OGC® DOCUMENT: 24-022

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

Open Geospatial Consortium

OGC OPEN SCIENCE PERSISTENT DEMONSTRATOR (OSPD) REPORT

ENGINEERING REPORT

PUBLISHED

Submission Date: 2024-12-20
Approval Date: 2025-02-18
Publication Date: 2025-03-26
Editor: Pedro Gonçalves, Ingo Simonis, Micah Brachman

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

LIST OF FIGURES

I EXECUTIVE SUMMARY

The Open Science Persistent Demonstrator (OSPD) is a collaborative initiative led by the Open Geospatial Consortium (OGC), in partnership with the European Space Agency (ESA) and the National Aeronautics and Space Administration (NASA). Its primary goal is to advance open science by enabling reproducible Earth Science research across global communities.

Key Objectives:

- **Promote Open Science**: Facilitate broader access to Earth Observation (EO) data and tools, encouraging cross-disciplinary research and informed decision-making.
- **Enhance Interoperability**: Test and demonstrate the integration of diverse Earth Observation and Earth Science cloud technologies and infrastructures developed by ESA, NASA, and other international organizations.
- **Develop a Persistent Demonstrator**: Create a 24/7 web application that showcases scientific workflows across multiple platforms, demonstrating how different organizations' platforms can be utilized for collaborative research and data representation.

The Open Science Persistent Demonstrator (OSDP) project is a multi-year project, with just the first completed in late 2024. OSPD brought four essential elements together.

1. OGC Standards that enable interoperability between platforms, platforms and applications, and applications with data sets
2. A set of international Earth observation platforms, including
    a) CRIM provides federated platforms for climate data analysis with tools for geospatial data storage, visualization, and modeling using advanced interoperability standards.
    b) GeoLabs offers an open-source geospatial processing engine, ZOO-Project, supporting OGC standards for deploying workflows and integrating with high-performance computing resources.
    c) Terradue delivers on-demand processing services for satellite data applications, such as flood mapping, using OGC-compliant APIs and workflows.
    d) Ellipsis Drive, a scalable geospatial data storage and sharing platform that supports multiple OGC protocols for seamless dataset access and management.
    e) PolarTEP, a European Space Agency platform enabling remote data access, interactive development, and machine learning tools for polar science research.

f) Terrabyte (DLR), a high-performance data analytics platform hosted by the German Aerospace Center, offering extensive computational resources for Earth observation workflows.

g) Development Seed implemented a NASA VEDA (Visualization, Exploration, and Data Analysis) instance for EO data ingestion, visualization, and STAC catalog integration to support interoperable workflows.

h) Open Science Studio, facilitates the creation and sharing of APIs from Jupyter Notebooks, promoting reproducibility and accessibility in research workflows.

i) openEO / Google Earth Engine, provides a standardized interface to connect various EO data sources with the computational capabilities of Google Earth Engine for reproducible workflows

j) iGUIDE provided the CyberGIS-Compute framework which provides transparent access to HPC resources, while enabling users to manage containerized computational models and tools that can be configured and launched on HPC through an interactive interface.

3. The Galaxy Project, a free open-source web application that supports users to create, reuse, and publish scientific workflows for the analysis, integration, and visualization of data. Galaxy was used in the project for scientific workflow orchestration and execution.

4. The Open Science Framework, OSF is a free online research platform designed to support researchers to openly and transparently share their work at all stages of the research project lifecycle, and facilitate collaboration, documentation, archiving of work, together with the sharing of research projects, materials, and data. OSF was used in the project as on online registry for all OSPD platforms, workflows, and applications.

All material, background information, an introduction to collaborative open science in general, and lessons learned is available from the project website.

II KEYWORDS

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

open science, workflows, Earth observation, reusability, portability, transparency

III OVERVIEW

Addressing today's complex challenges requires Collaborative Open Science that enables integrity, provenance, and trust and fosters cross-domain integrations. However, building workflows and data flows that can operate across sectors remains technically challenging and resource-intensive, hindering whole-system change. Organizations in various sectors aspire to demonstrate accountability but often lack effective tools. The Open Geospatial Consortium (OGC) Open Science Persistent Demonstrator (OSPD) Pilot aims to promote collaborative open science. It facilitates responsible innovation linked to Earth Observation (EO) by simplifying the connection of data and platforms in transparent, portable, reproducible, standards-conformant workflows.

The OSPD promotes open science by embodying principles of reusability, portability, and transparency. Reusability involves consistent utilization of EO data and workflows, together with sector-specific data, across platforms to maximize efficiency. Portability ensures seamless transition of EO data and applications across platforms for broader applicability. Transparency provides clear insights into EO data processing, building trust within the community. Standards enable interoperability and quality assurance. Key work in the development of the OSPD includes implementing OGC standards for diverse platforms to make geospatial data and services available on a workflow platform, utilizing templates to document services and workflows, and providing linked learning materials for user accessibility. The OSPD is built around four key components: a community of geospatial computing platforms that utilize OGC Standards-conformant data and services; Galaxy, which enables users to build and test cross-platform workflows; The Open Science Framework (OSF), which serves as an archive and discoverability hub; and OGC Standards such as OGC API-Processes to enable cross-platform interoperability.

IV FUTURE OUTLOOK

Important planned features of the OSPD support essential open science actions such as assigning persistent identifiers, writing documentation, implementing version control, standardizing data formats and APIs, transferring licenses and metadata, and ensuring cross-platform portability and testing. Use cases developed for the OSPD demonstrate how it supports users in composing and documenting open science workflows across multiple platforms and illustrate the value of open science, not only for research, but for all organizations that need to provide transparent data-driven decision making or serve in positions of public trust.

