OGC® DOCUMENT: 25-043

External identifier of this OGC® document: http://www.opengis.net/doc/PER/OSPD-2025

Open Geospatial Consortium

OPEN SCIENCE PERSISTENT DEMONSTRATOR (OSPD) 2025 REPORT

ENGINEERING REPORT

PUBLISHED

Submission Date: 2026-02-12
Approval Date: 2026-04-29
Publication Date: 2026-08-11
Editor: Pedro Goncalves, Rob Atkinson, Sina Taghavikish

Note

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

CONTENTS

I OVERVIEW

The OSPD 2025 initiative addresses a long-standing challenge in the scientific community: enabling computational workflows to be **reproduced, validated, and reused** across different data and computing infrastructures. In many research contexts, workflows remain tightly coupled to specific platforms, lack consistent metadata, and are difficult to share or reproduce beyond their original execution environment.

OSPD responds to this challenge by demonstrating a standards-based approach to workflow description and execution. Through the coordinated activities of its four Work Packages, the initiative establishes common metadata schemas, semantic models, and reusable specification components that together support transparent and interoperable workflow execution.

Each workflow documented within the demonstrator contributes to a federated collection of reusable scientific methods, aligned with FAIR data principles and grounded in the OGC Open Science Building Blocks framework. The initiative combines technical development with multi-organization collaboration, showing how common standards and open technologies can support open science at scale.

II EXECUTIVE SUMMARY

The **Open Science Persistent Demonstrator (OSPD) 2025** is an international collaboration led by the **Open Geospatial Consortium (OGC)** with the support of the **European Space Agency (ESA)**, **NASA**, the **EU FOCAL project**, and **I-GUIDE**. Building on the outcomes of OSPD Phase 1 (OGC 24-022), the initiative advances open and reproducible science by demonstrating interoperable approaches for workflow description, sharing, and execution across distributed computing environments.

The second phase of OSPD focuses on operationalizing the Open Science Building Blocks introduced in 2024. It brings together scientific institutions, platform operators, and technology developers to demonstrate that scientific workflows can be consistently described, executed, and reused using standardized metadata, shared ontologies, and containerized execution environments.

The work was organised into four complementary Work Packages (WPs):

- **WP 1 – Workflow Discovery and Evaluation:** identified representative scientific workflows and harmonised their metadata using STAC-based schemas and parameter documentation;
- **WP 2 – Open Science Ontology:** developed a minimal semantic model linking workflows, parameters, datasets, and execution platforms;
- **WP 3 – Workflow Reuse Experiments:** validated workflow portability and reproducibility across multiple infrastructures; and

- **WP 4 – OGC Workflow Building Blocks:** formalised schema-driven specification components supporting interoperability across open-science systems.

Taken together, these activities demonstrate the feasibility of combining ontologies, reusable specification components, and workflow reuse experiments into a coherent interoperability framework. OSPD 2025 delivers a set of validated technical assets, including schemas, ontology examples, and workflow records, that provide a concrete foundation for future OGC Best Practice documentation on open science workflow interoperability.

III KEYWORDS

The following are keywords to be used by search engines and document catalogues.

Open Science, FAIR Principles, OGC Building Blocks, Open Science Ontology, Cross-Platform Workflow Reuse

IV CONTRIBUTORS

All questions regarding this document should be directed to the editor or the contributors:

NAME	ORGANIZATION	ROLE
Pedro Goncalves	Terradue Srl	Editor
Rob Atkinson	OGC	Editor
Sina Taghavikish	OGC	Editor
Christophe TRIQUET	CS Group	Contributor
Tyna Dolezalova	EOX	Contributor
Stephan Meissl	EOX	Contributor
Gérald Fenoy	GeoLabs	Contributor
Stelios Contarinis	HARTIS Integrated Nautical Services	Contributor

NAME	ORGANIZATION	ROLE
Furqan Baig	I-GUIDE	Contributor
Nicholas Car	KurrawongAI	Contributor
Tom De Nies	KurrawongAI	Contributor
Deyan Samardzhiev	Lampata	Contributor
Krasen Samardzhiev	Lampata	Contributor
Nenad Radosevic	RMIT University	Contributor
Cameron Sajedi	Starling Foundries	Contributor
Alex Hayward	Telespazio VEGA UK Ltd	Contributor
Garin Smith	Telespazio VEGA UK Ltd	Contributor

V FUTURE OUTLOOK

OSPD 2025 establishes a technical and conceptual foundation that extends beyond the duration of the demonstrator itself. The validated ontology, workflow descriptions, and Building Blocks produced during this phase provide an evidence base for future refinement and standardization activities within the OGC.

In the near term, the results of OSPD 2025 are expected to inform the development of an **OGC Best Practice** on open science workflow interoperability. Further validation, profiling, and constraint definition may be undertaken in subsequent initiatives to strengthen consistency and automation across platforms.

In the longer term, the approaches demonstrated in OSPD 2025 can be applied to a broad range of scientific and operational domains, including climate resilience, environmental monitoring, and geospatial analytics. The long-term vision is a persistent, community-driven ecosystem in which research workflows, data services, and computational resources are connected through shared standards, enabling transparent and reproducible science across domains and organizations.

VI VALUE PROPOSITION

The **Open Science Persistent Demonstrator** delivers clear value to the scientific and technical communities by providing practical evidence of how interoperability can be achieved through open standards.

Key benefits include:

- **Interoperability:** workflows and associated data can be described and executed across heterogeneous systems with reduced platform-specific adaptation;
- **Transparency:** workflow configuration, execution context, and results are expressed using standardized, machine-readable representations;
- **Efficiency:** shared schemas, ontologies, and specification components reduce duplication and simplify the publication and reuse of scientific methods;
- **Scalability:** containerized workflows and OGC API standards support execution from exploratory research environments to operational services; and
- **Collaboration:** a common technical framework strengthens cooperation among agencies, research institutions, and industry partners.

By aligning diverse participants around a shared interoperability approach, OSPD 2025 demonstrates how open standards can make scientific practice more accessible, reliable, and reusable across the global geospatial community.

1

INTRODUCTION

1 INTRODUCTION

The **Open Science Persistent Demonstrator (OSPD)** is an inter-agency initiative led by the **Open Geospatial Consortium (OGC)** in collaboration with the **European Space Agency (ESA)**, **NASA**, the **EU FOCAL project**, and **I-GUIDE**. The initiative aims to advance open, reproducible, and interoperable Earth and Environmental Science by developing and validating frameworks that enable transparent research workflows across distributed platforms and data environments.

The first phase of the demonstrator, completed in 2024 (OGC 24-022), established the conceptual foundation for representing scientific workflows and introduced the notion of reusable “building blocks” for open-science processes. The 2025 phase extends this work into a persistent, collaborative demonstrator that matures technical solutions, validates them through operational implementations, and consolidates best practices for community adoption.

1.1. Aims

The OSPD 2025 Demonstrator focuses on the practical implementation of concepts introduced in Phase 1 to demonstrate interoperability, reproducibility, and transparency of scientific workflows. It is organized into four interconnected Work Packages (WPs).

- **WP 1 – Workflow Discovery and Evaluation**: identify representative workflows, describe their structure and metadata (inputs, outputs, parameters) using **STAC** schemas and auxiliary CSV files, and establish persistent references to reusable workflow assets.
- **WP 2 – Open Science Ontology**: develop a minimum common ontology aligned with **PROV**, **CWL**, and **EarthCODE** concepts to describe workflows, parameters, data, and execution environments.
- **WP 3 – Workflow Reuse Experiments**: test the portability and reproducibility of workflows across multiple cloud and HPC platforms operated by participating organizations.
- **WP 4 – OGC Workflow Building Blocks**: define and validate modular schema components that define profiles of schemas for defining **APIs**, **metadata records** and **provenance** that act as interoperability contracts for workflow exchange and execution.

The key enablers for enhanced interoperability provided by the Building Blocks approach are:

1. Easy identification of reuse of common schemas and semantics
2. Easy extensibility of metadata to handle complex aspects such as workflow descriptions and provenance

3. Provision of profiling mechanisms to apply generic standards to specific application needs
4. Mapping schemas to ontologies using JSON-LD
5. Combination of simple testable JSON-LD mappings of sub-schemas into the complex JSON-LD structures needed to work with rich metadata models
6. Extensive automated testing of examples at both component and complex schema levels
7. Publication of interoperability resources in a standardized form using a flexible knowledge graph and Linked Data compatible artifacts.

The OSPD phase 2 project introduces these capabilities to the EO community and performs testing, refinement and validation of both an extensive suite of reusable Building Blocks and the methodology and tooling support required.

Together, these work packages establish a complete activity chain: **discover → describe → reuse → standardize**.

1.2. Objectives

The specific objectives of the OSPD 2025 Demonstrator are to:

1. Demonstrate **cross-platform reproducibility** of Earth-science workflows using standardized metadata and open interfaces;
2. Advance the **Open Science Ontology** to support semantic alignment between workflow, data, and platform descriptions;
3. Validate the **OGC Open Science Building Blocks** through capture of information required to support real implementations in partner platforms such as GEP, VEDA, Polar TEP, and EarthCODE;
4. Capture **best practices** for workflow packaging, provenance, and FAIR data compliance; and
5. Provide **recommendations** toward future OGC standards or best-practice documents supporting open-science interoperability.

2

WORK PACKAGES AND RESULTS

2 WORK PACKAGES AND RESULTS

This section describes the activities and results of the four Work Packages (WPs) that compose the **Open Science Persistent Demonstrator (OSPD) 2025** initiative. Each Work Package addresses a distinct aspect of the open-science workflow lifecycle, from discovery and description, to reuse and standardization, and contributes to the collective objective of enabling reproducible and interoperable research across platforms.

The Work Packages are closely interlinked.

- **WP 1: Workflow Discovery and Evaluation** defines the approach to identify, document, and structure representative scientific workflows, including their inputs, outputs, and parameters.
- **WP 2: Open Science Ontology** provides the semantic framework required to describe workflows, parameters, datasets, and platforms in a machine-interpretable form.
- **WP 3: Workflow Reuse Experiments** validates these descriptions by executing the selected workflows across multiple cloud and high-performance computing platforms.
- **WP 4: OGC Workflow Building Blocks** consolidates the results into reusable, standards-ready schema components for long-term interoperability within the OGC ecosystem.

Together, these Work Packages demonstrate the technical and organizational feasibility of establishing a **persistent open-science environment** built on shared ontologies, standardized metadata, and reproducible workflow execution.

Each subsection below presents, for its corresponding Work Package:

- the objectives and main activities performed,
- methods and tools used,
- achieved results and findings, and
- identified challenges or lessons learned.

2.1. WP 1 – Workflow Discovery and Evaluation

2.1.1. Objectives

WP 1 focuses on identifying, documenting, and evaluating **representative workflows** that demonstrate the open-science principles of reproducibility, transparency, and interoperability.

The purpose is to collect a diverse set of scientific workflows that can serve as **test cases** for the OSPD-2025 demonstrator and provide the metadata basis for reuse and cross-platform execution in later work packages.

Each workflow is described using common metadata elements.

- **Input and output datasets**, described through STAC records or, when needed, CSV parameter tables.
- **Workflow identifiers and artefacts**, referencing CWL definitions, Jupyter notebooks, or OGC Records.
- **Parameters**, their names, ranges, and semantic meaning.
- **Execution examples**, linking to EarthCODE or other registries.

The objective is to ensure each workflow can be referenced, discovered, and reused as part of a **persistent demonstrator**.

2.1.2. Activities

Several “champion” workflows were proposed, documented, and refined by participating partners. Each workflow entry includes a short description, input and output definitions, processing steps, and references to associated repositories.

Current workflows under WP 1 as as follows.

- **Water Bodies Detection** – proposed by **Telespazio UK**: Uses the NDWI and Otsu threshold on Sentinel-2 COG STAC items to identify surface-water areas. Implemented in CWL as two workflows (`detect_water_body`, `water_bodies`) producing STAC catalogs of GeoTIFF results. Repositories and example records are published through the **EarthCODE** Open Science Catalogue (OSC, staging deployment) (`https://osc-staging.earthcode.eox.at/workflows/waterbodies/record`).
- **PolarWarp** – proposed by **EOX**: Geolocates and aligns Sentinel-1 SAR images in polar regions using sea-ice drift and model data (TOPAZ4, ICON, tide models). Applies thin-plate-spline warping for drift correction and temporal alignment. Inputs include SAR imagery and drift vectors; outputs are drift-corrected composites accessible through the **EarthCODE** Open Science Catalogue (OSC, staging deployment) (`https://osc-staging.earthcode.eox.at/products/polarwarp/collection`).
- **POLARIS** – proposed by **EOX**: Computes the Polar Operational Limit Assessment Risk Indexing System for ships in ice-covered waters using WMO ice-chart data. The workflow integrates SIGRID-3 ice charts, ship class, and ice-risk tables to generate a Risk Index map (RIO). Output examples and metadata records are provided through **EarthCODE** Open Science Catalogue (OSC, staging deployment) (`https://osc-staging.earthcode.eox.at/products/polaris/collection`).
- **Mangrove Detection and Biomass Estimation** – proposed by **Starling Foundries**: Estimates above-ground biomass and carbon stocks from Sentinel-2 imagery. Uses vegetation

indices (NDVI, NDWI, SAVI) and validated allometric models ($R^2 = 0.72$) to derive biomass and CO_2 equivalence. Implemented as a Jupyter notebook (`mangrove_workflow.ipynb`) querying AWS STAC catalogs; potential for LiDAR and InSAR integration is noted.

- **Coastal Vulnerability Index (CVI)** – proposed by **HARTIS Integrated Nautical Services**: Implements the USGS CVI method using four open parameters: land cover, slope, erosion, and elevation. Uses ESA WorldCover, Copernicus DEM, and Coastal Hazard Wheel datasets to produce transect-based GeoJSON outputs visualized in Folium. Configuration files (`cvi_scoring_simple.json`) define ranking thresholds; data inputs are catalogued in a STAC JSON (`cvi_stac_catalog.json`).

These workflows cover a broad range of environmental applications—hydrology, cryosphere, marine risk, ecosystem monitoring, and coastal hazards—and collectively represent the core demonstrator use cases for OSPD 2025.

