services, experiments, and educational material. As part of the European Commission's AI strategy, the platform promotes openness, transparency, trustworthiness, and interoperability within the European AI innovation ecosystem. AIoD has evolved through a series of EU-funded initiatives,[4–7] which collectively established the platform as a sustainable hub for AI research, collaboration, and knowledge exchange. The platform plays an important role in supporting the European scientific community by enabling researchers to exchange computational resources, share reproducible experiments, and collaborate across institutions and scientific domains. AIoD supports a broad community of stakeholders, including researchers, SMEs, public institutions, Digital Innovation Hubs, and EU-funded projects, by enabling access to shared AI resources and fostering collaborative development of reproducible and trustworthy AI applications.

Scientific Workflows and the WaaS paradigm

A *scientific workflow*[8] is the description of a process for accomplishing a scientific objective, usually expressed in terms of tasks and their dependencies. Typically, scientific workflow tasks are computational steps for scientific simulations or data analysis steps. A *scientific workflow system* is a specialized form of a workflow management system designed specifically to compose and execute a series of computational or data manipulation steps, or workflow, in a scientific application. Scientific workflow systems are generally developed for use by scientists from different domains like astronomy, earth science, and bioinformatics.

The *Common Workflow Language* (CWL)[9] is a standard for describing computational data-analysis workflows. The development of CWL is focused particularly on serving the data-intensive sciences. A key goal of CWL is to allow the creation of a workflow that is portable and thus may be run reproducibly in different computational environments. Also, the declarative nature of CWL allows workflows to be described in a clear and concise manner, focusing on what needs to be done rather than how to do it. CWL workflows define computational steps, their inputs and outputs, execution dependencies, and associated containerized software environments in a portable and platform-independent manner. Once a workflow is defined, it can be executed using workflow engines supporting the CWL standard, ranging from local execution tools such as cwltool [3] to distributed execution systems targeting cloud and HPC infrastructures. These execution systems provide portability, scalability, and integration with container technologies such as Docker [4] or Singularity [5].

Workflows-as-a-Service (WaaS)[10] is a paradigm in which computational workflows are exposed as reusable and remotely accessible services through standardized interfaces. Instead of treating workflows as isolated execution scripts, WaaS systems enable workflows to be persistently registered, discovered, invoked, and reused across users, applications, and research environments. This approach promotes modularity, interoperability, and reproducibility by allowing workflows to be shared and executed independently of the underlying execution infrastructure. In scientific computing, WaaS facilitates collaborative research and enables workflow-based analyses to be integrated into larger computational ecosystems and research platforms.

In WaaS environments, provenance information becomes particularly important because workflows are repeatedly executed, reused, and shared across users and computational environments, requiring transparent tracking of data lineage, execution context, and responsibility.

Modeling data provenance with PROV-O

Data provenance refers to the origin, lineage, and history of data, capturing its entire lifecycle from creation to its current state. It plays a crucial role in ensuring transparency, accountability, and trustworthiness by documenting how data was generated, manipulated, and used over time. Provenance has been recognized as a key enabler in many scientific domains such as bioinformatics, astronomy, and earth sciences. The literature consistently emphasizes that without provenance information, it becomes extremely difficult to validate scientific findings, reproduce computational results, or provide the level of accountability increasingly expected in European and international research initiatives.

[2] https://www.aiodp.eu
[3] https://github.com/common-workflow-language/cwltool
[4] https://www.docker.com
[5] https://apptainer.org

The W3C standard PROV-O (Provenance Ontology)[3] provides a structured way to model this information by representing the key components of provenance: *Entities* (data artifacts or things), *Activities* (processes that act on entities), and *Agents* (people or systems responsible for activities or entities). PROV-O defines relationships such as *used* (an activity used an entity as input), *wasGeneratedBy* (an activity produced an entity), *wasAssociatedWith* (an agent was responsible for an activity), and *wasAttributedTo* (an agent was responsible for an entity), enabling detailed tracing of how data evolves. Through such modeling, data provenance supports reproducibility, auditability, and informed decision-making in data-driven environments.

A high-level overview of the PROV-O concepts can be shown in Figure 1.

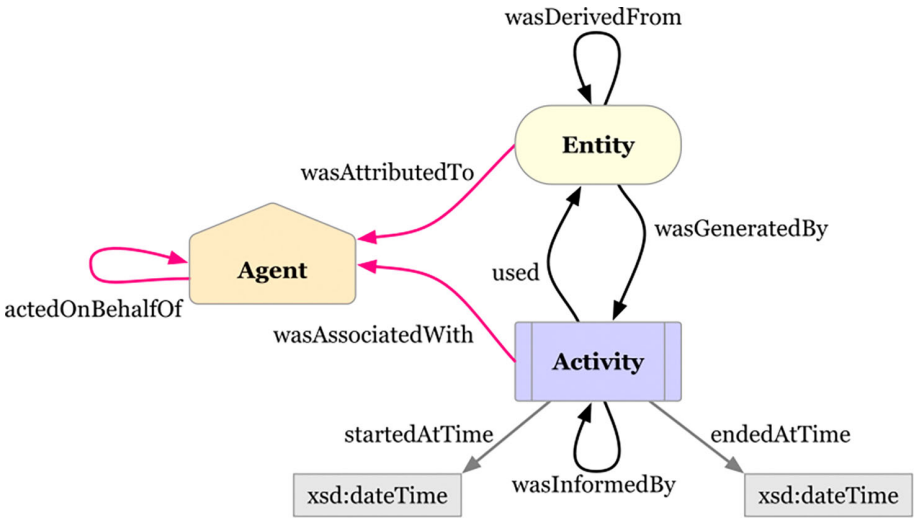

Figure 1. High-level overview of core PROV-O concepts and relationships (Image source: https://www.w3.org/TR/prov-o).