V VALUE PROPOSITION

The OSPD generates value for organizations that provide geospatial data and services by improving their discoverability and usability. Organizations will have greater impacts with their data and applications, because OSPD enhances the visibility of all elements by connecting data with descriptions, platforms with services, and users with applications. It provides value to science and research users by facilitating discovery of open standards-based geospatial tools and services, supporting them to maintain data integrity and provenance when reusing them in novel workflows and applications, and accelerating research by enabling reuse of well-documented standards-based tools.

1

THE OSPD INITIATIVE

1 THE OSPD INITIATIVE

The OGC Open Science Persistent Demonstrator (OSPD) initiative will be carried out over several years as part of the OGC Collaborative Solutions and Innovation Program. This report discusses the results of the first phase. To ensure the most sustainable development possible, some parts, for example the Clause 2.2, are defined across phases. In the first phase, they primarily serve to provide orientation and a framework and will be fully implemented in the further course of the initiative.

2 INTRODUCTION

2 INTRODUCTION

2.1. Aims

Collaborative open science is essential to addressing complex challenges whose solutions prioritize integrity, provenance, and trust, and require cross-domain integrations. Today, building workflows, processes, and data flows across domains and sectors remains technically difficult and practically resource intensive, creating barriers to whole-systems change. While organizations in the public and private sector, as well as in research, increasingly aim to demonstrate accountability, they often lack the tools to act effectively. The Open Geospatial Consortium (OGC) Open Science Persistent Demonstrator (OSPD) aims to promote collaborative open science and enable responsible innovation linked to Earth Observation (EO) by making it simple to connect data and platforms together in transparent, reusable and reproducible workflows.

The OSPD's design enables reproducible open science by creating practical mechanisms to put the principles of reusability, portability, and transparency into action.

- **Reusability**, as envisioned by the OSPD, involves consistent utilization of EO data, processes, and scientific workflows across diverse platforms and sectors. By embedding this principle in tools, the OSPD aims to maximize the value derived from each segment of EO data, thereby promoting efficiency.
- **Portability** underscores the need for EO data, applications, and insights to transition seamlessly across various platforms, ensuring broader applicability and flexibility.
- **Transparency** facilitates a clear view into the mechanisms of EO data processing and use. By offering a transparent system, the OSPD endeavors to foster trust and provide clarity for its users, stakeholders, and the broader EO community.

This report provides an overview of the principles and values guiding the development of the OSPD, the initial design of the OSPD, lessons learned from early design and prototyping experiments, and plans for the next phase of OSPD development.

2.2. Scenarios

The OSPD Pilot series is designed foremost as a tool for technical professionals who are developing tools and systems to support scenarios where integrity, provenance, trust, and cross-domain integrations are required. Two example scenarios are provided below to illustrate situations in which a technical user would be developing a tool, model, or system to meet these

requirements. The scenarios will be implemented and refined step by step during the current and future phases of the OSPD Pilot.

2.2.1. Scenario 1 — Water quality degradation due to harmful algal blooms

- **Problem Overview**

  Harmful algal blooms (HABs) refer to the overgrowth of algal species, which can have harmful effects on human and animal populations . HABs occur naturally and are not always harmful and the degree of harm and damage to humans or animals is caused by the type of species involved in the bloom and the type of toxins released (Gobler).

  Human activity is contributing to more algal blooms throughout the planet, mainly through the pollution of waterways with nitrogenous and phosphate-rich waste from agricultural and industrial activity, inadequate wastewater treatment and road runoff (Guo et al). Warmer weather associated with global climate change is also contributing to the greater frequency of algal blooms worldwide (Gobler et al).

  When harmful, the production of toxins by cyanobacteria in freshwater and brackish water systems and dinoflagellates and diatoms in marine water systems can lead to dead zones killing off all flora and fauna (NIH). On humans, these toxins can lead to neurological, gastrointestinal, and respiratory effects as well as affect skin and tissue. HABs also have economic implications as fisheries, wildlife and recreational activities, and tourism are impacted.

- **User Personas**

  - *Health Departments*: As a state or local health agency, I want continuous monitoring of cyanobacteria levels in bodies of water within my area jurisdiction because both sampling (expensive, labor-dependent) and community reporting are inconsistent and unsustainable.

  - *Federal Health Authorities*: Federal health authorities and agency personnel will want continuous monitoring of bodies of water across the U.S. so that I can provide this as a service to state health authorities. Government agencies such as NASA, the ESA, NOAA, USGS, HHS, the EPA and/or others may wish to task satellites to provide continuous monitoring of cyanobacteria levels in bodies of water of particular interest on behalf of state and local health authorities who may not be able to afford to do so.

  - *Health Systems*: As a health system/hospital, I want to know about the presence (location, type, extent) of an algal bloom so I can prepare as possible to provide care to the infected.

  - *Water Companies*: As a water company, I want to know about algal blooms along with any issues that may impact any body of water I may be using as a source for providing drinking water.

  - *Fisheries industry*: As a Fishery, I want to know about algal blooms along with any issues that may impact any body of water I may be using as a source of seafood.

- *The Public*: As a member of the public, I want to know how scientists and policy makers understand and make decisions about algal blooms and determine which bodies of water are potentially unsafe, and the impacts in my community.