2.1.3. Results

Each workflow entry includes structured descriptions of inputs, parameters, and outputs, along with repository references and draft STAC schemas describing example datasets. These materials provided the primary input for ontology alignment activities under WP 2 and for cross-platform workflow reuse experiments conducted in WP 3.

2.1.4. Challenges and Next Steps

WP 1 highlighted several recurring challenges related to workflow description and discovery.

In particular, significant variation was observed in how workflows describe their inputs, outputs, and configuration parameters. While STAC provides a common foundation for dataset description, additional conventions or supporting artefacts were often required to document workflow-specific parameters and processing assumptions.

These observations directly informed the development of the Open Science Ontology in WP 2 and the identification of reusable specification patterns addressed through the OGC Open Science Building Blocks in WP 4.

2.2. WP 2 – Open Science Ontology

2.2.1. Objectives

WP 2 aims to define a **minimum common ontology** to represent workflows, parameters, datasets, and execution environments in a machine-interpretable manner. The ontology provides the semantic foundation linking workflow metadata (from WP 1) with platform capabilities and provenance information, enabling workflows to be described, shared, and executed consistently across participating systems.

The ontology design aligns with established standards and vocabularies, including:

- **DCAT** (Data Catalog Vocabulary) for generic description of resources,
- **PROV-O** for provenance and activity modeling,
- **Common Workflow Language (CWL)** for process definition, and
- **EarthCODE** concepts for workflow and experiment representation.

Together, these foundations form a minimal, extensible model connecting workflows ("what is done"), parameters ("how it is configured"), and platforms ("where it runs").

Additional ontologies required to address more specific concepts were reviewed and, where necessary, incorporated as modular extensions.

2.2.2. Activities

The work focused on defining the ontology's core structure and selecting reference models to ensure interoperability with existing community standards.

Given the use of STAC and the associated OGC API – Records and Features models, existing Building Blocks linking these to DCAT, GeoJSON, and GeoSPARQL provided a suitable baseline.

This baseline is compatible with a wide range of external communities such as EU Data Spaces.

PROV-O was adopted as the primary model for describing provenance, building on existing tooling and prior work aligning it with CWL schemas. However, it was observed that CWL, while precise, is too fine-grained for general ontology-driven workflow description due to the complexity of its schema structures.

As a result, an **intermediate-complexity** model was identified as necessary to describe workflows at a level suitable for discovery, reuse, and execution comparison. The **wf4ever** ontologies (**wfdesc.owl** and **wfprov.owl**) were therefore selected and evaluated for this purpose.

NOTE: wf4ever is not currently published as a governed standard by a recognized international standards organization. Nevertheless, the evaluation indicated that, if further validated, the model could be republished under a stable namespace and governance framework. The Building Blocks approach supports such refactoring, enabling controlled migration of namespaces and regression testing from a single authoritative source.

Elements from the **wf4ever** ontologies and the **Common Core Ontologies (CCO)** were reviewed and reused where appropriate to establish semantic alignment with existing OGC and FAIR data initiatives.

The resulting conceptual model includes the following workflow-specific components:

- **Workflow** – an **Entity** describing the computational process;
- **Parameter Description** – an **Entity** capturing configurable variables;
- **Workflow Execution (Experiment)** – an **Activity** representing a specific execution instance;
- **Input and Output Datasets** – **Entities** providing or resulting from execution; and
- **Platform and Data Access Services** – **Agents** responsible for execution and data exposure.

The ontology adopts a **black-box execution model**, in which workflows are described through their inputs, outputs, and provenance, without requiring disclosure of internal logic. Data inputs and outputs are described using **STAC** schemas, extended where necessary to support domain-specific parameters and metadata.

This abstract model can be augmented incrementally, allowing additional detail to be introduced through modular extensions and profiles as required by specific application domains.

2.2.3. Results (to date)

WP 2 delivered an initial ontology framework connecting workflow descriptions, parameters, datasets, and execution activities across multiple platforms.

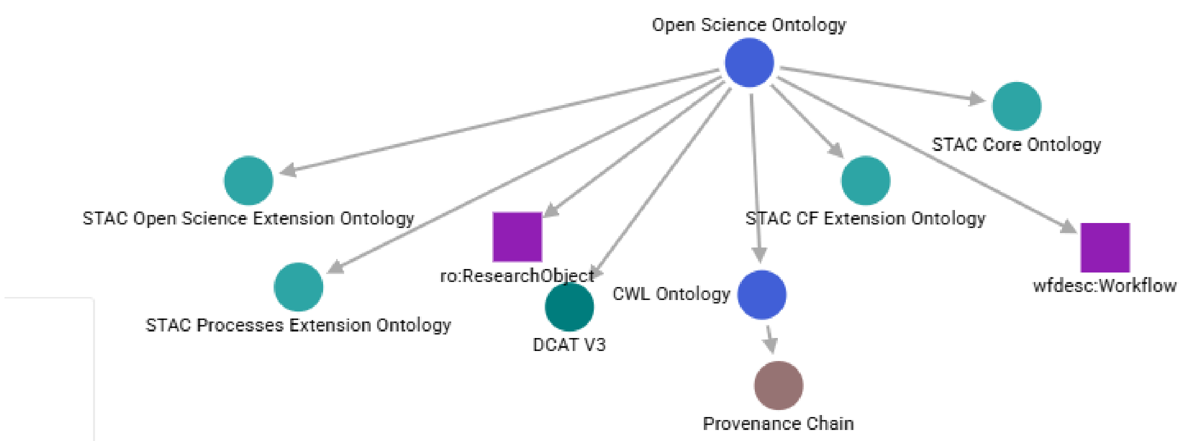


Figure 1

Preliminary mappings demonstrate that workflow metadata from platforms such as GEP and EarthCODE can be harmonized using PROV-O relationships combined with CWL and EarthCODE concepts. A first **JSON-LD** representation was implemented and tested for integration with the OGC Open Science Building Blocks and STAC-based workflow descriptors.

The evaluation confirmed the relevance of the **wf4ever** model for describing workflows and provenance at an intermediate level of detail, with **cwlprov** providing additional execution-specific information where required.

The use of JSON-LD for semantically annotating schema-based Building Blocks required the introduction of schema-specific ontologies. While these ontologies focus on structural elements, they were successfully used to support metadata generation and validation activities, including experiments with AI-assisted metadata formulation conducted under WP 4.

Metadata profiles required to describe workflows were aligned through a common mapping between OGC API – Records, STAC, and DCAT.

This alignment was validated through the publication of workflow, experiment, and product records in EarthCODE-compatible catalogues using existing OGC API – Records implementations, without requiring platform-specific modifications.

Overall, participants agreed that the ontology framework provides a necessary foundation for interoperable workflow description. At the same time, the work identified several areas requiring further investigation, including:

1. modeling interactions between workflows and data catalogues;
2. exploring automation of workflow translation between execution environments;
3. supporting AI-assisted workflow adaptation across platforms.

2.2.4. Challenges and Next Steps

WP 2 confirmed that a minimal, modular ontology is both feasible and appropriate for supporting open-science workflow interoperability.

The black-box execution model proved effective for enabling workflow discovery and reuse without imposing constraints on internal implementation. At the same time, the experiments highlighted the limits of current provenance and schema mappings when attempting to capture execution context and platform-specific behavior in a consistent manner.

Further work is required to formalize profile constraints, extend validation rules, and strengthen alignment with evolving OGC components, particularly the Workflow Building Blocks and related STAC extensions.

2.3. WP 3 – Workflow Reuse Experiments

2.3.1. Objectives

WP 3 aims to validate the interoperability and reproducibility of scientific workflows across multiple computing platforms. The objective is to demonstrate that workflows described with standardized metadata, ontology elements, and reusable components can be executed in different environments while maintaining consistency in configuration, provenance, and results.

This Work Package provides the practical validation layer of OSPD 2025, linking the workflow discovery and semantic modelling activities of WP 1 and WP 2 with the specification-oriented work carried out under WP 4.

2.3.2. Activities

Workflow reuse experiments in WP 3 follow a collaborative approach involving three complementary roles:

- **Scientists**, who provide workflow descriptions, parameter settings, and representative datasets that reflect real scientific use cases;
- **Building Block Developers**, who translate these descriptions into interoperable schema definitions — including **workflow**, **experiment**, and **product** specifications — ensuring they comply with the OGC Open Science Building Blocks; and
- **Platform Operators and Developers**, who deploy and execute the workflows on their respective infrastructures and assess execution behavior and results.

Each workflow selected for WP 3 follows a structured testing approach:

- description of inputs, outputs, and parameters using metadata structures derived from WP 1 and aligned with the ontology defined in WP 2;
- preparation of execution environments using containerized workflows or CWL-based Application Packages;
- execution of workflows through platform-specific services, including OGC API – Processes where available;
- capture of execution metadata and provenance information, with a focus on inputs, parameters, and produced outputs; and
- validation of results against schema definitions and expected structural constraints.

This approach supports a transition from isolated, platform-specific workflow executions towards interoperable and reusable workflow descriptions.

2.3.3. Results

By late 2025, several workflow reuse experiments have been carried out across participating platforms, including GEP, EarthCODE, and other OSPD environments. These experiments demonstrate that workflows described using shared metadata and reusable components can be prepared and executed on multiple infrastructures with limited adaptation. In particular, workflows and associated metadata were reused across EarthCODE-based environments and other OSPD platforms without requiring changes to the underlying platform implementations.

The reuse activities also provided practical validation of the Open Science Building Blocks, confirming their applicability while revealing areas requiring further refinement. In particular, workflow descriptions, parameter handling, and execution interfaces could be applied consistently, whereas the capture and interpretation of provenance information varied across platforms.

Detailed examples of workflow reuse experiments and validation results are provided in the annexes to this report.

2.3.4. Challenges and Next Steps

The workflow reuse experiments confirm the feasibility of using common descriptions and interfaces to support interoperability across heterogeneous platforms. They also highlight several recurring challenges that are relevant for future refinement.

A key observation concerns provenance capture because, while existing provenance models provide a useful foundation, the experiments showed that execution context, runtime environment, and platform-specific behavior are not yet consistently captured in a way that fully supports traceability and reproducibility across platforms.

Another observation relates to the flexibility of current schema definitions: while permissive schemas support early experimentation, they can also lead to divergent interpretations during validation. This indicates the need for clearer constraints, profiles, or conventions to support more robust and predictable interoperability.

Overall, WP 3 demonstrates the value of the Open Science Persistent Demonstrator as a practical testbed for open science interoperability. The findings from the reuse experiments provide concrete technical feedback that informs subsequent refinement of workflow descriptions, provenance approaches, and reusable specification components, as further documented in the annexes and in the OSPD 2025 Building Blocks Report.

2.4. WP 4 – OGC Workflow Building Blocks

2.4.1. Objectives

WP 4 develops and validates a set of **OGC Workflow Building Blocks**, modular, schema-driven components that define how workflows and experiments can be described, shared, and reused in an interoperable manner. These Building Blocks act as **interoperability contracts**, specifying the structure of APIs, schemas, and metadata elements used to represent scientific workflows across multiple systems. Their purpose is to consolidate the descriptive and execution models produced in WP 1–WP 3 into reusable technical assets aligned with broader OGC standards.

2.4.2. Activities

The Open Science Building Blocks developed under WP 4 are maintained in the public OGC Incubator registry at **https://ogcincubator.github.io/bblocks-openscience/**. The registry provides machine-readable schemas, example encodings, and accompanying documentation.

Each Building Block defines a well-scoped specification component that can be combined with others to create complete workflow descriptions. The Building Blocks developed in OSPD 2025 focus on the following areas:

- **Application** – describing the logical structure and intent of a scientific workflow or analytical process;
- **ApplicationPackage** – describing how an application is implemented and deployed, typically using containerized environments or CWL definitions;
- **Parameter** – describing input, output, and configuration variables in a standardized, machine-readable form; and
- **Platform** – describing the computational environment or services on which workflows are executed.

The Building Blocks follow a **schema-driven** approach, reusing existing OGC and external standards wherever possible. Semantic descriptions are inferred and refined based on validated workflow examples derived from reuse experiments conducted under WP 3.

Validated examples were used iteratively to refine schema structures and populate the registry. Tutorials and worked examples illustrate how Building Blocks can be composed and tested within the shared environment.

2.4.3. Results

By October 2025, initial Building Blocks for **Application**, **ApplicationPackage**, **Parameter**, and **Platform** were drafted and published in the OGC Incubator registry. These Building Blocks demonstrate how workflow descriptions from participating platforms can be aligned to a common schema structure.

A consistent JSON Schema pattern was established, enabling validation of metadata examples derived from workflows documented in WP 1 and executed across platforms in WP 3. The Building Blocks were applied in practice to support workflow description, validation, and reuse activities within OSPD 2025.

The detailed definition, structure, and validation of the Building Blocks are documented in the **Open Science Persistent Demonstrator (OSPD) 2025 Building Blocks Report (OGC 25-044)**.

2.4.4. Challenges and Next Steps

WP 4 confirmed that the Building Blocks approach provides a practical and scalable mechanism for formalizing reusable workflow specification components.

At the same time, the work highlighted the need to carefully balance flexibility and validation. While permissive schemas support early experimentation across heterogeneous platforms, clearer constraints and profiles will be required to support more predictable interoperability. Validation activities also highlighted differences in schema support and validation behavior across platform implementations, reinforcing the need for incremental adoption and profiling of Building Blocks.

The experience gained under WP 4 provides a concrete foundation for future OGC Best Practice and standardization activities related to open-science workflow interoperability.

3

OPEN SCIENCE DEMONSTRATORS OF OSPD 2025

3 OPEN SCIENCE DEMONSTRATORS OF OSPD 2025

The Open Science Demonstrator 2025 is a collaborative initiative led by the Open Geospatial Consortium (OGC), co-sponsored by NASA, the European Space Agency (ESA), the European Union through the FOCAL project, and the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE). This pilot aims to advance the principles of open science by developing and validating a framework for the standardized description, sharing, and reuse of scientific workflows. The Open Science Persistent Demonstrator (OSPD) is a long-term inter-agency initiative aiming to enable and communicate reproducible Earth Science across global communities of users and amplify inter-agency Earth Observation mission data, tools, and infrastructures.