REANA

REANA (Reproducible Analyses)[11] is an open-source computational platform for the execution and management of reproducible computational workflows. By supporting workflow languages such as CWL and containerized execution environments, REANA enables workflows to be executed consistently across heterogeneous infrastructures, including local systems, cloud platforms, and high-performance computing environments. The platform emphasizes reproducibility, portability, scalability, and collaborative sharing of computational analyses.

REANA is particularly relevant for AI and data-intensive research, where complex workflows and machine learning pipelines must be executed reproducibly across different environments. By automating workflow orchestration and execution, REANA facilitates the sharing, validation, and reuse of computational experiments within collaborative research settings. However, while REANA provides robust support for reproducible workflow execution, it does not natively provide workflow registry capabilities or structured provenance management aligned with semantic provenance standards such as PROV-O.

Methods

System architecture

ReProv is a provenance-aware workflow management system that extends the REANA platform with WaaS capabilities and structured provenance tracking. Building upon technologies and standards defined in the previous section, the system combines workflow registration, execution orchestration, metadata management, and provenance capture within a modular API-driven architecture, while delegating containerized workflow execution to an existing REANA installation.

ReProv follows a service-oriented design in which workflows are treated as reusable resources that can be registered, executed, and shared through well-defined APIs. The system is based on three core principles: automated provenance capture aligned with the W3C PROV-O standard, reusable workflow execution through the WaaS paradigm, and interoperability with external platforms, particularly the AIoD Platform.

The architecture consists of four main components: (i) a RESTful API responsible for workflow and provenance operations, (ii) a persistent relational database for metadata, execution records, and provenance storage, (iii) a module for provenance tracking and visualization, and (iv) an authentication and authorization layer for secure multi-user access.

Figure 2 presents the high-level architecture of ReProv and its interaction with other systems.

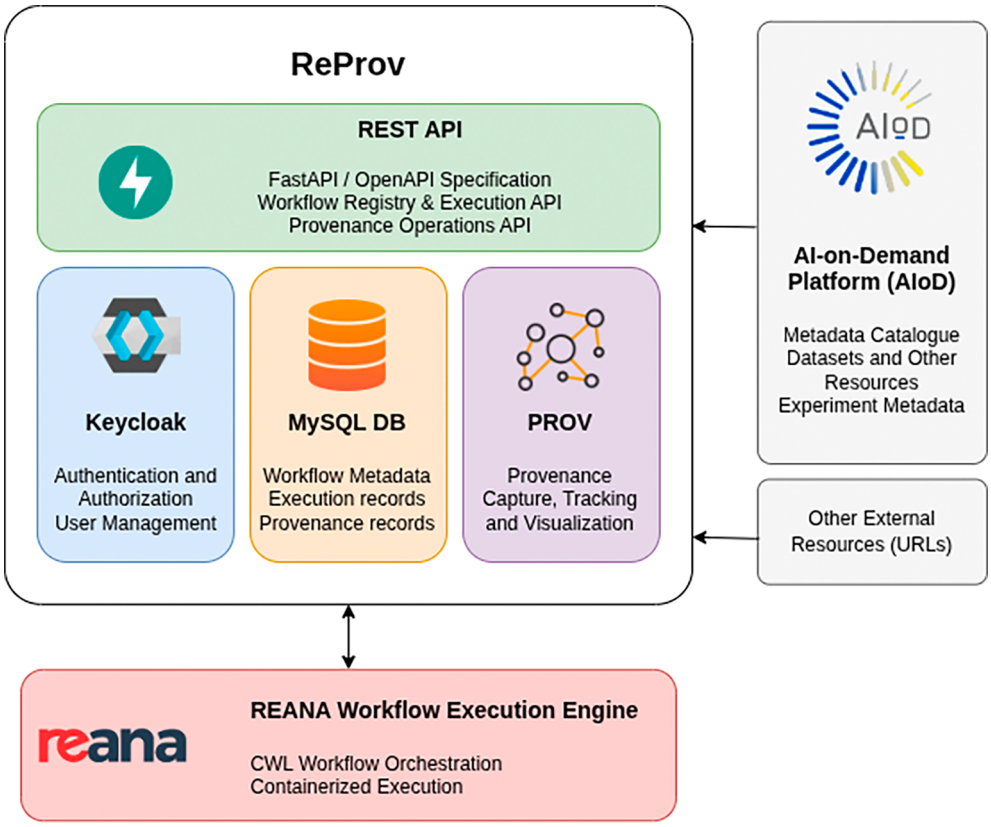

Figure 2. High-level architecture of ReProv, showing its API, database, provenance module, and authentication layer, and its integration with REANA for execution and AIoD for resource linkage.

Workflow lifecycle

ReProv manages workflows throughout their full lifecycle, including workflow registration, execution, provenance capture, and reuse. Workflows are treated as reusable research assets that can be persistently registered, executed multiple times, and associated with provenance information describing their execution history and produced artifacts.

Workflows described using CWL can be registered together with descriptive metadata such as workflow name and version. Once registered, workflows become reusable resources that can be discovered, referenced, and executed across different analyses and projects. This design follows the WaaS paradigm by decoupling workflow definition from workflow execution.

Registered workflows can be executed multiple times using different input parameters and datasets. ReProv coordinates workflow execution through the underlying execution infrastructure while maintaining associations between workflows, executions, generated outputs, and execution metadata. This enables workflows to be reused and integrated into larger computational pipelines.

ReProv automatically captures provenance information throughout workflow execution in order to support traceability, reproducibility, and accountability. Provenance records describe the relationships between workflow definitions, execution activities, intermediate and final data artifacts, and the users or systems involved in the execution process.

This information enables reconstruction of workflow lineage and supports transparent inspection of computational analyses.