- **Open Science Requirements**
  - **MUST**:
    - Correctly identify current algal blooms over 90% of the time (90% accuracy)
    - Consistently identify algal blooms given the same parameters (99% consistency)
      - Provide transparency in methodology used
        - Technical details
        - Scientific rationale
      - Contain a mechanism for ingesting feedback from expert users
      - Contain a mechanism for incorporating useful feedback into system updates
  - **SHOULD**:
    - Exhibit enough repeatability and reproducibility to be a trustworthy tool for public health experts
    - Be timely in its output
    - Provide output that is useful for intervention
    - Have a user interface that requires minimal onboarding and training for tasks such as finding and reproducing workflows
    - Have a platform that is sustainable by non-experts over a long term
    - Have a structure for updates and review of data in a cyclical and known time frame
  - **COULD**:
    - Contain links and open resources related to the material for further education
    - Be combined with other monitoring platforms and tools by expert users, taking OSPD workflow outputs into external analytical tools
    - Contain data that can be downloaded by expert users
    - Exist as both a desktop version and a mobile app version
  - **WOULD**:

- Would like to have the platform be visually engaging and user friendly
- Would like to have the platform have either a version that is user friendly towards the lay public or sections specifically designed for the lay public
- Would like to have the input of the lay public as well
    - Ex: swimmer's clubs, habitual beach users, commercial fishing
- Would like to advertise our work in the scientific literature and scientific spaces

2.2.2. Scenario 2 — Water quality degradation due to floods and droughts

- **Problem Overview**

  Natural disasters like floods and droughts severely impact water sources, leading to significant challenges in water consumption and safety. Floods can contaminate water with pathogens, pesticides, pollutants from factories, and heavy metals by overwhelming treatment facilities and inundating agricultural lands, while droughts reduce water availability, concentrating pollutants and exacerbating competition for scarce resources. These events compromise the quality and availability of drinking water, posing risks of waterborne diseases such as diarrhea, cholera, and hepatitis, alongside long-term health effects from chemical contaminants. The socioeconomic implications of these disasters are profound, amplifying inequalities in water access, particularly affecting marginalized communities. Recovery efforts are often costly and lengthy, exacerbating water scarcity issues. Addressing these challenges necessitates resilient water management strategies, infrastructure improvements, and community-focused initiatives to ensure equitable access to clean and safe water for at-risk aged groups.

- **User personas**
    - *Emergency Response*: As a representative of the emergency response community, I want to know the risks to drinking water as both access to water is important in a disaster (e.g., fire) as well as protecting water from contamination.
    - *Health Departments*: As a state or local health agency, I want to know if any water resource is compromised so that we can issue the necessary alerts (e.g., boil advisory) and begin remediation.
    - *Health Systems*: As a health system/hospital, I want to know about unsafe drinking water so I can prepare as possible to provide care to those who have consumed unsafe water.
    - *Water Companies*: As a water company, I want to know about unsafe drinking water so we can issue the necessary alerts (e.g., boil advisory) and begin remediation.
    - *The Public*: As a member of the public, I want to know how scientists and policy makers evaluate and make decisions about the status of my drinking water, both to be able to

understand how their actions are designed to protect my health, and impacts of this in my community.

- **Open Science Requirements**
  - **MUST**:
    - Correctly identify current contaminated water reservoirs over 90% of the time (90% accuracy)
    - Consistently identify contaminated water given the same parameters (99% consistency)
    - Provide transparency in methodology used
      - Technical details
      - Scientific rationale
    - Contain a mechanism for ingesting feedback from expert users
    - Contain a mechanism for incorporating useful feedback into system updates
  - **SHOULD**:
    - Exhibit enough repeatability and reproducibility to be a trustworthy tool for public health experts.
    - Be timely in its output
    - Provide output that is useful for intervention
    - Have a user interface that requires minimal onboarding and training
    - Have a platform that is sustainable by non-experts over a long term
    - Have a structure for updates and review of data in a cyclical and known time frame
  - **COULD**:
    - Contain links and resources related to the material for further education
    - Be paired with other monitoring platforms and tools by expert users
    - Contain data that can be downloaded by expert users
    - Exist as both a desktop version and a mobile app version
  - **WOULD**:

- Would like to correctly identify current contaminated water reservoirs over 100% of the time (100% accuracy)
- Would like to have the platform be visually engaging and user friendly
- Would like to have the platform have either a version that is user friendly towards the lay public or sections specifically designed for the lay public
- Would like to have the input of the lay public as well
- Would like to advertise our work in the scientific literature and scientific spaces

2.3. Technical Requirements

Technical users of the OSPD might include the staff of small businesses working on contracts with public agencies, data engineers or analysts in public agencies or non-profits working in the public interest, or university researchers engaged in use-inspired or applied projects. For technical users like these to implement tools that can address scenarios like these, the OSPD needs to address the following requirements.

- I need to discover algorithms that can be used, alone or in combination with other algorithms, to address my problem.
- I need to discover scientific literature that provides evidence that the algorithms are validated (peer reviewed) and robust.
- I need to discover platforms where these algorithms can be run using the data I need to address my problem.
- I need practical instructions on how I can reuse the algorithms I found to address my problems.
- I need to be able to discover and use specific implementations (versions) of the selected algorithms.
- I need to design and build a workflow that uses a combination of multiple algorithms to address my problem.
- I need to reuse algorithms on their respective platforms and to be able to change paramenters in them.
- I need to combine several algorithms on the same platform into a workflow by using the output of one algorithm as the input to another.
- I need to combine several algorithms on different platforms into a workflow by using the output of one algorithm as the input to another.
- I need to take the results of a workflow and use them in subsequent analyses.

- I need to share my workflow and analysis so that someone else can reuse them with confidence.
- I need the workflows I build to be portable to other environments, including my own organziation's infrastructure.
- I need to run an experiment on a different platform where I have the necessary credits and credentials to execute a workflow.