This section provides an overview of the platforms, as well as access information concerning each of these platforms. The platforms were at varying levels of maturity at the commencement of the pilot; however, nearly all are capable of facilitating access to the various workflows tested within the OSPD 2025 pilot. These persistent demonstrators are available for a year (till September 2026). But access and availability will be the responsibility of the participating entities (i.e., CS Group, EOX, Hartis, I-GUIDE, GeoLabs, Terradue srl, and Telespazio UK).

3.1. Objectives

The primary objective of a Demonstrator is to address the growing need for transparency, reproducibility, and interoperability in scientific research. By enabling workflows to be described in a machine-readable, modular, and standards-compliant manner, the initiative seeks to:

- Facilitate the reproducibility of scientific results across platforms and institutions;
- Promote the reuse of workflows in new contexts, reducing duplication of effort;
- Support interoperability through the adoption of open standards such as those developed by OGC; and
- Enhance transparency by documenting data sources, processing steps, and assumptions in a structured format.

3.2. CS Group Platform

3.2.1. Platform description

CS Group is involved as platform operator and delivers a dynamic and evolving research environment and an orchestrator that fulfills several key requirements: reproducibility, efficiency, interoperability, transparency, performance and scalability, monitoring and diagnostics, open standards and sharing.

AntFlow is a lightweight, standards-based orchestration platform developed by CS GROUP to streamline scientific workflow deployment and execution.

AntFlow follows a Git-first approach: workflows are packaged as OGC Earth Observation Application Packages (EOAP) stored in GitLab and are discovered automatically (based on repository metadata) and parsed at runtime. From these EOAP descriptors (graph CWL), AntFlow generates OGC API Processes process descriptions and submission schemas on-the-fly, so processes become available without custom service development and evolve as the underlying Git repositories are updated.

The platform is designed for standards compliance and interoperability, leveraging CWL, EOAP best practices, OGC API Processes, and STAC conventions for describing inputs/outputs. This architecture ensures reproducibility through version-controlled workflow definitions and efficient execution by optimizing resource allocation across heterogeneous computing environments. AntFlow supports deployment on hybrid platforms including cloud infrastructures, High-Performance Computing (HPC) systems, and local machines, enabling workflows to be executed where computational resources are most appropriate and cost-effective.

3.2.2. Technology Stack

AntFlow is a new development built on open standards including:

- CWL (Common Workflow Language)
- OGC Earth Observation Application Packages
- OGC API Processes
- STAC (SpatioTemporal Asset Catalogue)
- Git (de facto standard for version control)

AntFlow exposes the available processes through a standard OGC API Processes interface (e.g., process listing and execution endpoints). This API can be consumed by any OGC API Processes compliant client.

Jobs can be executed on heterogeneous backends (cloud/Kubernetes, HPC, or even local machine) through a distributed client-server architecture. The platform leverages open-source CWL runners including cwltool (the standard CWL implementation), Calrissian (for Cloud infrastructures), and Toil (work in progress) as execution backends. Runner client components automatically register with the server, advertise their computational capabilities (such as GPU availability, memory capacity), and claim matching jobs from the server.

In addition, AntFlow provides an ergonomic web interface built on top of the same standardized API. The UI generates process launch forms on-the-fly and adapts controls to the declared input schemas, including datetime picker, numerical range and EOAP standardized bounding box inputs (interactive map widget).

Access control is aligned with Git hosting permissions (users can only see and execute processes from repositories they can access). Process versions can be pinned to Git references (branch, tag, or commit), and a single repository can host multiple EOAPs that are discovered automatically.

3.2.3. Prerequisites

Before accessing AntFlow, ensure you have:

- GitLab Account: Valid credentials on https://www.gitlab.com
- Project Access: Appropriate permissions on GitLab projects you wish to use

3.2.4. How to Access AntFlow

3.2.4.1. OSPD Instance

The OSPD dedicated instance of AntFlow is available at:

- AntFlow UI: https://antflow.ospd.csgroup.space/ui
- Documentation: https://antflow.ospd.csgroup.space/ui/#/docs

Note: This instance is connected to the public GitLab instance at https://www.gitlab.com

3.2.4.2. Authentication

1. Navigate to https://antflow.ospd.csgroup.space/ui
2. Click on the **Sign In with GitLab** button
3. Authenticate on GitLab
4. Grant the necessary permissions when prompted

5. You will be redirected to the AntFlow homepage upon successful authentication

3.2.4.3. Getting Started

3.2.4.3.1. Configure Your GitLab Project

Follow the project setup instructions to configure your GitLab project for AntFlow integration.

3.2.4.3.2. Access Your Project in AntFlow

Once configured, your project will appear on the AntFlow homepage.

3.2.4.3.3. Execute Workflows

1. Select your project from the home page market place
2. Choose the workflow you want to execute
3. Configure execution parameters
4. Launch the process
5. Monitor execution progress in real-time on the job details page

3.3. EOX Platform

3.3.1. Platform description

The EOX platform named **EOxHub Workspaces** provides a cloud-based platform environment that supports interactive analysis, workflow orchestration, and reproducible research for OSPD. It is built on open standards and open-source components, enabling scientists, developers, and platform operators to design, execute, and share workflows in a secure and scalable environment.

At its core, EOxHub Workspaces integrate JupyterLab for interactive development and exploration, and Argo Workflows for scalable, containerized execution of complex workflows, including support for parameterized inputs and reproducible outputs with the pygeoapi layer for exposing the Algorithm as a Service.

EOxHub Workspaces emphasize discoverability, interoperability, and reproducibility. Workflows are described and published as OGC API Records and STAC via the Open Science Catalogue (OSC, part of EarthCODE initiative), ensuring that execution environments and results are consistently documented and openly accessible.

3.3.2. Technology Stack

Here is a summary of key features that are available with EOxHub Workspaces in OSPD.2:

- Interactive analysis and development via JupyterLab;
- Execution of containerized workflows using Argo Workflows and pygeoapi;
- Integration with OGC APIs and STAC;
- Persistent, shareable environments; and
- Workflow cataloging and publication using Open Science Catalogue via OGC API Records with the STAC Application Extension.

3.3.3. How to access

In the scope of OSPD, there is one dedicated workspace made available at **https://workspace.deside.hub-ovh.eox.at/** using the `deside` namespace for experiments and attempts to reproduce the workflows.

3.4. GeoLabs Platform

3.4.1. Platform Description

This OpenAPI showcases the EOEPCA Application Deployment and Execution Service (ADES) building block. Based initially on the ZOO-Project, the ADES is now an integral part of it, operating under the code name ZOO-Project-DRU.

The ZOO-Project is an open-source processing platform created in 2008 and announced in 2009 at the FOSS4G conference in Sydney, Australia. Released under MIT/X11 license, it provides a generic processing platform to facilitate communication between your existing software and libraries.

3.4.2. Technology Stack

OGC API — Processes

The OGC API — Processes — Part 1: Core standard defines how a server implementation provides access to executable processes through a Web API and the capability to invoke them from a client application.

3.4.3. How to access

Endpoint	https://d113.preprod.ospd.geolabs.fr/swagger-ui/oapip/
The mangrove-workflow (KindGrove) Application Package	https://github.com/GeoLabs/KindGrove/blob/v0.0.1-rc5/mangrove_workflow_ogc_application_package.cwl

Authentication To use parts of this API, authentication with an OpenID Connect Provider is required. For this demonstration, we will use a Keycloak instance. Endpoints that require authentication are marked with an open lock icon on the right. To authenticate, please press the button, then a window appears, as shown below.

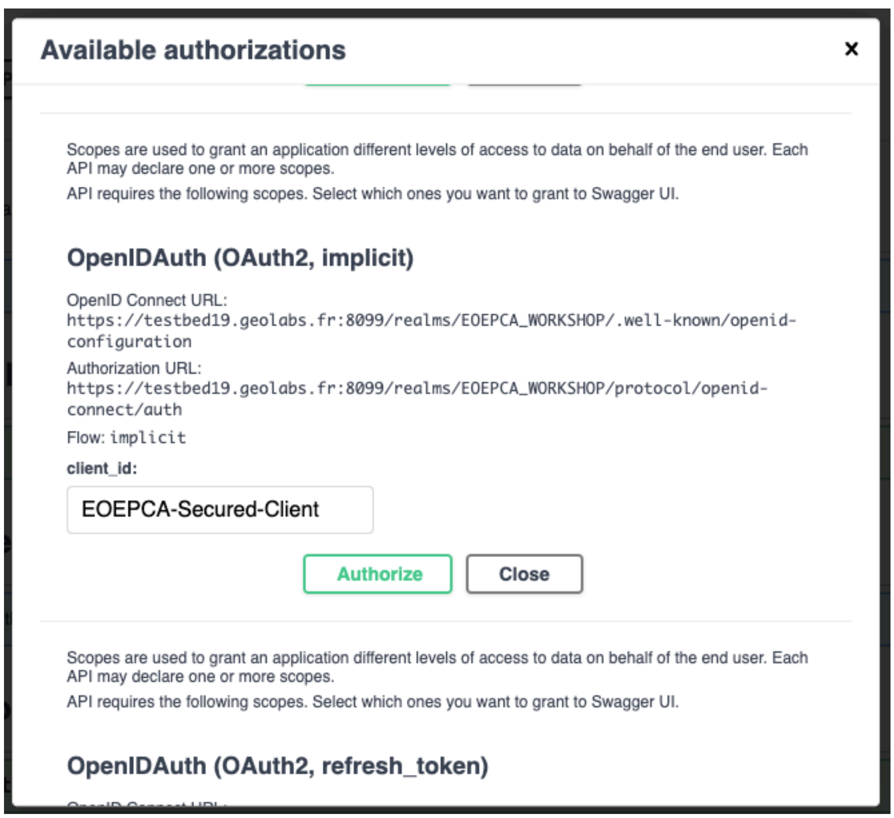


Figure 2 — Window to set client_id and authenticate

From there, use the following client_id: ‘zoo-project’ in the section OpenIDAuth (OAuth2,implicit) from the available authorizations. The login interface shown below allows the authentication.

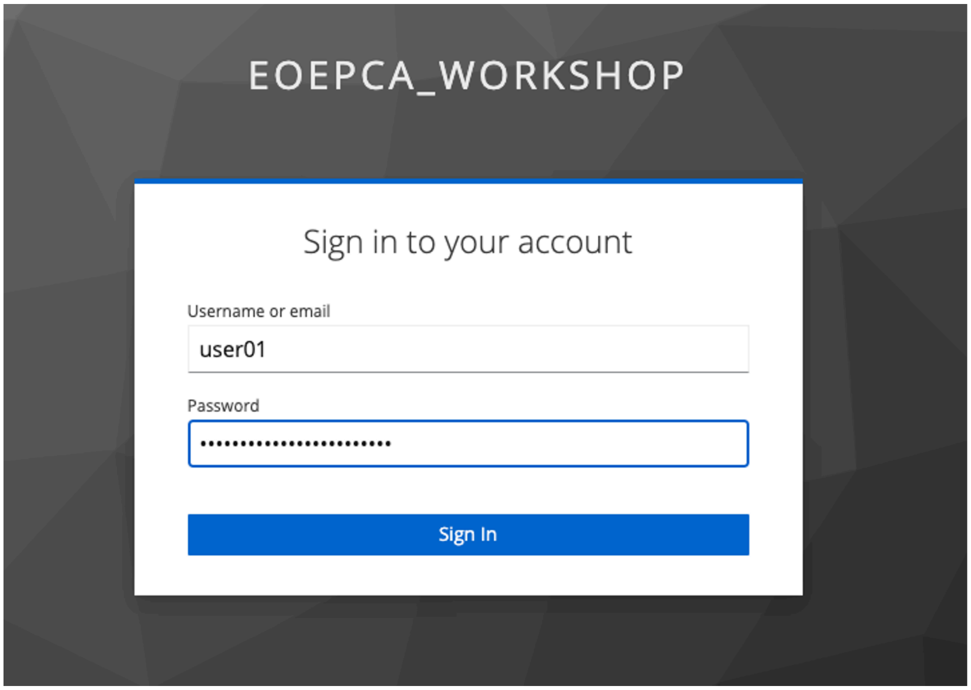


Figure 3

By clicking on the ‘OGC GitLab’ or ‘ORCID’ button depending on the type of user’s account, the process will continue. Once authenticated, the endpoint or any other secured endpoint will be used. In other cases, the user should get an exception response with a 401 status code.

3.5. Hartis Platform

3.5.1. Platform Description

The Hartis OSPD Platform is a multi-user execution environment designed to bridge interactive scientific analysis with scalable, reproducible workflow orchestration.

3.5.2. Technology Stack

Hybrid Interactive-Batch Execution System using:

- JupyterHub

- Kubernetes
- Argo Workflows

3.5.3. How to Access

Interactive Analysis Platform (JupyterHub)	https://hub.hartis.org
Workflow Execution Engine (Kubernetes/Argo)	https://argo.hartis.org
Platform Status & Updates	https://gitlab.ogc.org/ogc/OSPD-2025/-/issues/7

3.6. I-GUIDE platform

3.6.1. Platform description

Institute for Geospatial Understanding through Integrative Discovery Environment (I-GUIDE) is a National Science Foundation (NSF) funded institute that seeks to enable transformative discovery and innovation to tackle fundamental societal challenges by harnessing the large diversity of geospatial data being collected and managed by various organizations. With “geospatial data-on-demand” as its central theme, the I-GUIDE platform enables seamless integration of advanced cyberinfrastructure and cyberGIS capabilities to empower users to undertake computationally reproducible and data-intensive geospatial analytics at scale. The foundational unit of data on the platform is a knowledge element. Knowledge elements encompass datasets, codes, computational notebooks, learning modules, and research papers.

As part of OGC-OSPD, the I-GUIDE Platform demonstrated reproducibility of the Mangrove Biomass Estimation workflow hosted at https://github.com/starling-foundries/KindGrove

3.6.2. Technology Stack

The I-GUIDE Platform consists of the following major components:

- JupyterHub based execution environment for reproducible knowledge elements;
- MinIO based distributed object storage for hosting and accessing data knowledge elements;
- Neo4j and OpenSearch based backend for scalable and efficient knowledge discovery; and

- AI enabled advanced search capabilities for discovering relations among knowledge elements.