ReProv additionally supports interoperability with external research platforms and distributed data sources, particularly with the AIoD Platform. Workflow inputs may reference externally hosted datasets and resources through AIoD dataset identifiers or persistent URLs, enabling workflows to directly consume datasets maintained outside the local execution environment.

Provenance modeling and tracking

Data provenance in ReProv is modeled according to the W3C PROV-O standard.[3] Provenance capture is performed automatically during workflow execution, enabling the reconstruction of complete workflow lineage.

To capture data provenance during workflow execution in our system, we make the following modelling decisions, based on the PROV-O model and its definitions of entities, activities and agents:

- Each workflow specification is treated as an *entity*, representing its definition and structure. Similarly, every file produced by a workflow step is considered an *entity*, enabling the tracking of all intermediate and final data artifacts. Datasets referenced from external catalogues, such as the AIoD platform, are likewise recorded as entities linked to the steps that consume them, so that data pulled from outside the local execution environment remains part of the captured lineage.

- The execution of the entire workflow, as well as the execution of each individual step, is modeled as *activities*, allowing us to document how and when data is processed or generated.

- The user who submits the workflow is represented as an *agent*, reflecting their role in initiating the execution. Since users typically belong to a broader organizational unit, the group to which the user belongs is also modeled as an agent, capturing collective responsibility.

To capture data provenance during workflow execution, ReProv records how workflow steps consume input data, generate intermediate and final outputs, and are associated with the users and software systems responsible for their execution. This information enables the reconstruction of complete workflow lineage, including the relationships between workflow definitions, execution steps, produced artifacts, and responsible agents.

The following RDF graph showcases a minimal yet complete provenance representation of a workflow execution using PROV-O, capturing agent responsibility, execution context, and data lineage.

```
@prefix prov: <http://www.w3.org/ns/prov#>.
@prefix ex: <http://example.org/reprov/>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
### Agents
ex:user a prov: Person.
ex:executionSoftware a prov: SoftwareAgent;
     prov:actedOnBehalfOf ex:user.
### Entities
ex:workflow a prov: Entity;
     prov:wasAttributedTo ex:executionSoftware.
ex:input_data a prov: Entity.
ex:output_data a prov: Entity;
     prov:wasGeneratedBy ex:execution.
### Workflow Execution
ex:execution a prov: Activity;
     prov:startedAtTime "2026-05-09T20:45:09"^^xsd:dateTime;
     prov:endedAtTime "2026-05-09T23:01:33"^^xsd:dateTime;
     prov:wasStartedBy ex:workflow;
     prov:used ex:input_data;
     prov:wasAssociatedWith ex:executionSoftware.
```

The RDF graph models the relationships between workflow definitions, execution activities, data artifacts, and responsible agents during workflow execution. The workflow execution is represented as an activity associated with both the workflow definition and the software agent responsible for execution. Input and output entities establish the lineage between consumed datasets and generated results, while timestamps capture the temporal context of the execution. Together, these relationships provide a compact provenance representation that supports traceability, reproducibility, and accountability of computational analyses.

In addition to storing provenance records in structured form, ReProv provides graph-based visualizations to make workflow lineage easier to inspect. These visualizations represent workflow executions as provenance graphs, where nodes correspond to workflow definitions, execution steps, data artifacts, users, and software agents, while edges represent dependencies such as data consumption, artifact generation, and responsibility relationships. The resulting diagrams allow users to quickly identify how outputs were produced, which intermediate artifacts were involved, and how different workflow steps are connected. This visual layer is particularly useful for debugging, auditing, and communicating computational analyses to users who may not directly inspect the underlying provenance metadata.

FAIR alignment

ReProv is designed to support the FAIR principles[12] for scientific workflows. In particular:

- *Findability* is supported through persistent workflow registration together with descriptive metadata, including workflow names, versions, and execution-related information.

- *Accessibility* is achieved through the API-driven architecture and accompanying Python SDK, which allows workflows, execution metadata, and provenance information to be programmatically queried and retrieved. The system additionally supports controlled access through authentication and authorization mechanisms based on Keycloak.

- *Interoperability* is facilitated through the adoption of established community standards, including CWL for workflow specification and the W3C PROV-O ontology for provenance representation.

- *Reusability* is strengthened through containerized execution, persistent workflow versioning, and systematic provenance capture. By recording workflow lineage, execution context, intermediate artifacts, and responsible agents, ReProv enables computational analyses to be inspected, reproduced, and reused in future studies.

Together, these capabilities support FAIR-compliant and reproducible workflow-based research.

Implementation

ReProv is implemented [6] as a containerized platform. All system components are deployed using Docker and orchestrated via docker-compose, ensuring portability and consistent runtime environments. ReProv supports workflows described using CWL.

The backend is implemented in Python using *FastAPI* [7], which provides the RESTful backend API and automatic generation of OpenAPI specifications for workflow and provenance operations. The API exposes endpoints for workflow registration, execution triggering, execution monitoring, provenance capture, and provenance visualization. In addition to the RESTful API. a Python SDK is provided that abstracts API interactions into high-level programmatic operations for workflow registration, execution, and provenance retrieval.

Authentication and access control are handled using *Keycloak* [8], which provides OAuth2/OpenID Connect-based identity management. Keycloak enables user authentication, group membership, and role-based access control, ensuring that workflow operations are securely scoped in multi-user environments.

Persistent storage is provided by a *MySQL relational database* [9], which stores workflow specifications, metadata, execution records, and provenance information. The database schema supports versioned workflow registration, multiple

[6] https://github.com/id-is/ReProv-API
[7] https://fastapi.tiangolo.com
[8] https://www.keycloak.org
[9] https://www.mysql.com

executions per workflow, and step-level execution tracking. Although provenance semantics follow the PROV-O model, a relational backend was selected to support efficient querying and interoperability with external systems.