2.4. Objectives

To achieve its aims, the OSPD is developing a distributed cyberinfrastructure with four key components. **OGC standards (1)** (OGC API-Processes and -Features) are implemented to enable **diverse platforms (2)** to make their geospatial data, processing and analytical services available on a workflow building and execution platform, **Galaxy (3)**, so that researchers can easily **reuse** and **remix** them in their own workflows and **port** them to new platforms. Templates are implemented in **osf.io (4)**, a domain-agnostic trusted digital repository, to guide providers of data and analytical services and creators of Galaxy workflows to document their services, workflows, and outputs, so that they are more **findable** and **transparent**. The OSPD is creating linked learning and outreach materials that make participating platforms visible and accessible to a wide range of users. **Use cases** demonstrate how the OSPD supports its users to compose and document open science workflows leveraging diverse EO platforms.

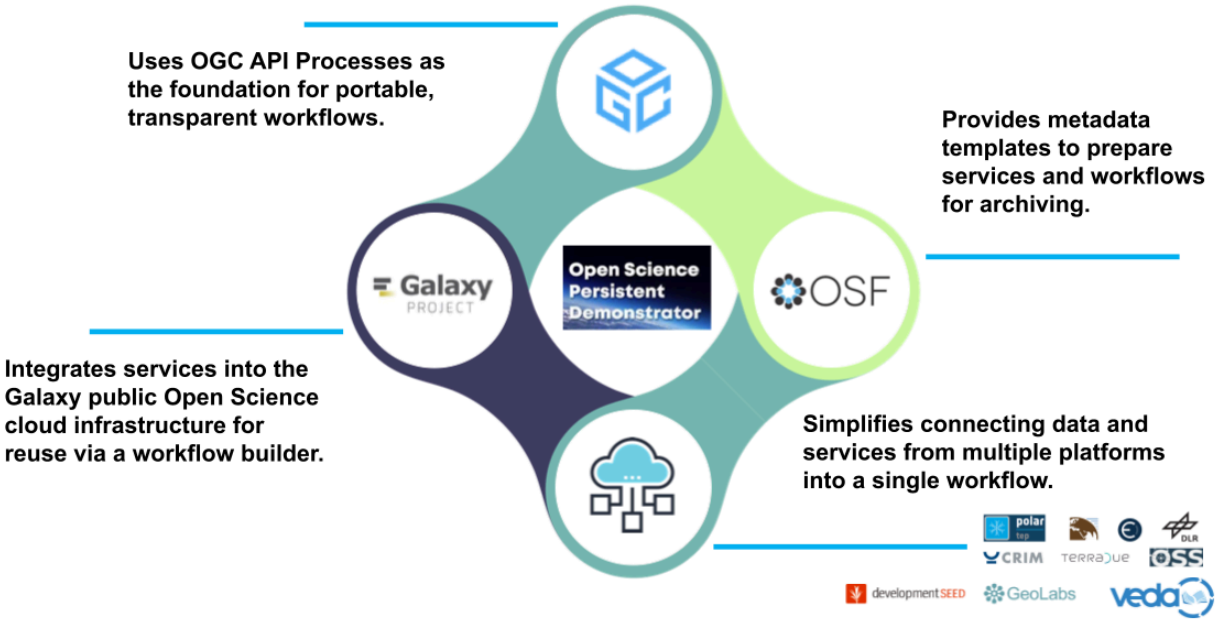

Figure 1 — Figure The Open Science Persistent Demonstrator as a place to discover, prototype, share, and archive geospatial services and workflows

The currently implemented and future planned features of the OSPD support essential open science actions, including the following.

- **Assigning DOIs or UUIDs**: Persistent identifiers like Digital Object Identifiers (DOIs) ensures entities are persistently accessible and citable, embedding them within the broader discourse of linking data-code-documentation and cross-indexing.
- **Writing Documentation and Intended Application Good Practices**: Describing data and application workflows using well-documented practices, metadata, and templates facilitates anyone, regardless of prior knowledge, in engaging with a dataset or workflow and aids those looking to extend or adapt work for new contexts.
- **Implementing Version Control Systems or Containerization**: These technologies enable tracking and recall of versions of algorithms, parameters, and configurations.
- **Implementing Standardized Data Formats and APIs**: These formats and tools ensure platforms interact seamlessly, promoting effective interoperability.
- **Transferring licenses and metadata**: Collection and passing all metadata, data and code licenses used between platforms following a consistent structure enables reuse.
- **Cross-Platform Portability and Testing**: Regular testing across platforms ensures that inconsistencies are identified and addressed and results are consistent.

2.5. User Journeys

Based on the use cases above, two key user groups were identified who would engage with the OSPD. The first user group is platform providers, represented below by the user journey of a developer named Grace. Scientists and researchers comprise the second user group, and are represented by Lila's user journey. The training materials developed for this project are designed to support each group of users on their journey through the OSPD.

The following user journey describes Grace, a developer at CODEXYZ Ltd, who wants to create an open science workflow within the OSPD demonstrator ecosystem for scientists and researchers to use freely. CODEXYZ has built their own in-house processing platform, and uses this to provide commercial products and services to their clients. The algorithms, data, and processing methodologies are considered CODEXYZ's intellectual property, but Grace is able to use the OSPD to create a freely accessible workflow that is made interoperable through implementation of OGC API — Processes.