3.6.3. How to access

The platform itself is accessible at https://platform.i-guide.io/ Specifically the OSPD-25 related knowledge element can be accessed at https://platform.i-guide.io/notebooks/e4cbbc02-b826-408d-84f3-a937bf34436f

3.7. Terradue Srl Platform

3.7.1. Platform description

Terradue participates in OSPD 2025 as a platform operator through the **Geohazards Exploitation Platform (GEP)**. GEP is an operational open-science platform designed to support the exploitation of Earth Observation data for geohazards research, risk assessment, and monitoring applications.

GEP provides a persistent environment where scientific workflows can be deployed, executed, and validated close to the data. It supports reproducible research by combining standardized workflow descriptions, containerized execution environments, and structured metadata publication aligned with OGC standards.

Within OSPD 2025, GEP is used as a demonstrator platform to:

- execute and validate selected scientific workflows identified under WP1,
- test ontology-driven workflow descriptions developed under WP2,
- participate in cross-platform reuse experiments under WP3, and
- provide practical feedback for the refinement of OGC Open Science Building Blocks under WP4.

The platform is operated by Terradue and builds on experience gained from previous ESA and OGC initiatives, including the development and operation of Thematic Exploitation Platforms.

3.7.2. Technology Stack

The Geohazards Exploitation Platform relies on a modular, standards-based technology stack, including:

- **Containerized execution environments**, supporting reproducible and scalable processing;
- **Common Workflow Language (CWL)** for workflow definition and execution;
- **OGC Application Packages** to package workflows and their dependencies;
- **OGC API – Processes** for exposing workflows as executable services; and
- **STAC** and **OGC API – Records** for the discovery and description of input datasets, workflow artefacts, and results.

This stack enables workflows executed on GEP to be described and registered in a manner compatible with other OSPD platforms, supporting interoperability and reuse.

3.7.3. Role in OSPD 2025

In the context of OSPD 2025, GEP acts as both:

- a **reference execution environment** for validating workflow descriptions and metadata, and
- a **persistent demonstrator** showcasing how open standards can support operational open science.

Workflows executed on GEP contribute execution examples, provenance information, and metadata records that are used to validate the Open Science Ontology and the OGC Open Science Building Blocks. The platform also provides feedback on practical aspects such as parameter handling, data access patterns, and execution constraints, helping to ensure that the proposed standards remain applicable to real-world operational environments.

3.7.4. How to Access

Access to GEP within the scope of OSPD 2025 is provided through dedicated workflow descriptors and repositories. Selected workflows are packaged as Application Packages and made available through public version-controlled repositories, where execution descriptors and examples can be inspected and reused.

Workflow Descriptor	app-ml4floods.cwl
GitHub repo endpoint	app-ml4floods

Access details, including workflow references and execution endpoints, are provided to OSPD participants as part of the demonstrator activities and are subject to platform usage policies at the platform web site https://geohazards-tep.eu/

3.8. Telespazio UK Platform

3.8.1. Platform description

Telespazio UK leads the delivery of the ESA EarthCODE ecosystem of platforms and also leads the delivery for ESA's EOEPCA (Earth Observation Exploitation Platform Common Architecture). EarthCODE is ESA's strategic initiative to bring its vision of a FAIR and Open approach in EO science, applications and industry, and of a sustainable open innovation ecosystem. It integrates:

1) the Open Science Catalogue, making it effortless to discover novel research outcomes from ESA Science Clusters activities; 2) access to commercial platform services; and 3) a wealth of community resources and tools that help maintain data and research quality.

The Open Science Catalogue has been used by OSPD to publish Workflow and Experiment metadata. In addition, the EOX Platform that is integrated into the EarthCODE portal has been used by OSPD to deploy and demonstrate existing workflows. Complementary to EarthCODE, the overall objective of EOEPCA is to evolve, develop, and operate the generic standards and interoperable software building blocks (software components) enabling consolidation and harmonization of federated EO cloud and platform offerings that can support common utilization domains of the EO Science, R&D, and applications community. As a result of this, the EOEPCA Demo Platform has been used by OSPD to deploy and demonstrate existing workflows.

3.8.2. Technology Stack

The EarthCODE and EOEPCA technology stack currently applicable to OSPD includes:

- OGC API Records – to describe experiment and workflow metadata;
- STAC – to describe experiment results or product metadata;
- Git – to authenticate users and allow submission of publishing requests;
- CWL – to support the workflow lifecycle; and
- OGC Application Packages – to package workflows.

3.8.3. How to access

The EarthCODE ecosystem of Platforms provides access to a number of resources to support FAIR Open Science.

This includes the ESA Open Science Catalogue (live instance) which can be used to publish approved research. OSPD has made use of the ESA Open Science Catalogue (staging instance) which can be used to publish experiments and workflows for the purpose of OSPD. The

EarthCODE ecosystem includes EOX's EOxHub Platform and Terradue Srl Platform along with others. Access to the EarthCODE commercial platform services can be gained using ESA Nor Sponsorship NoR Network of Resources.

The EOEPCA Demo Platform was used by OSPD to demonstrate executing a workflow, as an OGC Application Package, on an available platform before attempting to reproduce this experiment on another platform, EOXHub Workspaces. The EOEPCA Demo Platform is accessed by teams further developing and experimenting with EOEPCA.

3.9. OGC Rainbow Platform

3.9.1. Platform description

The OGC Rainbow is envisioned as a knowledge graph reference holding the ontologies and building blocks instrumental to sharing knowledge in the geospatial domain, and specifically in the case of the OSPD: open science.

The OGC Rainbow eventually needs to become capable of housing and presenting all aspects of reference objects (ontologies, vocabularies, etc.), specialist objects (Building Blocks, OGC Standards), and instances of certain things needed for demo — e.g. instances of workflows represented using this project's WF ontologies.

Prez (https://github.com/rdflib/prez) is the underlying system that is used by the OGC Rainbow to host its knowledge graph in an API- and user-friendly way.

The details of the specific improvements made by KurrawongAI to Prez and the OGC Rainbow software in the scope of this project are listed in the annex in OGC 25-044, Open Science Persistent Demonstrator (OSPD) 2025 Building Blocks Report.

3.9.2. How to access

OGC Rainbow	https://defs.opengis.net/prez/#
Staging version	https://defs-dev.opengis.net/prez/#
Prez	https://github.com/rdflib/prez
Prez-ui	https://github.com/rdflib/prez-ui

4

OUTLOOK

4 OUTLOOK

4.1. A Roadmap to Facilitate the Adoption of Platforms Used in the OSPD 2025 Project

This roadmap outlines the technical steps required to facilitate the adoption of the platforms used during the project Open Science Persistent Demonstrator 2025. It is based on experiences collected during several reproducibility attempts aiming to run scientific workflows across different platforms. These scientific workflows comprise (see Clause 2.1):

1. Coastal Vulnerability Index (CVI) running on the HARTIS platform;
2. POLARIS and Polarwarp running on the EOxHub Workspaces platform;
3. Mangrove Detection and Biomass Estimation; and
4. Water bodies detection.

The providers of the workflows 1. and 2. also acted as platform operators and offered a platform to rerun the algorithms. Another platform joining the stack was I-Guide. However, the idea was to test and facilitate cross-platform execution of these workflows with a particular focus on interoperability, reproducibility, and portability.

4.1.1. Challenges

The reproducibility attempts carried out during the project revealed a number of challenges that hampered the execution of workflows. The mitigation of these challenges is an important step towards cross-platform reproducibility.

4.1.2. Computational Environment

Problems related to the computational environment are among the most common and complex challenges when it comes to reproducibility. This has also proven to be true in this project. For instance, the specification of libraries (or packages) and their versions was not always complete resulting in deviating computational environments created on the different platforms. Also, the kernels sometimes addressed the needs of a particular workflow. It was not clear how to configure it on another platform and if a differently configured kernel is actually a problem. The implications are equally complex: impeded executability caused by missing libraries or deviating outputs caused by deviating library versions.

4.1.3. Documentation

Issues related to the documentation comprised two aspects: How to-guiding steps and documentation specifying the workflows themselves. While the steps to run the workflows were not always clear from the beginning, their description improved significantly after contacting the corresponding workflow providers and platform operators. However, this is probably specific to projects such as OSPD where the participants received dedicated funding and less common in “typical” research projects where documentation is perceived as an additional task and burden.

Each workflow came with inputs such as input datasets and parameters and produced some outputs such as tables or images. While the provision of inputs ensures executability, they were not clearly specified making it difficult to understand how the inputs need to be like and what is needed or possible when using other datasets (“replication”). In addition, it was not clear if the generated outputs were correct as there were no outputs to compare with (example results). The idea of using building blocks to specify inputs and outputs can greatly mitigate these issues.

4.1.4. Accessibility

In some cases, the execution of workflows was only possible for users having an academic account. It was required to assign the same privileges to users not having such an account. This can limit reusability by a larger number of people. In other cases, credentials were required from third-party providers, which might become a problem if they apply similar or further restrictions.

4.1.5. Platform Limitations

Challenges also apply to the platforms themselves. For instance, because of missing job states and log messages it was not always clear why a workflow execution failed or took longer. It is not easy to anticipate such issues as a workflow developer and address them beforehand. Hence, this and the aforementioned issues should be seen as lessons learned.

4.1.6. Roadmap

4.1.6.1. Step 1: Establish a Portable Reproducibility Basis

First, a portable reproducibility basis requires information on the execution model (i.e., Jupyter notebook, Argo/Kubernetes, CWL etc.) and the runtime environment (incl. runtime, libraries, and versions, kernels). To ensure these definitions are interoperable, it is recommended to adhere to community-standard schemas.

In OSPD the utilization and extension of OGC API Record records as proposed in the Open Science Catalogue (OSC) was experimented with. The Listing 1 below (based on Polarwarp) shows the usage of the proposed STAC Application Extension to define the execution model and runtime environment.

Note the `conformsTo` which lists OGC API Records core, the STAC Application Extension, as well as to-be-defined application type schema for `application-type-argo-workflow`.

Further note the `properties` prefixed with `application:` and the links with a `rel` of `vcs`, `application`, and `application-originating-platform` which define links to the version control system like a GitHub repository, the application code itself, and the platform the workflow was originally developed on respectively.

```
{
  "id": "polarwarp",
  "type": "Feature",
  "conformsTo": [
    "http://www.opengis.net/spec/ogcapi-records-1/1.0/req/record-core",
    "https://stac-extensions.github.io/application/v0.1.0/schema.json",
    "https://raw.githubusercontent.com/EOEPCA/metadata-profile/refs/heads/1.0/
schemas/application-type-argo-workflow"
  ],
  "properties": {
    "type": "workflow",
    "application:type": "argo-workflow",
    "application:container": true,
    "application:language": "Python",
  },
  "links": [
    {
      "rel": "vcs",
      "title": "Git source repository",
      "href": "https://github.com/gtif-cerulean/polarwarp.git",
      "vcs:type": "git",
      "vcs:branch": "main"
    },
    {
      "rel": "application",
      "title": "Polarwarp workflow",
      "href": "https://github.com/gtif-cerulean/polarwarp/blob/main/workflow.
yml",
      "type": "application/x-argo-workflow-yaml",
      "application:type": "argo-workflow",
      "application:container": true,
      "application:language": "Python",
      "argo-workflow:": {
        "requirements": [
          {
            "temp_storage": "10GB"
          }
        ]
      }
    },
    {
      "rel": "application-originating-platform",
      "title": "EOxHub Workspaces",
      "href": "https://workspace.cif.hub-otc.eox.at/",
      "type": "text/html",
      "application:platform_supports": [
        "argo-workflow"
      ],
      "application:preferred_app": "argo"
    },
```

```
    {
      "rel": "related",
      "href": "https://harshness-map.gtif.eox.at/processes/execute-polarwarp-
gcps",
      "type": "text/html",
      "title": "Trigger workflow via API provided by pygeoapi"
    }
  ]
}
```

Listing 1 — Workflow definition example (truncated)

ESA's EOEPCA (Exploitation Platform Common Architecture) proposes to further elaborate and verify this approach (see for example the discussion on GitHub among others).

Second, inputs and outputs should be clearly defined, for example, in the form of building blocks allowing others to run the workflow not only with the original inputs but also with another configuration.

In the OSC concrete inputs (see example below) and outputs as Products are tied to Workflows using Experiments (Polarwarp example).

```
$comment: >-
  Lat and Lon are actually not Latitude and Longitude but coordinates in arctic
projection (EPSG:3413)
parameters:
  lon_min: 796454.0390764615
  lat_min: -2364879.127844905
  lon_max: 916915.4324799285
  lat_max: -2245914.126716489
  date: 2025-02-25
  num_gcps: 4
```

Listing 2 — Input definition example

Third, it is also necessary to list the tools needed to run the workflows such as Docker, Argo, STAC or Kubernetes.

4.1.6.2. Step 2: Improve Cross-Platform Reproducibility

Creating a container image is a good first step to facilitate cross-platform reproducibility. Tools like Docker are rather lightweight, commonly known, and easy to deploy. Also, it is crucial to revise the computational environment specified in step 1 and check if the used runtime, libraries, and kernels can actually run on different platforms or if they are limited, for instance, to specific operating systems or require certain computational resources (e.g., RAM, CPU, GPU). Equally important are the use of standards, such as those provided by the OGC API family.

4.1.6.3. Step 3: Add Logging Status Updates

Other users often first try to execute the workflow and start studying the steps if they fail. Learning about the issues is way easier if there are sufficient logging messages and clear status updates (e.g., in progress, failed, completed). Intermediate results can help to localize errors. This is particularly important considering that others may use the workflow later when the original developer is not available anymore.

4.1.6.4. Step 4: Provide Access & Sufficient Documentation

The platforms usually require authorization to run the workflows but the materials can still be provided on code sharing platforms such as GitHub or GitLab. This is also the place where a README including a how to-guide can significantly facilitate the execution by others.

A helpful guiding question in this context might be: Who should be able to run the workflow? The larger the number of targeted users, the more details are needed to accommodate the different skill levels. Ideally, instructions are provided for different user groups. If access is limited to certain user groups, information is needed on how users can get access (and if at all).

Finally, documentation extends beyond the `README.md`. To ensure the workflow is discoverable by the community, developers should add metadata to a centralized catalog, such as the Open Science Catalogue (OSC). This involves creating a metadata entry that explicitly references the workflow's Git repository. By registering the workflow in the catalog, a direct link between the discovery mechanism (the catalog) and the implementation (the Git repo) is created, ensuring that users finding the science can immediately access the source code required to reproduce it.