For provenance visualization, ReProv uses the *prov Python library* [10] to generate graph-based representations of provenance records. These are rendered as PNG images depicting entities, activities, and agents, along with their relationships, providing an intuitive view of workflow structure and data lineage.

Operation

ReProv operates in a Linux-based environment with container support and requires access to a running REANA instance. Deployment involves starting the API backend, authentication service, and database as containerized services, with configuration managed through environment variables. This approach allows the system to be adapted to different infrastructures without modifying source code.

From the user's perspective, operation follows a service-oriented interaction model. Users authenticate through the API using credentials managed by Keycloak and register workflows described in CWL by uploading workflow specifications along with metadata such as name and version. Registered workflows become persistent resources that can be executed multiple times without redefinition.

To execute a workflow, a user submits an execution request referencing the workflow's registry identifier together with the required input parameters. ReProv forwards the corresponding CWL workflow specification and inputs to REANA, which performs the containerized execution and returns execution metadata, including workflow names, run identifiers, and execution status. During execution, users can query the API to monitor execution status and retrieve execution metadata.

Provenance capture is performed in two phases. During execution, a background monitor tracks each CWL step as it progresses through REANA and records its start and end timestamps, executed command, exit code, captured logs, and per-step resource usage (CPU time, peak memory, disk I/O) measured by a lightweight Python wrapper that ReProv inserts into every step at submission time. This means that provenance for a long-running pipeline becomes observable incrementally, rather than only at completion. Once execution finishes, the user requests provenance extraction for a given execution, at which point ReProv assembles the recorded step entries into PROV-O–compliant records. These records can be retrieved programmatically or visualised as graph-based diagrams (export as W3C PROV-JSON is already supported), and remain permanently associated with the execution to support retrospective analysis, auditing, and reproducibility.

ReProv also treats failures as first-class. When a step exits with a non-zero status, the monitor finalises the corresponding step record with a failed status, attaches the tail of its log as an error message, and stores its exit code. REANA halts the workflow on the failed dependency, so subsequent steps are not scheduled. The provenance captured up to the point of failure remains intact and is included in the PROV-O output, so a failed run still carries actionable evidence of where and why it stopped rather than producing an incomplete or inconsistent record.

Through its API-driven operation model, ReProv supports both interactive use (e.g., via web interfaces or notebooks) and automated integration into larger workflow ecosystems.

Typical user journey

Figure 3 illustrates a typical end-to-end workflow execution scenario in ReProv.

[10] https://pypi.org/project/prov

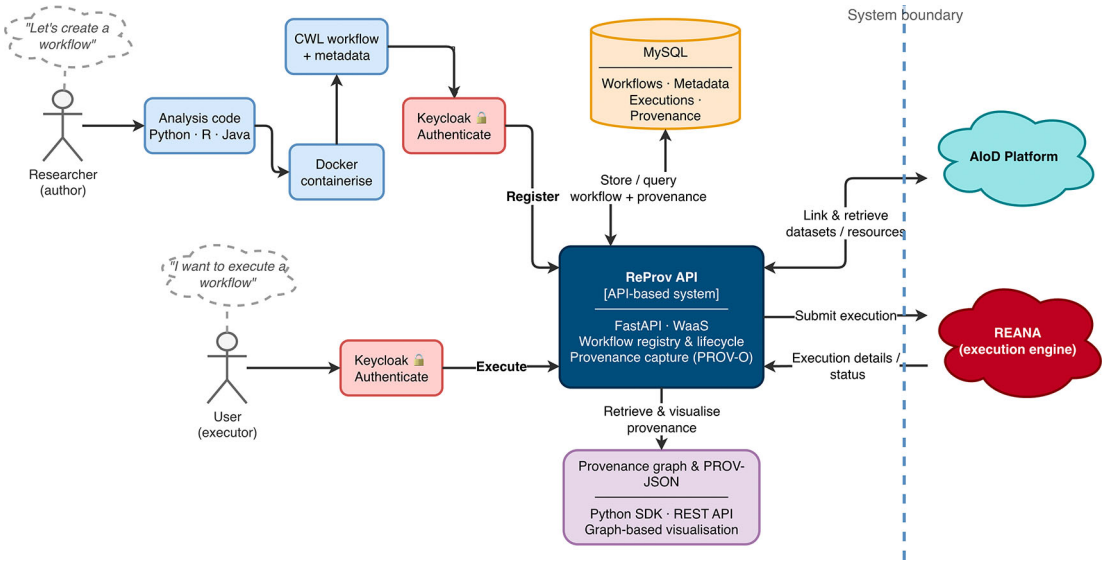

Figure 3. Typical ReProv user journey, showing how a researcher registers a CWL workflow and a second user executes it, with ReProv coordinating execution on REANA, linking resources via AIoD, and capturing provenance for retrieval and visualisation.

A typical user journey in ReProv begins with a *researcher* creating a computational workflow and registering it in the system together with descriptive metadata. Once registered, the workflow becomes a reusable resource that can be discovered and executed by other users through the platform.

A second *user* can subsequently execute the workflow by providing the required input data and execution parameters. ReProv manages the interaction with the underlying execution infrastructure and records information about the execution process, including generated outputs, intermediate results, and execution dependencies.

After execution completes, the system generates provenance information describing how the results were produced, which data and workflow steps were involved, and which users or systems participated in the execution. This information can be retrieved through the API or visualized as graph-based diagrams to support transparency, traceability, and reproducibility.

Use Cases

To illustrate the capabilities of ReProv, we present two representative use cases demonstrating provenance capture, visualization, and workflow monitoring in realistic execution scenarios.

Use Case 1: Provenance capture in successful and failed workflow executions

To illustrate the provenance capabilities of ReProv, we executed two variants of a simple machine-learning workflow based on the MNIST handwritten-digit dataset.[13] The workflow consists of three main stages: dataset preparation, model training, and model evaluation. The first workflow completes successfully, while the second intentionally introduces a validation step that terminates with a non-zero exit code before evaluation.