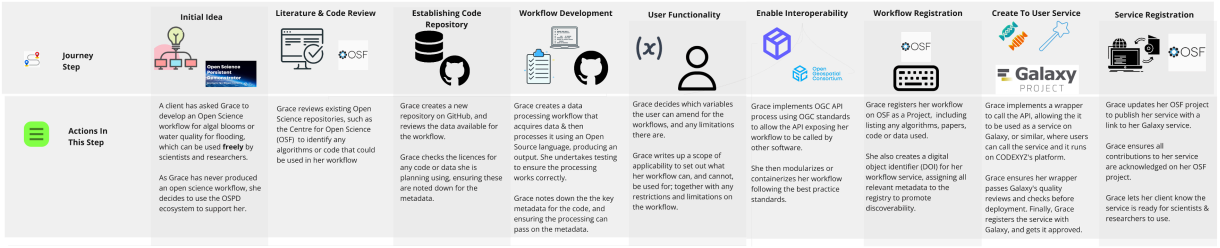

Figure 2 — Platform Provider User Journey

The next user journey describes Lila, a scientist who wants to conduct research focused on the health impacts of climate change. Lila doesn't have any programming experience and has limited access to geospatial data processing infrastructure. Despite these limitations, she is able to leverage the OSPD to identify a research project, find and analyze relevant geospatial datasets, and write a successful application for research funding.

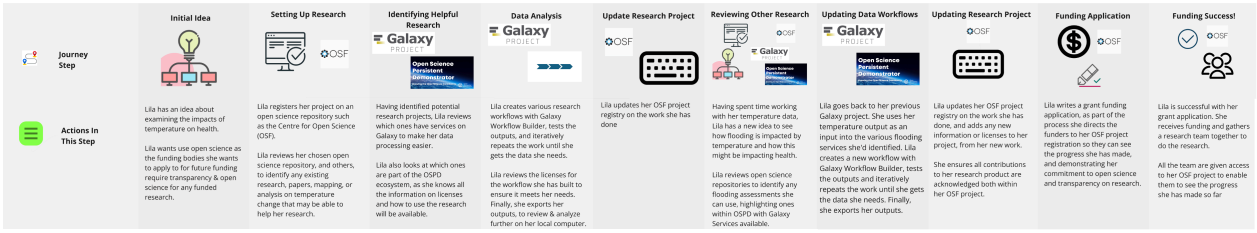

Figure 3 — Scientist User Journey

3

TECHNICAL FEATURES

3 TECHNICAL FEATURES

3.1. Discoverability and archiving

The OSPD builds on the discoverability and archiving capabilities of the OSF.io platform, which are complemented by the workflow building capabilities of Galaxy. The OSF.io platform supports discoverability both by encouraging creation of robust metadata through registration templates, which prompt contributors to add tags, and through its faceted search. OSF.io is a Trusted Digital Repository. An OSF Registry provides a permanent, transparent, easily accessible repository that enables the archiving, sharing, searching, and aggregating of funded study plans, designs, data, and outcomes. Researchers can create robust, timestamped registrations of research projects

Metadata plays a crucial role in enhancing the discoverability of these materials both on and off the platform. By providing detailed metadata during the registration process, researchers improve the visibility of their work to other scholars, institutions, and the broader academic community. Rich metadata, including title, contributors or authors, keywords, subjects, and licensing information, allows for more accurate and efficient searching and browsing, enabling researchers to locate relevant materials more effectively. After the registration is archived, researchers have the flexibility to append additional metadata such as the type of information included in the registration, the language in which it is written, details about the funding agency, award title, award number, and URI. Moreover, researchers can seamlessly link other pertinent research materials, whether they reside on or off the OSF.io platform, to the registration via a DOI.

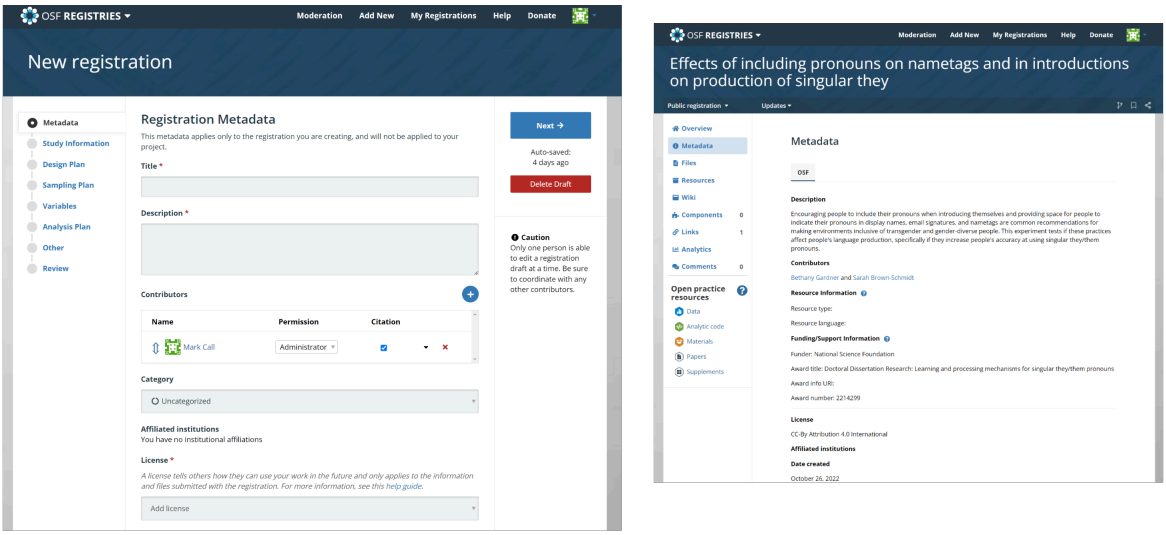

Figure 4 — Registrations are the basic archival entity in the OSF system. The OSPD will develop customized templates to capture consistent metadata about platforms, services, workflows and workflow instances.