4.1.6.5. Summary

The four steps outlined above are certainly not complete but should focus on certain aspects that played a major role in this project. Also, there are already numerous guidelines and best practices in literature but it is important to note that many workflows have individual challenges as it could be seen in this project. Hence, there is no one-size-fits-all solution but the concepts and technologies used in OSPD 2025 pave the way for more robust, transparent, reproducible, and portable research workflows.

5

SECURITY, PRIVACY AND ETHICAL CONSIDERATIONS

5 SECURITY, PRIVACY AND ETHICAL CONSIDERATIONS

An extensive assessment was carried out to detect possible issues related to security, privacy, and ethics. Upon thorough review, it was concluded that these aspects were not pertinent to the report's scope and subject matter. Consequently, no particular actions or safeguards were deemed necessary.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Pedro Gonçalves, Ingo Simonis, Micah Brachman, Open Geospatial Consortium: OGC 24-022, *OGC Open Science Persistent Demonstrator (OSPD) Report*. Open Geospatial Consortium (2025). http://www.opengis.net/doc/PER/ospd.

[2] OGC: OGC 25-044: Open Science Persistent Demonstrator (OSPD) 2025 Building Blocks Report, 2026

A

ANNEX A (NORMATIVE) ABBREVIATIONS/ACRONYMS

A ANNEX A (NORMATIVE) ABBREVIATIONS/ACRONYMS

ADES	Application Deployment and Execution Service
AI	Artificial Intelligence
AOI	Area of Interest
API	Application Programming Interface
AWS	Amazon Web Services
CCO	Common Core Ontologies
CF	Climate and Forecast
COG	Cloud Optimized GeoTIFF
CSV	Comma-Separated Values
CVI	Coastal Vulnerability Index
CWL	Common Workflow Language
DCAT	Data Catalog Vocabulary
DEM	Digital Elevation Model
EO	Earth Observation
EOAP	Earth Observation Application Package
EOEPCA	Earth Observation Exploitation Platform Common Architecture
ESA	European Space Agency
EU	European Union
FAIR	Findable, Accessible, Interoperable, Reusable
GEP	Geohazards Exploitation Platform

GPU	Graphics Processing Unit
HPC	High-Performance Computing
HTTP	Hypertext Transfer Protocol
I-GUIDE	Institute for Geospatial Understanding through Integrative Discovery Environment
InSAR	Interferometric Synthetic Aperture Radar
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
LiDAR	Light Detection and Ranging
NASA	National Aeronautics and Space Administration
NDVI	Normalized Difference Vegetation Index
NDWI	Normalized Difference Water Index
NOAA	National Oceanic and Atmospheric Administration
NoR	Network of Resources
OGC	Open Geospatial Consortium
OS	Operating System
OSC	Open Science Catalogue
OSPD	Open Science Persistent Demonstrator
POLARIS	Polar Operational Limit Assessment Risk Indexing System
PROV-O	PROV Ontology
RAM	Random Access Memory
RDF	Resource Description Framework
RIO	Risk Index Outcome
SAR	Synthetic Aperture Radar
SAVI	Soil-Adjusted Vegetation Index
SIGRID-3	Sea Ice Grid, version 3

STAC	SpatioTemporal Asset Catalogue
TEP	Thematic Exploitation Platform
UI	User Interface
UK	United Kingdom
URL	Uniform Resource Locator
USGS	United States Geological Survey
VCS	Version Control System
VEDA	Visualization, Exploration, and Data Analysis
WMO	World Meteorological Organization
WP	Work Package
XML	Extensible Markup Language

B

ANNEX B (INFORMATIVE) COASTAL VULNERABILITY INDEX (CVI) WORKFLOW OVERVIEW

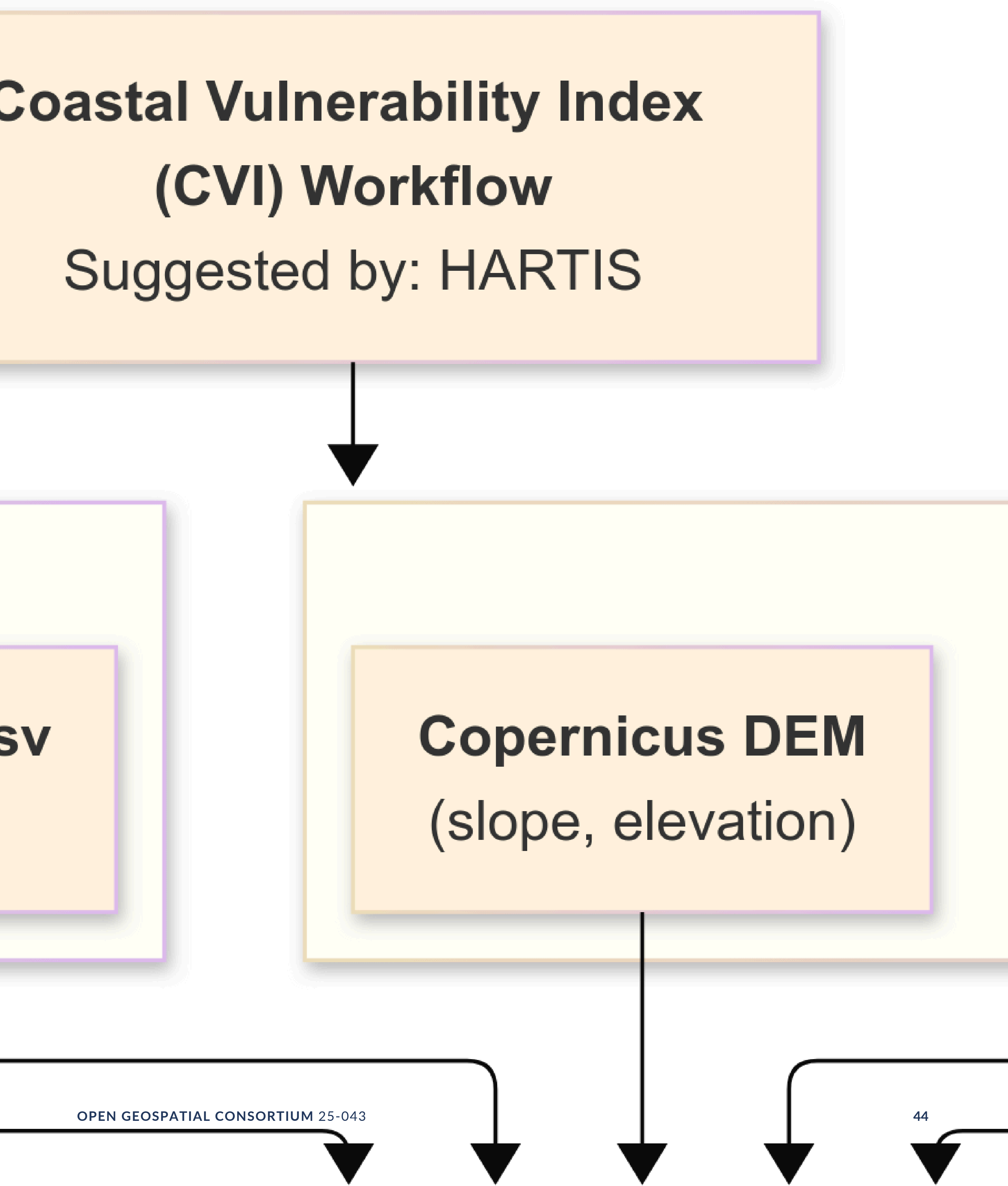
B ANNEX B (INFORMATIVE) COASTAL VULNERABILITY INDEX (CVI) WORKFLOW OVERVIEW

The **Coastal Vulnerability Index (CVI)** workflow is an automated, reproducible system designed to classify coastal risk based on the standardized USGS/NOAA methodology. The workflow integrates Earth Observation (EO) data—specifically Digital Elevation Models (DEM) and Land Cover—with generated coastal transects to compute physical parameters and assess vulnerability.

The implementation relies on the **Common Workflow Language (CWL)** and is encapsulated within **Docker containers**, ensuring execution portability across diverse environments, ranging from local deployments to High-Performance Computing (HPC) clusters and OSPD-compliant JupyterHubs.

B.1. Workflow Structure and Data Flow

The overall structure of the CVI workflow, including configuration files, input datasets, processing notebook, and generated outputs, is illustrated in Figure B.1.


Coastal Vulnerability Index
(CVI) Workflow
Suggested by: HARTIS
sv
Copernicus DEM
(slope, elevation)

The workflow is driven by a configuration file (`cvi_scoring_simple.json`) and a list of Areas of Interest (AOIs), and combines Copernicus DEM, ESA WorldCover, and Coastal Hazard Wheel data to compute transect-based CVI values.

B.2. Workflow Metadata Improvements

B.2.1. Input Data Description

To align with OSPD interoperability targets, the input data architecture is being refactored to support the **OGC Building Block for Bounding Box** (`ogc.geo.common.data_types.bounding_box`). While the current implementation relies on coordinate arrays, the planned integration of this building block will standardize the Area of Interest (AOI) definition, ensuring semantic consistency with other OGC API – Processes implementations.

In addition, input datasets (e.g. Copernicus DEM) are referenced through standardized metadata structures to facilitate machine-readable discovery and automated processing.

B.2.2. Output Data Description

Workflow outputs are structured to support immediate publication through **OGC API – Features** (for vector-based coastal transect data) and **OGC API – Coverages**.

Upon completion of execution, the results are wrapped in a static **STAC Catalog**, providing a standardized, self-describing manifest of the generated vulnerability classification maps and associated data tables.

An example visualization of the final CVI output, rendered as an interactive map, is shown in Figure B.2.

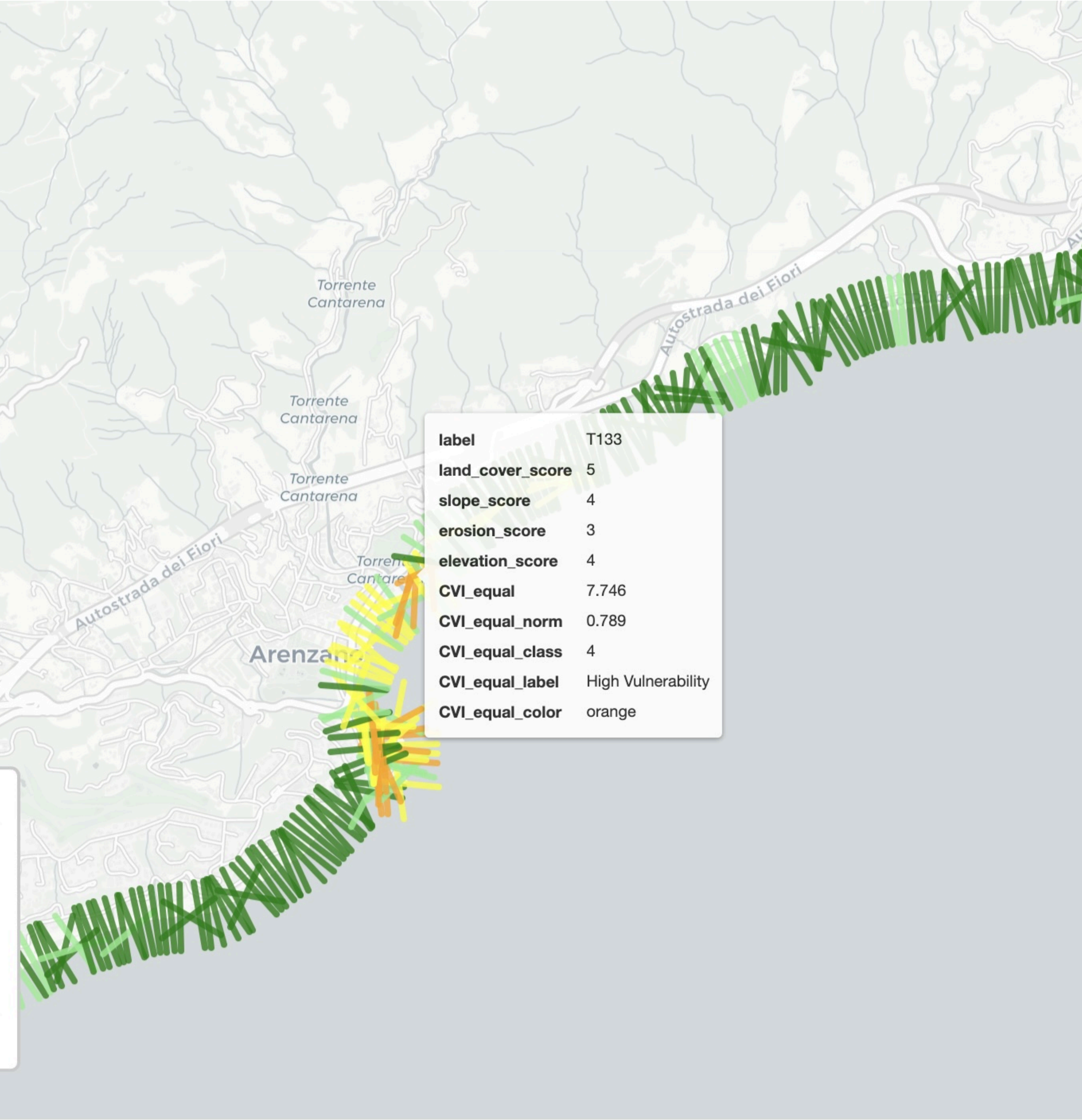


Figure B.2 — Example CVI result map showing transect-based vulnerability classes

B.2.3. Workflow Step Description

The workflow logic is formally described using **CWL**, decomposing the scientific process into discrete, independently validatable steps:

1. Transect generation
2. Satellite data retrieval
3. Parameter computation
4. Risk classification

This modular design allows individual processing components to be inspected, validated, and potentially reused within the broader OSPD ecosystem.

B.2.4. Interoperability of Execution Provenance

A formal **OGC API – Records** entry has been generated and staged within the Open Science Catalogue (OSC) to support execution provenance and lineage tracking.