The successful workflow contains three CWL steps:

- First, downloading and preparing the training and test datasets (download_ds).
- Second, training a neural-network model (train).
- Third, evaluating the trained model and producing a results report (evaluate).

Figure 4 presents the provenance graph generated by ReProv for this execution. The graph captures the complete lineage of the workflow, including the executed activities, intermediate artifacts (datasets and trained model), final outputs, and the responsible user and execution environment.

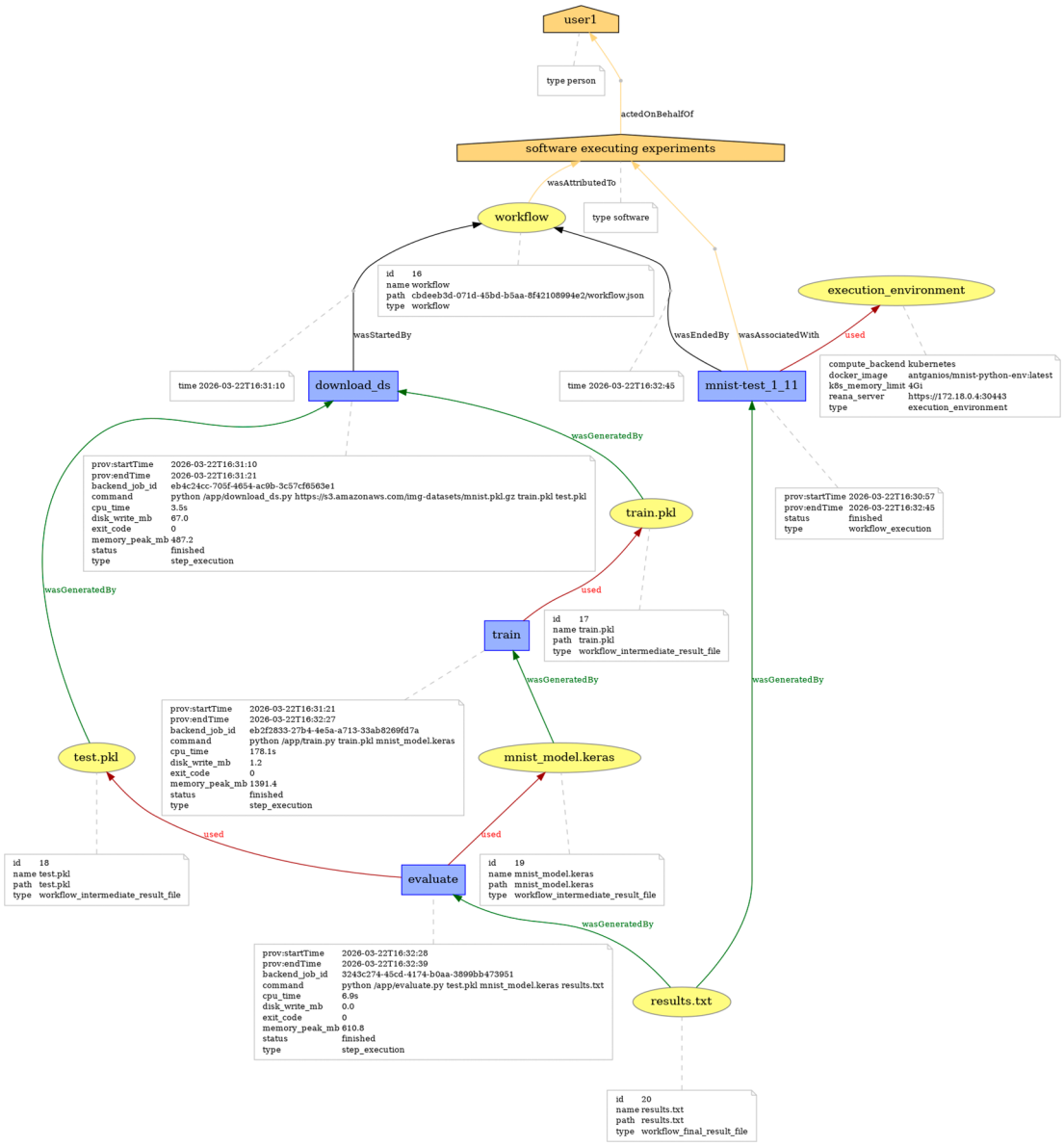

Figure 4. Provenance graph of a successful MNIST workflow execution.

To demonstrate ReProv's handling of execution failures, a second workflow was executed in which an additional validation step (`validate_model`) was inserted between model training and evaluation. This step intentionally fails and returns a non-zero exit code. As a consequence, the evaluation stage is never executed. Figure 5 shows the corresponding provenance graph.

Instead of generating provenance information only after successful completion, ReProv captures provenance incrementally during workflow execution. As a result, all activities and artifacts produced before the failure remain available in the provenance record. The failed step is explicitly represented in the graph and annotated with its execution status, error message, and exit code. This allows users to identify precisely where and why the workflow terminated.

In addition to lineage information, ReProv records execution metadata for each workflow step, including start and end timestamps, CPU time, memory consumption, and disk I/O usage. These measurements are collected automatically through lightweight wrappers inserted during workflow submission and become part of the provenance record. Together, these capabilities enable provenance to serve not only as a mechanism for reproducibility and traceability, but also as a practical tool for debugging, monitoring, and auditing workflow executions.