Metadata serves as the backbone of discoverability within OSF's search page. By leveraging the rich metadata associated with research materials, OSF's search page empowers researchers to pinpoint relevant information efficiently. Through the creation of filters based on the metadata mentioned earlier, users can fine-tune their search queries to align with their specific research interests and objectives. This granular level of filtering not only narrows down search results but also ensures that researchers are presented with highly relevant and contextually appropriate materials.

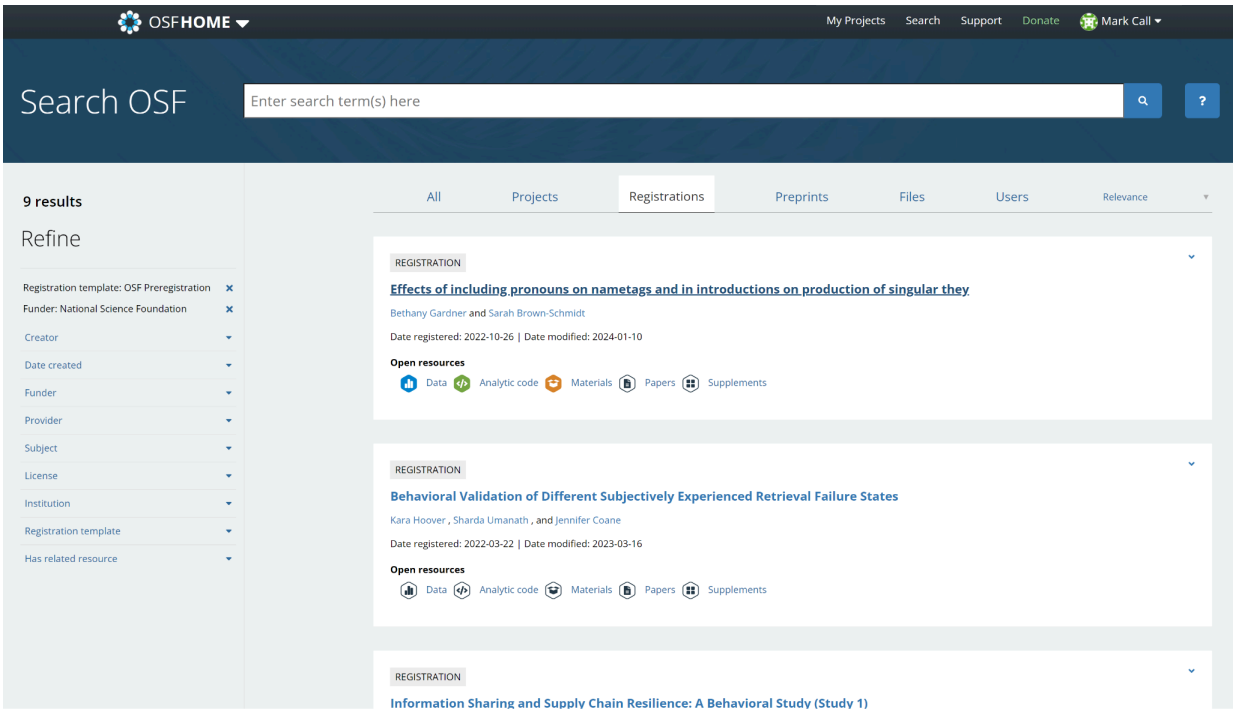

Figure 5 — The OSF supports the discovery of registrations and other entities based on their metadata both through free text and faceted search.

The OSF is actively restructuring their integrations platform to allow for a more robust set of integrations led by research communities. The CEDAR (Center for Expanded Data Annotation and Retrieval) Workbench is one of several upcoming integrations that expands metadata further to include key information that's important for different research communities. The restructure will make other integrations, such as that with Galaxy and others, easier.