The record (`id: cvi-workflow`) conforms to the `ogcapi-records-1` core specification and the OSC STAC extensions (`v1.0.0`), providing structured execution metadata, including:

- **Version control**: A direct link (`rel: vcs`) to the Git source repository https://github.com/hartis-org/cvi-workflow
- **Application definition**: Explicit linkage (`rel: application`) to the executable CWL definition (`cvi_workflow.cwl`), declared with MIME type `application/x-cwl`
- **Environment specification**: Metadata flags indicating containerized execution support (`application:container: true`) and language declaration (`application:language: Python`)

B.3. Workflow Engine Support Improvements

To support deployment across the federated OSPD architecture, workflow engine support was enhanced through the following measures:

1. **OSC metadata registration** The workflow has been registered in the Open Science Catalogue (OSC) staging environment. This metadata record enables OSPD platform operators (e.g. EarthCODE, EOxHub Workspaces) to

automatically harvest, catalogue, and expose the workflow as an available resource.

2. **Containerization for portability** The workflow engine logic has been strictly decoupled from the hosting environment using Docker. As defined in the metadata record, the `application:type` is set to `cwl-workflow` executed within a container, ensuring consistent runtime instantiation across cloud-based JupyterHubs and local execution environments.

B.4. Documentation and Endpoints

- **Workflow source and documentation**: https://github.com/hartis-org/cvi-workflow
- **OGC API Record (metadata)**: https://github.com/ESA-EarthCODE/open-science-catalog-metadata-staging/blob/main/workflows/cvi-workflow/record.json
- **Demonstration platform**: https://hub.hartis.org

C

ANNEX C (INFORMATIVE) CS GROUP CONTRIBUTIONS TO THE OGC OPEN SCIENCE PERSISTENT DEMONSTRATOR

C ANNEX C (INFORMATIVE) CS GROUP CONTRIBUTIONS TO THE OGC OPEN SCIENCE PERSISTENT DEMONSTRATOR

Here are the CS Group contributions to the OGC Open Science Persistent Demonstrator.

C.1. Overview

We present the contributions made within the framework of this project along 3 major axes that were worked on during the integration of the KindGrove and CVI-process processes on our execution platform called **AntFlow**.

CS Group is involved in OSPD as platform operator and has delivered and deployed **AntFlow**, a dynamic and evolving research environment and an orchestrator that fulfills several key requirements: reproducibility, efficiency, interoperability, transparency, performance and scalability, monitoring and diagnostics, open standards and sharing. AntFlow is a lightweight, standards-based orchestration platform that we have developed to streamline scientific workflow deployment and execution.

The general principle of AntFlow is to read EOAPs (from a Git repository, on GitLab) and translate them on-the-fly into an OGC API Processes service through automatic conversion of workflow inputs and outputs. After that, the platform enables the execution of processes through automatically generated graphical interfaces, and finally executes the processing on various computing platforms (cloud, HPC, local computer).

Our contributions to OSPD are organized around three key areas of work: **Packaging**, **Normalization**, and **CWL Extensions**. The work carried out on the AntFlow platform dedicated to OSPD enabled us to work on workflow execution while developing enhancements at both the AntFlow platform level and the workflow level.

C.2. Packaging

During the integration of the CVI process, we applied modifications to make the OGC application package more portable. We noticed that this package was using some Python scripts external to the Docker image, which made it less interoperable and more dependent on the execution environment. We therefore modified the CWL as well as the Docker image accordingly to address this issue. After that improvement, the Docker image is standalone and contains all the mandatory resources for the processing to run on a OGC EO application package compatible platform.

C.3. Normalization

This contribution axis in the OSPD project consisted of bringing evolutions to the processes proposed and tested on our platform to move towards greater compliance with OGC Application Package (EOAP) best practices.

C.3.1. Graph type CWL

When testing of the CVI process, we observed that the existing workflow diverged from the OGC best practices recommendations for Application Packages. Indeed, our AntFlow implementation follows this best practices guide and therefore expects to find a "graph" type CWL file. We consequently made this modification to enable the workflow to run on our platform.

C.3.2. Bounding box input support

In the scope of that project, we implemented on the AntFlow platform support of the EOAP standardized bounding box schema, enabling workflows to declare geographic inputs using community-maintained type definitions. This approach promotes interoperability and compliance with OGC best practices.

Workflow authors can declare bounding box inputs in CWL using the following EOAP schema reference:

```
inputs:
  bounding_box:
    label: Bounding box
    type: https://raw.githubusercontent.com/eoap/schemas/0.2.0/ogc.yaml#BBox
```

Listing C.1

The AntFlow server's CWL parser detects the input type and automatically generates an OGC API Processes-compliant schema with format ogc-bbox. This standardized format triggers the dynamic UI generation system on the frontend.

The generated OGC API Processes input definition follows this structure:

```
{
  "id": "ID",
  "title": "Mangrove Biomass Workflow",
  "inputs": {
    "bounding_box": {
      "title": "Bounding box",
      "description": "Area of interest bounding box [west, south, east, north]",
      "minOccurs": 1,
      "schema": {
        "required": [
          "bbox"
        ],
        "type": "object",
        "format": "ogc-bbox",
        "properties": {
          "bbox": {
            "maxItems": 6,
            "minItems": 4,
            "type": "array",
            "items": {
              "type": "number"
            }
          },
          "crs": {
            "enum": [
              "CRS84",
              "CRS84h"
            ],
            "type": "string",
            "default": "CRS84"
          }
        }
      }
    }
  }
}
```

Listing C.2

On the frontend, the dynamically generated interface automatically renders an interactive Leaflet map widget for bounding box inputs. Users can draw the bounding box directly on the map, making geographic area selection intuitive. The widget ensures coordinates are properly formatted and within valid ranges, eliminating manual coordinate array entry and reducing errors while improving the user experience for geospatial workflows.

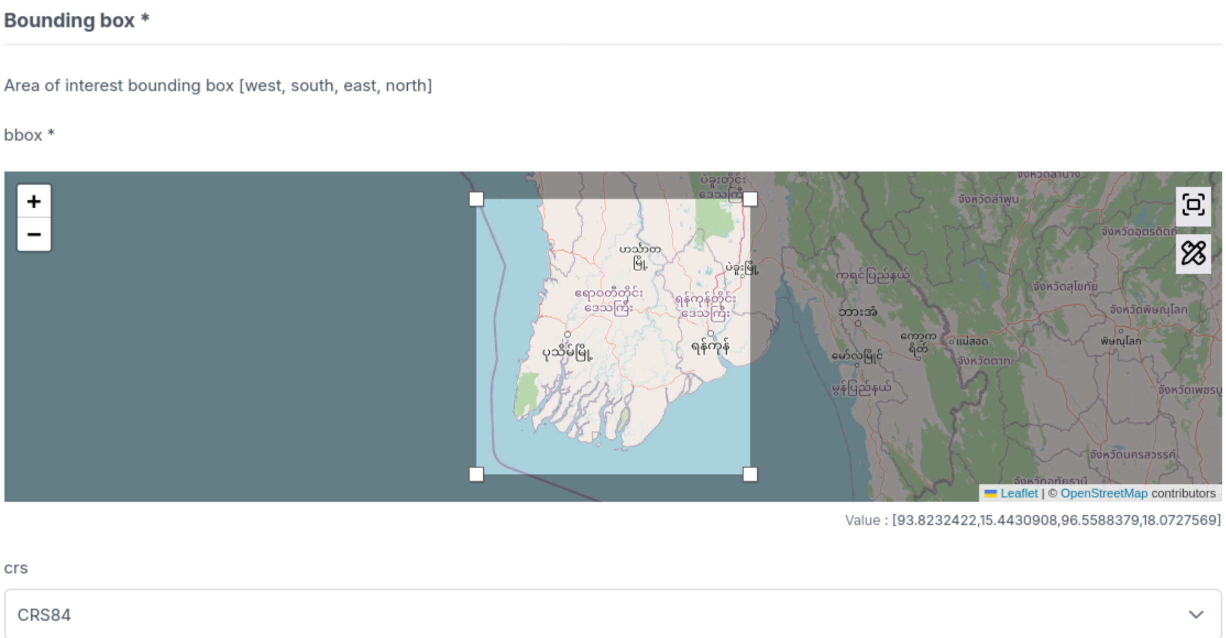


Figure C.1

C.3.3. Input management

The CVI processing was initially based on file-type inputs. In the context of web execution and to improve application package portability, we enhanced the application package to replace file-type inputs with inputs pointing to STAC item URLs. Following this change, we integrated a parallelized data-collect step, based on the EODAG component, an Open Source library developed by CS GROUP, which has the capability to automatically download assets from a STAC item.

Here is the CVI workflow after that enhancement:

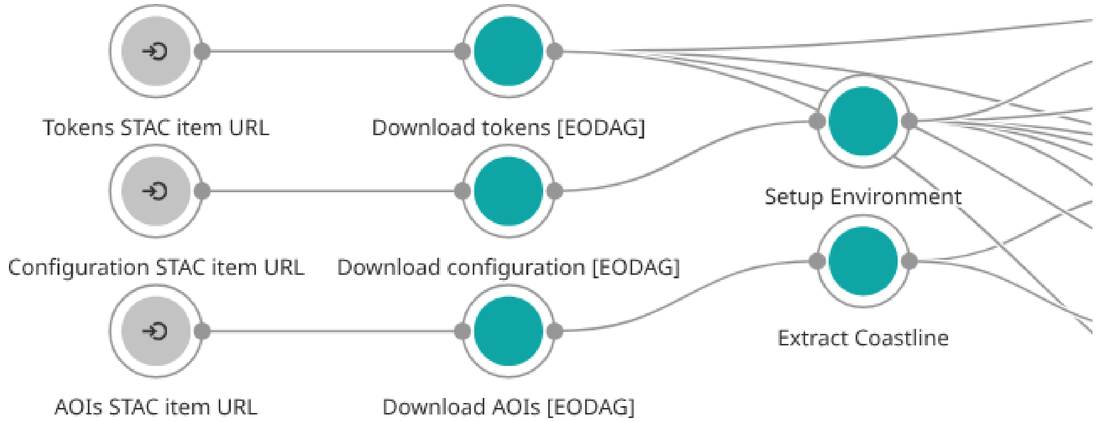


Figure C.2

C.3.4. Output management

Following the OGC EOAP best practices, the AntFlow platform implements automatic generation of static STAC catalogs for workflow outputs through a configurable wrapping mechanism. This enhancement addresses the need for standardized, self-describing output metadata that enables downstream discovery and processing.

The platform provides workflow orchestration capabilities that automatically wrap user workflows with an additional step for archiving results to S3 buckets and cataloging them in static STAC catalogs also store on S3 bucket. This approach allows workflows to focus on their core processing logic while the platform handles output management.

Here is an example of the CVI workflow where you can see the automatically added “Publication” step:

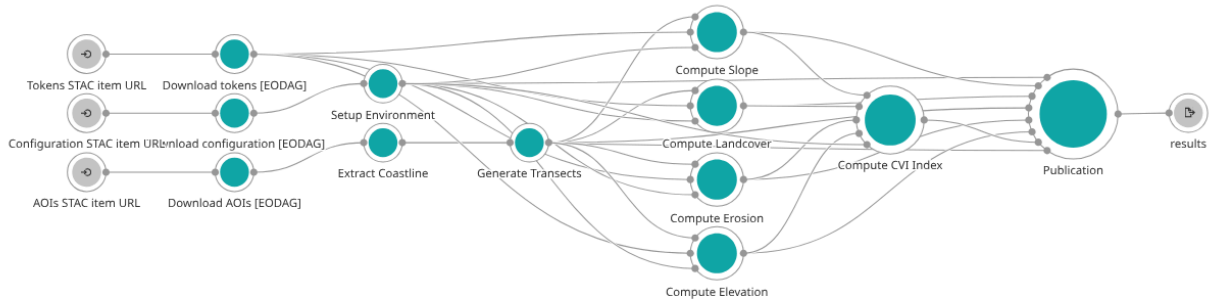


Figure C.3

The generated static STAC catalog structure includes:

- A root catalog.json describing the overall processing output
- An optional collection.json at the collection level, grouping related STAC Items by common characteristics (e.g., processing date, geographic area, or product type)
- Linked to individual STAC Items for each generated product with complete metadata (spatial extent, temporal information, assets)

This automatic STAC catalog generation provides several benefits:

- **Interoperability**: Outputs can be easily consumed by any STAC-compliant tool or service;
- **Traceability**: Complete provenance information is embedded in the STAC metadata;
- **Discoverability**: Processing results become searchable through STAC catalogs and APIs; and
- **Portability**: The self-contained catalog can be moved or archived while retaining all metadata.

See also the STAC catalog customization section for output customization.

C.4. Added extensions

C.4.1. How to control execution platform targeting

The AntFlow platform implements a flavor-based job routing system that allows workflows to specify execution requirements and enables launchers (the AntFlow process execution component) to claim jobs matching their capabilities. Flavors are declarative tags (like “hpc”, "high-memory", "kubernetes") that define specific execution characteristics or resource requirements.

CWL Workflow Definition: Workflow authors can declare required flavors directly in CWL files using the FlavorInfo hint class:

```
class: Workflow
hints:
  - class: FlavorInfo
    flavors: ["hpc", "high-memory"]
```

Listing C.3

Server Processing: When converting CWL workflows to OGC API Process description, the server extracts FlavorInfo hints and populates the job's metadata.

Job Matching: the AntFlow Launchers component registers with the server by declaring their supported_flavors (e.g. ["local", "kubernetes"], ["hpc", "high-storage"]). When a launcher polls the job queue, the server filters jobs that match the launcher's supported flavors against jobs' requested flavors. Jobs with no flavors or with flavors compatible with the launcher's capabilities are returned. The matching logic ensures that all requested flavors must be present in the launcher's supported set, enabling precise targeting of specialized execution platforms.

Launcher Execution: Once assigned, launchers execute workflows on their respective platforms (Kubernetes, local, HPC, etc.), with resource allocation and environment configurations tailored to the declared flavors. This decoupled architecture allows heterogeneous execution environments to coexist, with each launcher handling jobs suited to its infrastructure capabilities.

C.4.2. Dynamic UI generation

The AntFlow platform implements dynamic user interface generation for processes posting from JSON Schemas and JSON Forms UI Schemas, enabling automatic creation of adaptive forms without specific development for each process. The flow begins with the conversion of CWL (Common Workflow Language) workflows into OGC API processes-compliant process descriptions with JSON Schema on the backend. These schemas, enriched with metadata (date formats, geographic bounding boxes, numeric ranges), are then analyzed on the frontend to automatically generate UI Schemas compatible with React JSON Schema Form. The system intelligently determines which control to display: an interactive Leaflet map for bounding boxes, sliders with numeric inputs for min/max values, date pickers for temporal formats, etc. This approach delivers a rich and contextual user experience for process execution without requiring manual form development, while ensuring data validation through JSON Schema and automatically adapting to each workflow's specifications.