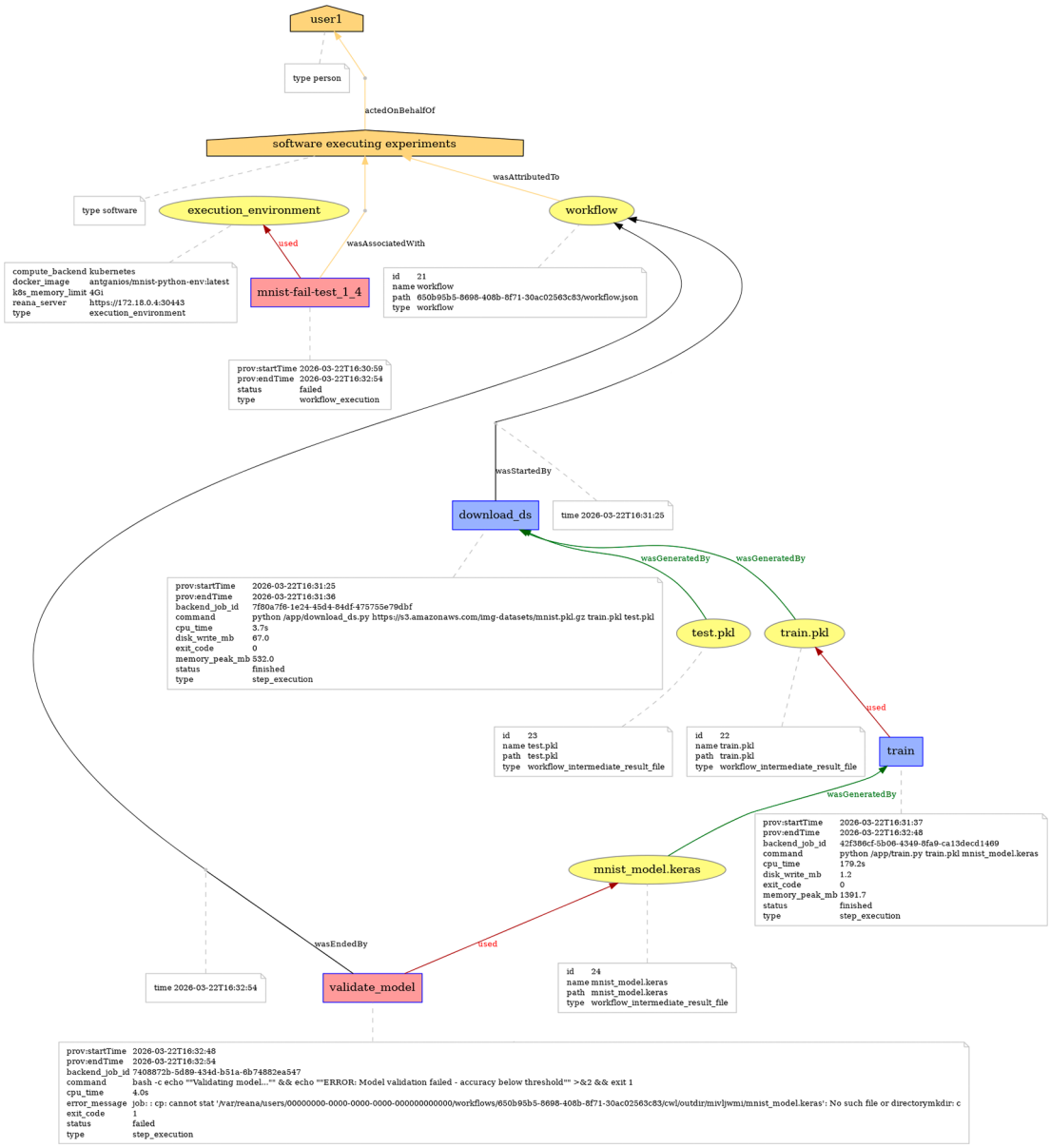

Figure 5. Provenance graph of a failed MNIST workflow execution showing failure-aware provenance capture.

Use Case 2: Larger-scale workflow: Next-day mediterranean heatwave prediction

To evaluate ReProv in a realistic scientific setting, we implemented a machine-learning workflow for next-day heatwave prediction over the Mediterranean region. Heatwaves are among the most impactful climate hazards in Europe, affecting public health, agriculture, and energy systems. Predicting their onset therefore provides a realistic and scientifically relevant use case for climate-data processing and machine-learning workflows. This workflow demonstrates how ReProv manages workflow execution, provenance capture, and artifact traceability throughout a complete data-processing and machine-learning pipeline.

The input dataset consists of ERA5 atmospheric reanalysis data[14] covering the Mediterranean basin during July and August for the period 1997–2025. The workflow is implemented as a six-step CWL pipeline:

- First, a *staging step* identifies the 29 yearly ERA5 data files required for processing and emits their paths.

- Second, a *preprocessing step* extracts daily atmospheric features from each yearly dataset, including near-surface daily-maximum temperature, daily-mean geopotential height at 500 hPa, specific humidity at 700 hPa, zonal and meridional wind components at 500 hPa. These variables capture atmospheric conditions commonly associated with heatwave development. The step is executed in parallel, producing one feature file per year.

- Third, a *merging step* combines the yearly outputs created in the previous step into a single chronologically ordered dataset spanning the full study period.

- Fourth, a *labeling step* generates the prediction target by identifying heatwave events according to a predefined temperature threshold and duration criterion. Specifically, a day is labeled as a heatwave event when the basin-mean daily maximum temperature exceeds the 90th percentile threshold computed from the 1997–2018 reference period for at least three consecutive days. The same threshold is then applied across the entire dataset.

- Fifth, a *training step* fits a gradient-boosted classification model[15] using data from 1997–2018 to predict whether the following day will be part of a heatwave event. In addition to the five original atmospheric variables, the workflow derives short-term historical features, such as lagged values and moving averages, to capture temporal dependencies. Model hyperparameters are selected through a systematic search procedure combined with time-series cross-validation, after which the best-performing model is retrained on the full training period and stored for later use.

- Finally, an evaluation step applies the trained model to the independent 2019–2025 period and computes predictive performance metrics.