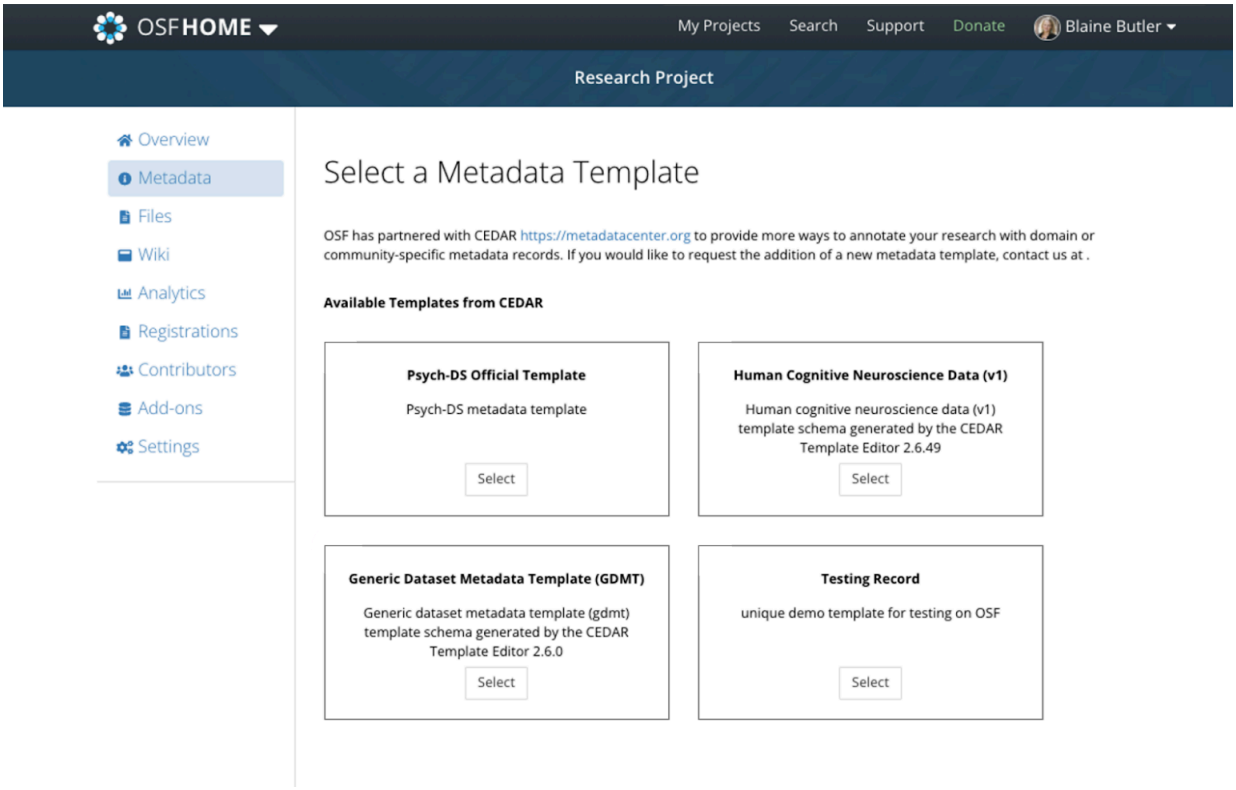

Figure 6 — The design of metadata templates in the OSF for the OSPD will be based on input from both platform providers and scientist end users involved in the development of the OSPD.

Metadata also serves as a bridge between the OSF.io platform and other scholarly resources by making information accessible through Application Programming Interfaces (APIs). APIs allow external platforms and services to interact with OSF.io data, enabling seamless integration and interoperability with a wide range of research tools and systems. By exposing metadata through APIs, OSF.io promotes data sharing and facilitates the exchange of research information across diverse platforms, ultimately enhancing collaboration and accelerating scientific progress.

OSPD users will be expected to register four types of entities in the osf.io system: **platforms, services, workflows and workflow instances**. Platforms are independent providers of geospatial data, search, processing, analytical or modeling services who can connect with the OSPD through use of OGC Standards conformant APIs and formats. Services are the specific data, search, processing, analytical or modeling capabilities provided by platforms to the OSPD using OGC Standards conformant APIs and formats. Workflows are the templates for operations that use one or more than one component (where a component is a service or data) to accomplish a task such as the execution of a multi-step scientific analysis or modeling of a scenario. Workflow instances are executed workflows where specific data, services and parameters have been used to generate an output.

Platform providers who are contributing services (data, search, analysis) to the OSPD will be expected to document their platform as an entity, enabling OSPD users to discover multiple services provided on a single platform and see the set of platforms in the OSPD ecosystem. Platform providers will also be expected to document the specific services they provide, which

will be linked from both platforms and workflows. Scientist users of the OSPD will be expected to document workflows they build on Galaxy to enable their reuse by others and their discovery. Scientist users will also be able to document and archive workflow instances — runs of a specific workflow using specific platforms and data — on osf.io, enabling transparency. The templates specifying the required metadata and documentation for each entity are being developed during the current phase of the OSPD Pilot.

3.2. Workflow building, testing and execution features

To meet the use cases' requirements, the OSPD selected Galaxy. Galaxy provides a toolset for building, testing, and executing workflows that combine data and analytical tools. These tools may be provided by any platform with services conformant to OGC API Processes. The Galaxy platform is a core component of the OSPD architecture, acting as a hub for running and orchestrating workflows composed of processes hosted on one or several independent platforms.

3.2.1. Key Features

Galaxy is a flexible and extensible open source web application to assist researchers in publishing and reusing reproducible workflows. The platform offers users a large number of tools to accomplish specific tasks, such as data manipulation (e.g., adding a column to an existing dataset), data analysis (e.g., running a statistical computation), or data visualization (e.g., creating figures). These tools can be combined into readily sharable, well-documented workflows. Galaxy is widely used in the research community, with over 50,000 users from over 100 countries as of 2023. Because Galaxy tools and OGC API Processes both use an input-processing-output logic for building workflows, integration of OGC API Processes into the Galaxy platform can enable the combined use of standards-conformant geospatial data and analytical tools and leverage those contributed by other researchers using Galaxy.