This approach is compatible with JSON Schemas, therefore natively compatible with both CWL and OGC API Processes specifications.

Here are several examples of that feature.

CWL date input:

```
date_start:
```

```
  label: Date start
  doc: Considered start date used to search input products
  type: string
  json-schema:format: date
```

Listing C.4

Generated OGC API processes date input:

```
"date_start": {
  "title": "Date start",
  "description": "Considered start date used to search input products",
  "minOccurs": 1,
  "schema": {
    "type": "string",
    "format": "date"
  }
},
```

Listing C.5

Generated UI widget:

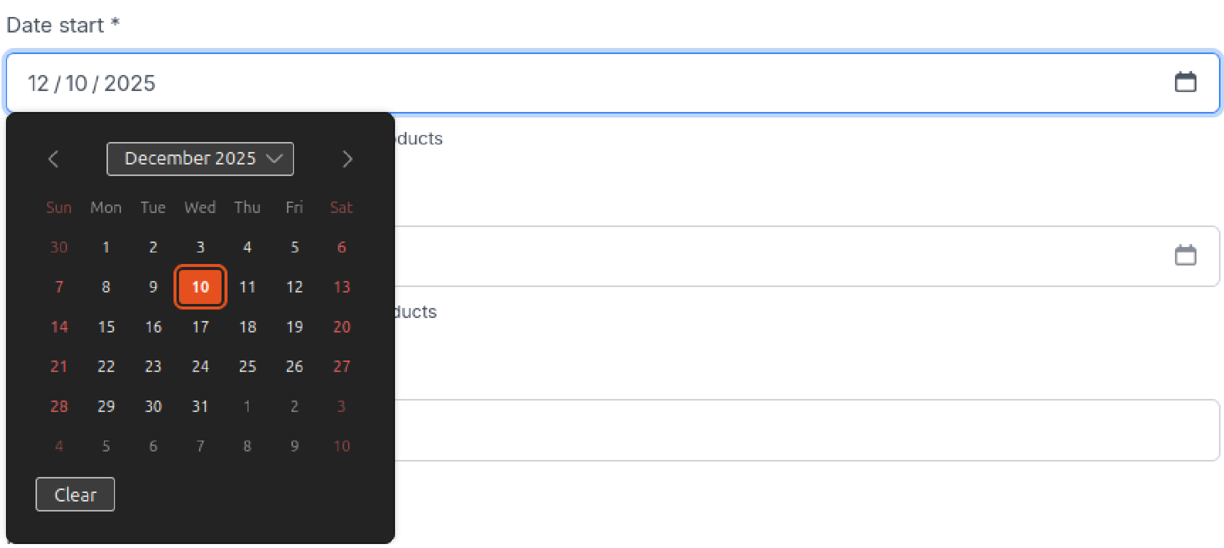


Figure C.4

CWL range input:

```
cloud_cover:
  label: Cloud Cover Percent Input
  doc: Cloud cover percent input field
  type: float
  json-schema:schema:
    minimum: 0
    maximum: 100
  default: 50
```

Listing C.6

Generated OGC API processes range input:

```
"cloud_cover": {
  "title": "Cloud Cover Percent Input",
```

```
  "description": "Cloud cover percent input field",
  "minOccurs": 1,
  "schema": {
    "type": "number",
    "minimum": 0.0,
    "maximum": 100.0,
    "default": 50
  }
},
```

Listing C.7

Generated UI slider widget:

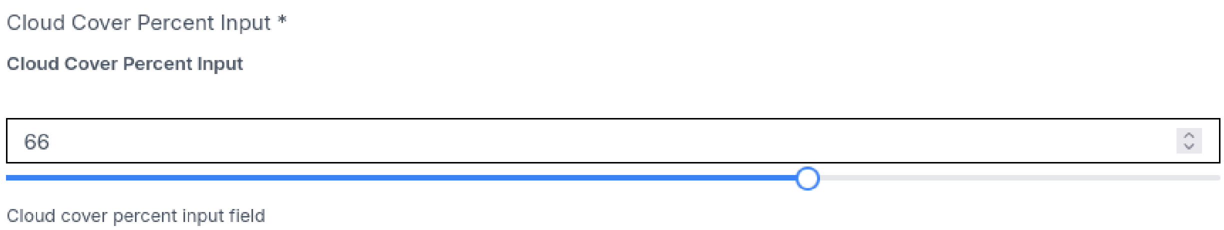


Figure C.5

Are also supported by AntFlow Boolean, fields (generating a checkbox UI widget), enumerations (generating a dropdown UI widgets) and arrays.

C.4.3. STAC catalog customization

The AntFlow platform enables workflow authors to customize the generated STAC (SpatioTemporal Asset Catalog) items and catalogs through the publish:stac CWL directives. This feature provides fine-grained control over output STAC metadata, allowing workflows to enrich output catalogs with domain-specific properties, custom links, and asset definitions that enhance discoverability and interoperability with downstream STAC-compliant tools.

The publish:stac configuration is passed directly from the CWL workflow to the publisher component during the workflow wrapping process. This approach separates concerns: workflow authors declare their output catalog structure declaratively in CWL, while the platform handles the technical implementation of STAC generation.

C.4.3.1. Custom Properties

Workflow authors can add arbitrary properties to STAC items to provide domain-specific metadata. These properties are merged into the STAC item's properties object:

```
publish:stac:
  properties:
    software: "GDAL 3.8, EODAG 2.12"
    processing_level: "L2A"
    product_type: "reflectance"
```

Listing C.8

This generates STAC items with enriched metadata that downstream consumers can filter and search by specific processing characteristics.

C.4.3.2. Managing Links

Links can be added to STAC items in two modes: static links and dynamic links based on asset pattern matching.

Static Links are added directly to the STAC item's links array without asset matching:

```
publish:stac:
  links:
    - rel: related
      href: "https://doi.org/10.1234/dataset"
      type: "text/html"
      title: "Related Publication"
```

Listing C.9

Dynamic Links use regular expressions to match assets and generate templated links. The system supports placeholders that are formatted based on matching assets:

- {0} — The matching asset path
- {0:url} — The full HTTP URL of the matching asset
- {0:url-path} — The URL path portion of the matching asset
- {0:name} — The asset filename

```
publish:stac:
  links:
    - rel: visualization
      type: "text/html"
      file_regex: "(.*)/layer.json"
      title: "View {0:name} in Viewer"
      href: "https://viewer.example.com/#data={0:url}"
```

Listing C.10

This example creates visualization links for any asset matching the pattern, with the asset URL automatically injected into the viewer's URL parameter.

C.4.3.3. Managing Assets

Asset definitions allow workflows to customize how output files appear in the STAC catalog. Similar to links, assets support both static and pattern-based configurations.

Static Assets define additional assets with explicit metadata:

```
publish:stac:
  assets:
    - href: "image.tif"
      roles: ["data"]
```

```
      media_type: "image/tiff; application=geotiff; profile=cloud-optimized"
```

Listing C.11

Pattern-Based Assets use regular expressions to match and customize asset metadata. This is particularly useful for organizing large numbers of generated files:

```
publish:stac:
  assets:
    - file_regex: "(.*)\\.tif$"
      roles: ["data"]
      media_type: "image/tiff; application=geotiff; profile=cloud-optimized"
    - file_regex: "(.*)/thumbnail\\.png$"
      roles: ["thumbnail"]
      media_type: "image/png"
```

Listing C.12

The system merges these configurations with automatically discovered assets, allowing workflows to override default metadata (like MIME types or roles) based on file patterns.

This customization capability was developed to support the OSPD requirements for rich, self-describing output catalogs that integrate seamlessly with existing STAC ecosystems while allowing scientific workflows to maintain domain-specific metadata standards.

C.5. Suggestions for improvements

A possible improvement would be to provide secret management through the AntFlow platform. The CVI process requires as input a file containing access secrets to an S3 bucket, which can expose sensitive information to leakage. AntFlow provides secure storage management for these secrets. This information is automatically added to the execution environment as environment variables when launching the processing. This ensures security in the management of these variables.

AntFlow provides a solution but it is not currently satisfactory. Secret management remains an open question.

C.6. Summary

This document outlines CS Group's contributions to the OGC Open Science Persistent Demonstrator (OSPD) through the development and deployment of **AntFlow**, a lightweight orchestration platform for scientific workflows. The contributions are organized around three main axes:

Packaging: Enhanced provided Earth Application Application Packages portability, making them fully standalone and not dependent to the execution environment.

Normalization: Implemented several improvements to align workflows with OGC EOAP best practices, including:

- Migration to graph-type CWL workflows;
- Support for standardized bounding box inputs with interactive map widgets;
- Replacement of file-type inputs with STAC item URLs and integration of parallelized data collection using EODAG; and
- Automatic generation of static STAC catalogs for workflow outputs, ensuring interoperability and traceability.

CWL Extensions: Developed platform enhancements to improve workflow execution capabilities:

- Job execution platform assignement with a flavored-based system;
- Dynamic job posting UI generation from JSON schemas, automatically creating adaptive forms with appropriate widgets; and
- STAC catalog customization, enabling workflows to enrich output metadata with custom properties, links, and asset definitions.

C.7. Contributions references

Proposed pull requests on workflows GitHub repositories:

- https://github.com/starling-foundries/KindGrove/pull/2
- https://github.com/starling-foundries/KindGrove/pull/5
- https://github.com/hartis-org/cvi-workflow/pull/1

D

ANNEX D (INFORMATIVE) EOX CONTRIBUTIONS TO THE OGC OPEN SCIENCE PERSISTENT DEMONSTRATOR

ANNEX D (INFORMATIVE) EOX CONTRIBUTIONS TO THE OGC OPEN SCIENCE PERSISTENT DEMONSTRATOR

D.1. Introduction

This section describes the contributions of EOX to the OGC Open Science Persistent Demonstrator (OSPD). EOX participated in the project in two roles:

- as a **Scientist**, through the development, maintenance, and execution of EO processing workflows; and
- as a **Platform operator**, by providing the EOxHub Workspaces platform as a reproducible execution environment supporting cross-platform validation and reuse.

The contributions are organized accordingly, first describing scientist-related activities and then platform operator activities and support.

D.2. Workflow Contributions

D.2.1. POLARIS Workflow

D.2.1.1. Overview

The **POLARIS** workflow was co-developed by EOX as part of the OSPD activities and served as an early demonstrator for reproducible workflow execution across platforms.

The workflow was containerized and integrated into an Argo-based execution model, allowing it to be executed on the EOxHub Workspaces platform within a controlled and reproducible environment. POLARIS was successfully executed on EOxHub and registered in the Open Science Catalogue (OSC) as an OGC API – Records entry (see below), enabling discovery and metadata-driven access.

Although POLARIS was not ultimately selected as the primary workflow for cross-platform reproducibility demonstrations within OSPD, it played an important role in validating workflow packaging, execution, and cataloguing mechanisms. The experience gained from its integration informed subsequent improvements to workflow portability and execution practices.

D.2.1.2. Documentation and Endpoints

- Workflow source and documentation:
  - https://github.com/gtif-cerulean/polaris
- OGC API Record (metadata) on the Open Science Catalogue, for now on the staging deployment:
  - Workflow: https://osc-staging.earthcode.eox.at/workflows/polaris/record
  - Product: https://osc-staging.earthcode.eox.at/products/polaris/collection
  - Experiment: https://osc-staging.earthcode.eox.at/experiments/polaris/record
- Demonstration platform:
  - https://workspace.deside.hub-ovh.eox.at

D.2.2. Polarwarp Workflow

D.2.2.1. Overview

The **Polarwarp** workflow implements a reproducible processing chain for geolocating, aligning, and warping Synthetic Aperture Radar (SAR) imagery in polar regions using forecast ice drift and environmental forcing fields. The workflow is designed to support monitoring of sea ice motion over short forecast windows while preserving accurate georeferencing of satellite imagery.

The core algorithm ingests SAR images (e.g. Sentinel-1) together with forecast and analysis data describing sea ice drift, wind, and tidal forcing. Ice drift vectors from multiple sources are translated into pixel-level displacements, which are then applied to the SAR imagery using a thin-plate spline warping approach. This method enables temporal alignment and drift correction of images while maintaining spatial consistency in polar projections.

A schematic overview of the workflow and data flow is provided in Figure D.1.

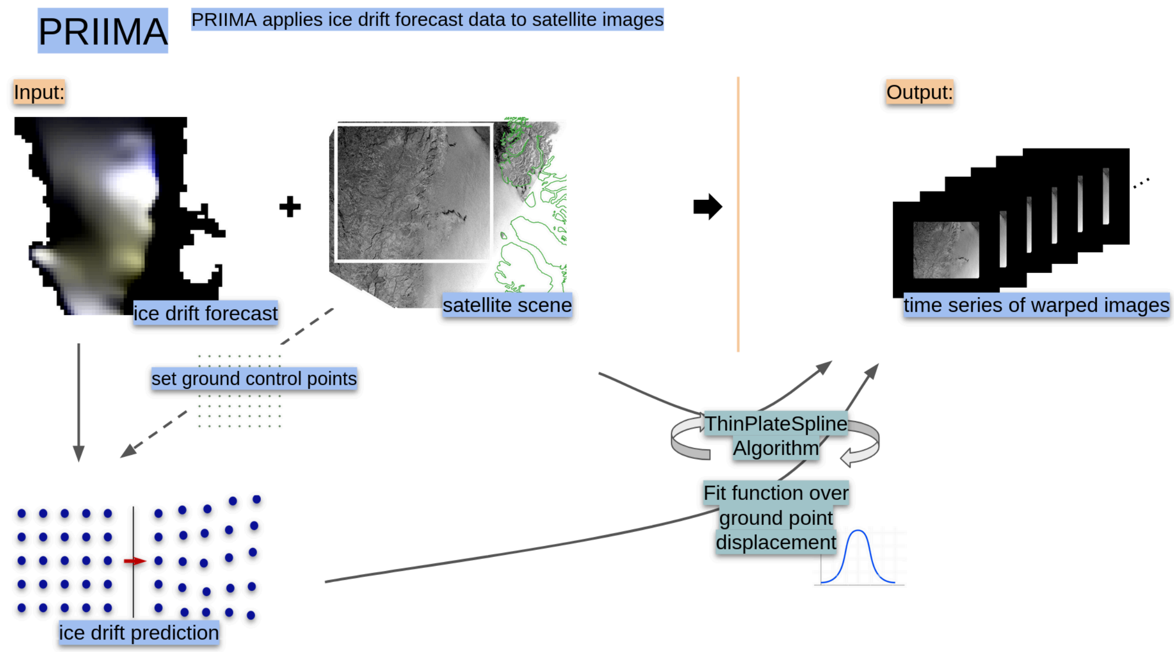


Figure D.1 — Polarwarp schematic overview

Input parameters include a region of interest defined by bounding box coordinates, the target date, forecast duration, and the number of ground control points (GCPs) used for the warping procedure. The workflow supports multiple GCP configurations, allowing users to balance computational cost and spatial accuracy. Input data sources include SAR data and model outputs accessed via authenticated services, while the workflow itself remains agnostic to the specific data provider as long as required projections and formats are satisfied.