The workflow produces a sequence of intermediate and final artifacts, including 29 yearly feature datasets, a merged climate dataset, a labeled dataset, a serialized machine-learning model, and an evaluation report.

Table 1 summarizes the key technical details of the Mediterranean heatwave prediction workflow used in the evaluation.

Table 1. Summary of the mediterranean heatwave prediction workflow.

Component	Description
Dataset	ERA5 reanalysis
Region	Mediterranean basin
Period	July–August, 1997–2025
Prediction task	Next-day heatwave occurrence
Raw predictors	Daily maximum temperature, 500 hPa geopotential height, 700 hPa specific humidity, 500 hPa zonal wind, 500 hPa meridional wind
Target definition	Temperature exceeds 90th percentile of 1997–2018 climatology for ≥3 consecutive days
Feature engineering	Same-day value, 1–3 day lags, 3-day trailing mean
Number of features	25
Model	Gradient Boosted Classifier
Hyperparameter search	Trees: {1500, 3000, 6000}; Depth: {6, 8}; Learning rate: {0.05, 0.1}
Validation	10-fold time-series cross-validation
Training period	1997–2018
Test period	2019–2025
Evaluation metrics	ROC-AUC, Precision, Recall, F1

To assess the runtime impact of provenance capture, the complete workflow was executed 10 times. ReProv recorded provenance information and resource-usage metadata for every workflow step and execution. Table 2 summarizes the average execution statistics across all runs, including wall-clock time, CPU time, peak memory consumption, disk usage, and the time spent recording provenance information.

Table 2. Execution statistics and provenance capture overhead for each step of the Mediterranean heatwave prediction workflow. Values are averaged over 10 workflow executions.

Step	Output Artifact	Wall-Clock Time (s)	CPU Time (s)	Provenance Time (s)	Peak Memory (MB)	Disk Usage
stage	slab_list.txt (29 paths)	21.0	0.01	0.000	8.5	1.2 KB
preprocess	era5_YYYY_07_08.daily.nc (scatter, 29 shards)	93.0	55.24	0.016	151.4	14.5 GB
merge	med_daily_summer.nc	5.3	2.81	0.022	118.4	1.8 GB
label	med_daily_labeled.nc	6.7	2.51	0.020	96.7	1.9 GB
train	heatwave_gbm.joblib	580.0	567.96	0.018	198.4	28.5 MB
evaluate	results.txt	6.2	5.42	0.006	185.2	4.1 KB

The results indicate that provenance capture introduces only a negligible overhead compared with the computational cost of workflow execution. Across all workflow stages, provenance-related operations required only a few milliseconds per step, while computational workloads ranged from seconds to several minutes. This behavior is expected because provenance capture primarily consists of lightweight metadata persistence and relationship recording operations. The experiment therefore demonstrates that comprehensive provenance tracking can be integrated into realistic scientific workflows without materially affecting execution performance.

Conclusions/Discussion

ReProv addresses an important gap in reproducible computational research by extending workflow execution with persistent registration, service-based invocation, structured provenance capture, and visualisation. Where systems such as REANA provide a strong foundation for containerised, reproducible execution, ReProv adds workflow lifecycle management and provenance-aware services around that layer, shifting the focus from running workflows in isolation to managing them as reusable, inspectable, and accountable research assets.

Its core contribution is the integration of WaaS capabilities with W3C PROV-based provenance tracking. Workflows can be registered, versioned, executed repeatedly, and monitored, while the provenance model records the relationships between workflow definitions, runs, step-level activities, artefacts, users, groups, and execution software. This lets users reconstruct how a result was produced and which agents were responsible, supporting reproducibility, auditability, debugging, and long-term reuse.

The use cases show that ReProv supports both simple and realistic workflows. The MNIST example captures provenance for successful and failed executions alike, treating failures as first-class events so that logs, exit codes, and intermediate artefacts remain available for diagnosis. The Mediterranean heatwave pipeline demonstrates a multi-step, data-intensive workflow, with measurements indicating that provenance capture adds only minimal overhead. These properties also advance FAIR-aligned research, supporting findability, accessibility, interoperability, and reusability through persistent registration, API and SDK access, community standards such as CWL and PROV-O, and systematic provenance capture.

Integration with the European AI-on-Demand Platform broadens this impact by linking workflows, datasets, and provenance to a wider AI ecosystem, which is especially relevant for trustworthy AI, where trust depends not only on model outputs but on the data, procedures, and environments that produced them.

Several limitations remain. ReProv currently targets CWL workflows executed through REANA; the relational provenance backend may need RDF-native serialisation for semantic-reasoning and linked-data use cases; current visualisations will need filtering or interactive exploration to scale to larger workflows; and deeper catalogue integration

alongside broader evaluation across scientific domains would further strengthen the evidence for scalability, usability, and generality.

Future work will therefore focus on extending ReProv in four main directions:

- Broader interoperability, including support for additional workflow systems, richer metadata schemas, and export formats such as PROV-N, PROV-JSON, RDF/Turtle, or RO-Crate.

- More complete integration with AIoD and other research catalogues, enabling workflows and their outputs to be discovered, cited, reused, and linked to external datasets and services.

- Improved provenance exploration, including interactive visualisation, query support, provenance comparison across executions, and automated identification of changed inputs, parameters, software versions, or outputs.

- Operational hardening, including larger-scale deployment, improved access-control policies, automated packaging of reproducibility bundles, and more extensive benchmarking of provenance overhead under different workflow sizes, execution granularities and infrastructures.

In summary, ReProv shows that provenance-aware Workflows-as-a-Service can be realised as a lightweight, practical extension to existing reproducible execution systems rather than as a separate infrastructure. By layering persistent workflow registration, service-oriented invocation, and automated PROV-O provenance capture on top of REANA's containerised CWL execution, and by exposing them through a REST API, a Python SDK, and graph-based visualisations, ReProv turns a workflow execution engine into a service-oriented research infrastructure for transparent, reusable, and accountable computational science. These capabilities are obtained at negligible runtime cost, and the captured lineage remains informative even for failed executions, recording where and why a run terminated. Integration with the European AI-on-Demand Platform further situates workflows, datasets, and provenance within a broader ecosystem of shared AI resources. Such properties are especially valuable for data-intensive and AI-driven research, where reproducibility depends not only on preserving final results but on capturing the complete computational context (inputs, intermediate artefacts, execution environment, and responsible agents) in which those results were produced.

Ethics and consent

Ethical approval and consent were not required.

AI and generative AI use declaration

During the preparation of this manuscript, the authors used ChatGPT (OpenAI, GPT-5.5) to assist with language editing and proofreading, specifically to review and refine English expression, grammar, and clarity while preserving the original meaning and scientific content. Claude Sonnet 4.6 by Anthropic was used to support the testing and auditing of the ReProv software. All substantive research content, including the study design, analysis, interpretation of results, and conclusions, is entirely the authors' own. The authors reviewed and verified all AI-assisted output and take full responsibility for the content of this work.

Data and software availability

Underlying data

This study uses two publicly available datasets:

- The first use case utilizes MNIST,[13] which is the standard handwritten-digit benchmark dataset and is retrieved at run time by the workflow's download step from its canonical distribution.

  Link: http://yann.lecun.com/exdb/mnist/

- The Mediterranean heatwave use case uses the ERA5 atmospheric reanalysis[14] produced by the Copernicus Climate Change Service and distributed through the Copernicus Climate Data Store (dataset reanalysis-era5-pressure-levels); geopotential, specific humidity, temperature, and the zonal and meridional wind components at the configured pressure levels over the Mediterranean basin (July–August, 1997–2025) were retrieved via the CDS API, and the download specification used is provided with the software.

  Link: https://cds.climate.copernicus.eu/datasets/reanalysis-era5-single-levels

Software availability

Source code available from: https://github.com/id-is/ReProv-API.[16]

Archived code at the time of publication: doi:10.1145/2939672.2939785

License: Apache License 2.0. https://www.apache.org/licenses/LICENSE-2.0

References

1. Klampanos IA, Magnoni F, Casarotti E, *et al.*: **DARE: A Reflective Platform Designed to Enable Agile Data-Driven Research on the Cloud.** *eScience.* 2019; 578–585.
Publisher Full Text
2. Klampanos IA, Themeli C, Spinuso A, *et al.*: **DARE Platform: a Developer-Friendly and Self-Optimising Workflows-as-a-Service Framework for e-Science on the Cloud.** *J. Open Source Softw.* 2020; **5**(54): 2664.
Publisher Full Text
3. Belhajjame K, Cheney J, Corsar D, *et al.*: **PROV-O: The PROV Ontology.** *W3C Recommendation.* 2013.
Reference Source
4. Markaki OI, Papapostolou A, Mouzakitis S, *et al.*: **Encouraging AI Adoption by SMEs: Opportunities and Contributions by the ICT49 Project Cluster.** *IISA.* 2023; 1–8.
Publisher Full Text
5. Troumpoukis A, Klampanos IA, Pantazi D-A, *et al.*: **European AI and EO convergence via a novel community-driven framework for data-intensive innovation.** *Futur. Gener. Comput. Syst.* 2024; **160**: 505–521.
Publisher Full Text
6. Karakolis E, Pelekis S, Mouzakitis S, *et al.*: **Artificial intelligence for next generation energy services across Europe–the I-Nergy project.** *International Conferences e-Society 2022 and Mobile Learning.* 2022; 61–68.
Reference Source
7. Troumpoukis A, Marinakis A, Gizelis CA, *et al.*: **DeployAI: AI-on-Demand Platform and Marketplace for Industrial Applications within Europe.** *AIAI Workshops.* 2025; **1**: 115–127.
Publisher Full Text
8. Suter F, Coleman T, Altintas I, *et al.*: **Rafael Ferreira da Silva: A terminology for scientific workflow systems.** *Futur. Gener. Comput. Syst.* 2026; **174**: 107974.
Publisher Full Text
9. Amstutz P, Crusoe MR, Tijanić N, *et al.*: **Stian Soiland-Reyes, Luka Stojanovic.** *Common Workflow Language.* 2016; **v1.0**.
Publisher Full Text
10. Abrishami S, Zandi FM, Nourbakhsh A: **Workflow as a Service Broker in Cloud Environment: A Systematic Mapping Study.** *CoRR abs/2501.12672.* 2025.
Publisher Full Text
11. Šimko T, Heinrich L, Hirvonsalo H, *et al.*: **REANA: A System for Reusable Research Data Analyses.** *EPJ web of conferences.* 2019; **214**.
Publisher Full Text
12. Wilkinson MD, Dumontier M, Aalbersberg IJJ, *et al.*: **The FAIR Guiding Principles for scientific data management and stewardship.** *Sci. Data.* 2016; **3**(1): 160018.
Publisher Full Text
13. LeCun Y, Cortes C, Christopher JC: *Burges: The MNIST Database of Handwritten Digits.* AT&T Labs and NYU.
Reference Source
14. Hersbach H, Bell B, Berrisford P, *et al.*: **The ERA5 global reanalysis.** *Q. J. R. Meteorol. Soc.* 2020; **146**(730): 1999–2049.
Publisher Full Text
15. Chen T, Guestrin C: **XGBoost: A Scalable Tree Boosting System.** *KDD.* 2016; 785–794.
Publisher Full Text
16. Troumpoukis A, Gkanios A, Klampanos I: **ReProv: Extending REANA with WaaS capabilities and data provenance on the European AI-on-Demand Platform - Software.** *Zenodo.* 2026.
Publisher Full Text