The Polarwarp workflow is fully containerized and executed through an Argo-based orchestration layer. EOX acts as maintainer and executor of the workflow and operates it operationally on the EOxHub Workspaces platform. Within the OSPD project, EOX focused on adapting the workflow to remove platform-specific dependencies and ensure that execution is reproducible across heterogeneous infrastructures.

As a result, the workflow could be successfully reproduced and executed on multiple external platforms in addition to EOxHub Workspaces. Lessons learnt during those adoptions will continue to be explored also outside of the project scope.

The Polarwarp workflow was registered into Open Science Catalogue (OSC) as a workflow utilizing OGC API — Records (see below). Additionally to this, an Experiment (single run of the workflow with registered inputs) was recorded in the OSC and could be reproduced. Outputs of the algorithm are registered as a Product in OSC and they are utilizing STAC to describe all metadata.

Polarwarp was ultimately used as one of the 3 selected workflows for cross-platform reproducibility demonstrations within OSPD.

D.2.2.2. Documentation and Endpoints

- Workflow source and documentation:
  - https://github.com/gtif-cerulean/polarwarp

- OGC API Record (metadata) on Open Science Catalogue, for now on the staging deployment:
  - Workflow: https://osc-staging.earthcode.eox.at/workflows/polarwarp/record
  - Product: https://osc-staging.earthcode.eox.at/products/polarwarp/collection
  - Experiment: https://osc-staging.earthcode.eox.at/experiments/polarwarp/record

- Demonstration platform:
  - https://workspace.deside.hub-ovh.eox.at

D.3. Platform Operator: EOxHub Workspaces

D.3.1. Reproducible Execution Environment

EOxHub Workspaces was provided by EOX as a stable, multi-user execution environment supporting reproducible Earth Observation workflows. Within OSPD, EOxHub Workspaces offered:

- a JupyterLab-based interactive development environment;
- Argo Workflows;
- container-based execution to ensure consistent runtime behavior;
- access to Earth Observation data sources;
- persistent storage for workflow inputs and outputs; and
- shared access for multiple users and teams.

These capabilities enabled workflows to be executed, inspected, and re-run in a controlled environment, supporting reproducibility and collaborative validation.

D.3.2. Cross-Platform Reproducibility Support

Beyond executing workflows locally, EOX actively supported cross-platform reproducibility efforts. This included assisting workflow owners and platform operators with:

- debugging execution issues;
- adapting workflows and environments for portability;
- validating the outputs; and
- clarifying execution requirements and assumptions.

Through these activities, EOxHub Workspaces served both as an execution platform and as a reference environment for reproducibility testing for the Polarwarp algorithm.

D.3.3. Support to other workflows

In addition to POLARIS and Polarwarp, the Mangrove detection and water bodies workflows developed by other partners were successfully executed and reproduced on EOxHub Workspaces during the project.

EOX supported these efforts by preparing execution environments, granting platform access, and providing documentation and operational guidance. This support focused on enabling workflow authors to validate and reproduce their workflows rather than modifying the scientific logic of the workflows themselves.

D.4. Summary

EOX contributed to the OGC Open Science Persistent Demonstrator as both a Scientist and a Platform operator.

At the workflow level, EOX co-developed and executed the POLARIS workflow and maintained and operationally executed the Polarwarp workflow, which was successfully reproduced across multiple platforms. At the platform level, EOxHub Workspaces provided a reproducible, container-based execution environment and supported cross-platform validation and reuse of workflows developed by multiple partners.

Together, these contributions supported the OSPD objectives of reproducibility, portability, and open scientific collaboration.

E

ANNEX E (INFORMATIVE) KINDGROVE MANGROVE BIOMASS MONITORING WORKFLOW

E ANNEX E (INFORMATIVE) KINDGROVE MANGROVE BIOMASS MONITORING WORKFLOW

E.1. Why Mangroves Matter

Mangrove forests are among Earth's most carbon-dense ecosystems, sequestering 3-5 times more carbon per hectare than terrestrial forests while providing critical coastal protection and biodiversity habitat. Yet they are disappearing at 1-2% annually—faster than tropical rainforests. Satellite-based monitoring is essential for tracking loss, guiding restoration, and enabling carbon credit verification at scale.

The **KindGrove** workflow demonstrates how OGC Building Blocks can enable reproducible, standards-compliant mangrove monitoring using freely available Sentinel-2 imagery. The workflow detects mangrove extent via NDVI thresholds and estimates above-ground biomass and carbon stocks using IPCC Tier 2 methodology.

E.2. Workflow Architecture

The implementation uses a **Marimo reactive notebook** for interactive analysis and is packaged as a **Common Workflow Language (CWL)** application for automated execution on OSPD-compliant platforms, including CS Group's AntFlow (see Annex C).

E.2.1. Temporal Sampling Strategy

The workflow implements a 4-point temporal sampling strategy designed to capture change dynamics:

1. **Initial baseline** (2017): Reference state measurement;
2. **Pre-disturbance** (May 1-19, 2020): Immediately before Cyclone Amphan;

3. **Post-disturbance** (June 5+, 2020): Impact assessment after the cyclone; and
4. **Current state** (2024): Most recent condition.

E.3. Study Sites

Table E.1

SITE	LOCATION	SIZE	DESCRIPTION
Can Gio Biosphere Reserve	Vietnam	~75,000 ha	UNESCO Biosphere Reserve, good single-tile coverage
Sundarbans	Bangladesh/ India	~900 km² subset	World's largest mangrove forest, 30×30km analysis subset
Thor Heyerdahl Climate Park	Myanmar	1,800 ha	Active restoration site in Ayeyarwady Delta
Wunbaik Reserved Forest	Myanmar	~4,000 ha	Source of biomass calibration equation

E.4. Methodology

E.4.1. Mangrove Detection

Mangrove extent is identified using NDVI thresholds derived from peer-reviewed literature:

- **NDVI range**: `0.4 < NDVI < 0.95`;
- Lower bound (0.4) excludes water, bare soil, and sparse vegetation; and
- Upper bound (0.95) avoids sensor saturation artifacts.

E.4.2. Biomass Estimation

Above-ground biomass is computed using an allometric equation calibrated at Wunbaik Reserved Forest, Myanmar:

```
Biomass (Mg/ha) = 250.5 × NDVI - 75.2    (R² = 0.72)
```

Listing E.1

Carbon stock is derived using the IPCC 2006 default conversion factor:

```
Carbon (Mg C/ha) = Biomass × 0.47
```

Listing E.2

E.5. Scale-Independent Metrics

A key contribution of this workflow is the development of **coverage-normalized metrics** that enable fair temporal comparisons despite variable Sentinel-2 scene coverage.

Early in development, we observed what appeared to be a “3× increase in mangrove area” between time points—an exciting but implausible result. Investigation revealed this was entirely an artifact of different scene coverage: one scene covered 15% of our study area, another covered 45%. Absolute metrics (total area, total carbon) were meaningless for comparison.

The following scale-independent metrics address this:

Table E.2

METRIC	DEFINITION	PURPOSE
Mean Biomass (Mg/ha)	Average per hectare of observed mangrove	Scale-independent density measure
Mangrove Fraction (%)	Percentage of valid pixels classified as mangrove	Coverage-normalized extent
Carbon Density (Mg C/ha)	Carbon per hectare of mangrove area	Comparable across scenes
Scene Coverage (%)	Percentage of bounding box with valid data	Transparency metric

E.6. Findings and Limitations

E.6.1. What We Observed

Using the scale-independent metrics, we detected subtle changes across the temporal windows. However, the current methodology **barely registers** the impact of Cyclone Amphan—a Category 5 storm that devastated large sections of the Sundarbans in May 2020.

This counterintuitive result stems from two factors:

1. **Post-storm regrowth activity**: Mangroves show rapid vegetative response after disturbance, and our post-storm window (June 5+) already captures early recovery signals mixed with debris; and

2. **Seasonal noise**: Opportunistic date selection based on cloud cover means we cannot control for seasonal phenology, which introduces variation that masks disturbance signals.

E.6.2. What More Time Would Have Enabled

The OSPD pilot demonstrated proof-of-concept, but time constraints limited what could be achieved scientifically. With extended resources, we would have the following.

- **Utilized HPC resources** (e.g., I-Guide platform in tandem with CS Group's AntFlow) to split workload across execution environments. Once workflows are defined in the platform, coordinating calls via Dask is not a significant technical leap—this was initially planned but deprioritized due to time constraints.

- **Pulled more full-coverage data and more time points**: Rather than opportunistic "best available" scene selection, deliberate temporal sampling with sufficient scenes to composite cloud-free mosaics would genuinely enable new scientific capabilities—detecting disturbance events that are currently lost in the noise.

- **Controlled for seasonality**: Comparing same-month imagery across years would isolate disturbance signals from phenological variation.

The workflow architecture is ready for this scale-up; the limitation was pilot duration, not technical design.

E.7. OGC Building Blocks Integration

The workflow integrates several OGC-compliant specifications:

- **STAC API**: Data discovery via AWS Earth Search (`https://earth-search.aws.element84.com/v1`);
- **Bounding Box**: Standard OGC bbox format for spatial queries;
- **CWL Packaging**: OGC Application Package compliant workflow definition; and
- **AntFlow Integration**: Executed on CS Group's EOAP platform.

E.7.1. Ecosystem Adoption Friction

Early in the pilot, we experienced significant friction around ecosystem adoption—integrating our notebook-based workflow with the emerging Building Blocks infrastructure felt like swimming upstream. Documentation was sparse, patterns were still solidifying, and each participant was solving similar problems independently.

This friction began to dissipate noticeably as the OGC Rainbow and Building Blocks concepts gained clearer adoption strategies within the OSPD community. By the end of the pilot, the path from "working notebook" to "portable CWL package" was well-trodden, and RMIT's wf4ever provenance mapping (see OGC 25-044 RMIT annex) demonstrated that our workflow could be formally represented in standard ontologies.

E.8. Documentation and Endpoints

- **Workflow source**: https://github.com/starling-foundries/KindGrove
- **Demo notebook**: `mangrove_workflow_marimo.py`
- **Provenance mapping**: See RMIT Building Block annex in OGC 25-044 for wf4ever ontology representation

F

ANNEX F (NORMATIVE) TERRADUE SRL: WORKFLOW TRANSPILATION AND REUSE EXPERIMENTS

F

ANNEX F (NORMATIVE) TERRADUE SRL: WORKFLOW TRANSPILATION AND REUSE EXPERIMENTS

F.1. Overview

This annex describes Terradue's contribution to OSPD 2025 through experiments on workflow portability and reuse, with a particular focus on the translation of workflow descriptions across execution environments.

The work is implemented through the **transpiler-mate** tool, developed by Terradue to explore systematic approaches to workflow translation and metadata harmonization.

Further technical documentation and examples are available at:

- https://terradue.github.io/transpiler-mate/

F.2. Motivation and Context

Scientific workflows are often tightly coupled to specific execution frameworks and workflow languages. This coupling limits portability and complicates reuse across heterogeneous platforms participating in OSPD 2025.

The transpiler-mate activity investigates how workflows can be described using consistent metadata conventions that enable automated or semi-automated translation between execution models, while preserving semantic intent and configuration parameters.

F.3. Scope of the Experiment

The transpiler-mate experiments focus on:

- analyzing workflow structure and declared metadata;
- mapping workflow components to alternative execution representations; and
- translating workflows between CWL-based descriptions and other execution frameworks.

The scope is intentionally limited to selected workflow patterns and does not aim to provide complete semantic equivalence across all workflow languages.

F.4. CWL Metadata Conventions

A key aspect of the work is the definition of **CWL metadata conventions** that support workflow reuse and translation. These conventions define how descriptive metadata is embedded in CWL workflows to make implicit assumptions explicit and machine-interpretable.

The conventions address, among others:

- identification of workflows and workflow components;
- explicit declaration of inputs, outputs, and parameters;
- annotation of execution context and assumptions; and
- separation between workflow logic and execution-specific configuration.

These conventions enable transpiler-mate to reason about workflow structure and configuration independently of a specific execution backend.

The conventions are documented and maintained at:

- https://terradue.github.io/transpiler-mate/conventions/

F.5. Relation to OSPD Work Packages

The transpiler-mate activity contributes primarily to:

- **WP3 – Workflow Reuse Experiments**, by demonstrating practical mechanisms for reusing workflows across platforms; and
- **WP4 – OGC Workflow Building Blocks**, by providing feedback on the adequacy of workflow description patterns and parameter representations.

The work also reinforces the importance of clear semantic descriptions, as addressed in **WP2 – Open Science Ontology**.

F.6. Observations and Lessons Learned

Initial experiments indicate that workflow translation is feasible when workflows are described using explicit and consistent metadata conventions. At the same time, the experiments highlight several limitations:

- differences in execution semantics between workflow languages;
- incomplete provenance capture during translation; and
- the need for stronger constraints and profiles to support automation.

These findings are consistent with the broader observations reported in the main body of this Engineering Report.

F.7. Future Directions

Future work may explore tighter alignment between metadata conventions, ontology-driven descriptions, and OGC Workflow Building Blocks, as well as validation across a wider range of workflows and execution environments.