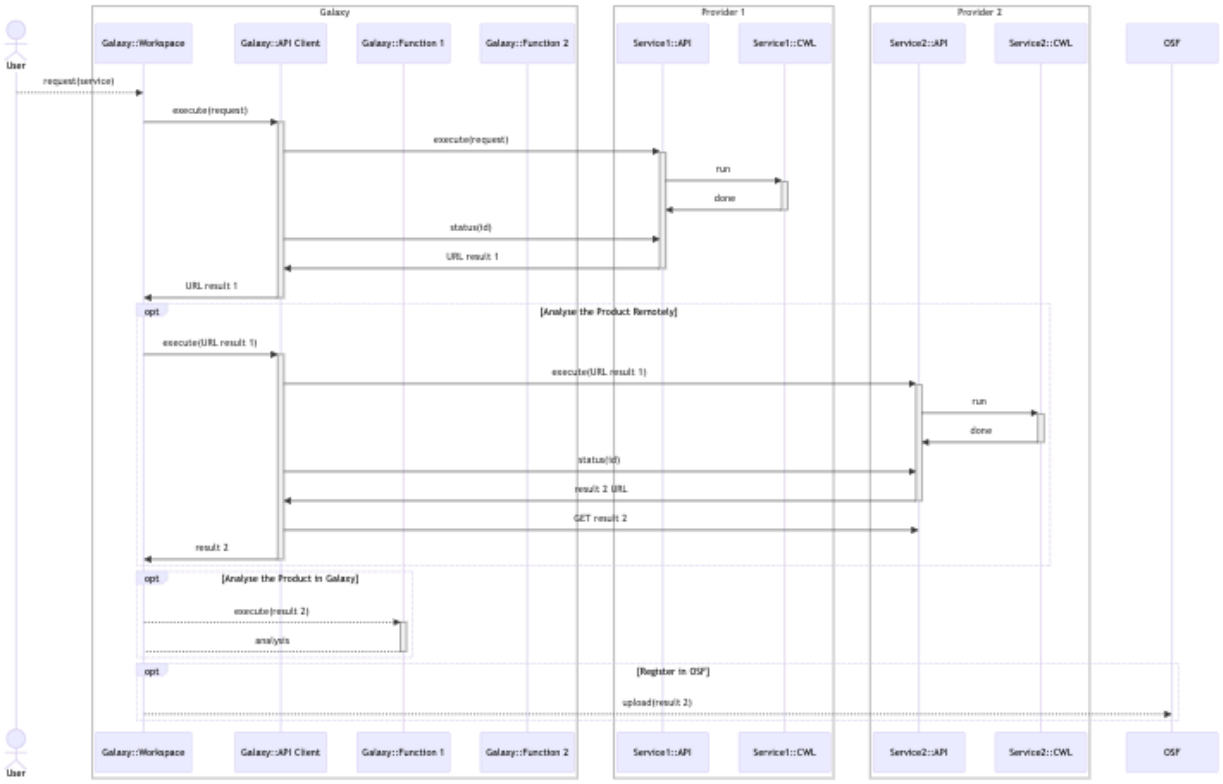

Figure 7 — Multiple services on different platforms using OGC API Processes are integrated into a single workflow on Galaxy, as shown in the high-level schematic.

The challenge of the OSPD is to explore **how to develop the necessary integrations to maximize usability and value for platforms providing data and analytical and modeling tools and researchers using Galaxy** to build and execute workflows for open science applications.

Through discussions with the Galaxy community, Galaxy's core developers, and the partners involved in the OSPD project, the following key requirements for integrating OGC API Processes were identified.

- The services provided by partner platforms should still run on their infrastructure.
- Since geodata can be large, data transfer to and from Galaxy should be avoided.
- Users should be able to configure the input parameters via the Galaxy user interface.
- Users should be able to connect services in Galaxy to create workflows.

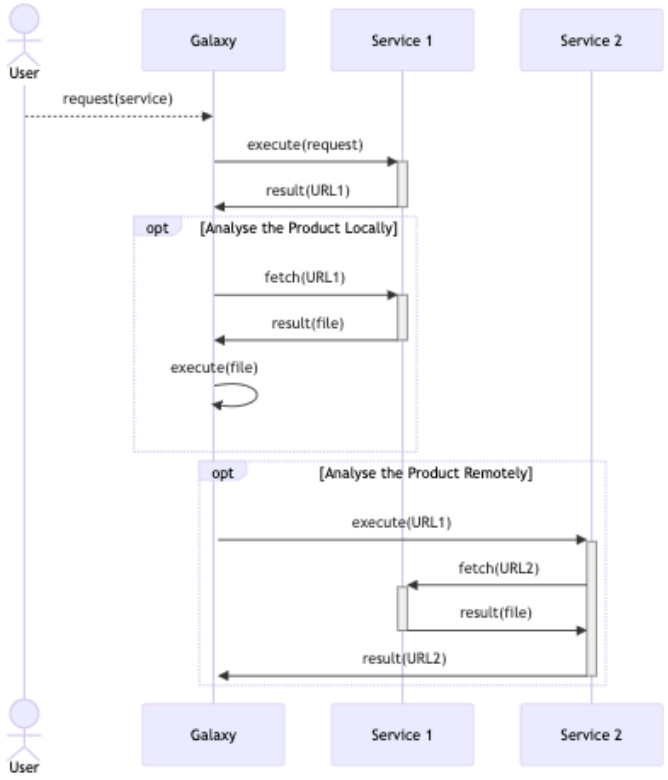

Figure 8 — Results are passed as files containing URLs to enable data to remain at rest and keep processing on partner platforms' compute infrastructures.

Based on these requirements, several integration options were outlined. The Wrapper option described below stands out for its capability to meet the requirements, and was preferred by the OSPD platform providers, representing a key stakeholder community. The strengths, weaknesses, opportunities, and threats of use of the Wrapper option were explored in detail, described below, together with brief discussion of alternative options.

3.2.2. Integration options

3.2.2.1. The Wrapper option

The idea of the Wrapper option is to wrap the OGC API Processes in a Galaxy tool and enable a tool-service communication flow as described in Figure 1X. The tool (Galaxy service wrapper) runs on the Galaxy platform and collects all required input parameters from the user. The tool then passes the parameters to the process and executes it via its API. Through the job ID that was received from the process, the tool requests the job status in regular intervals and fetches the result once the process finishes successfully. An example in-development wrapper

integration is available on GitHub in a Galaxy fork[1]. We anticipate that most platforms will pursue this option to integrate with Galaxy.

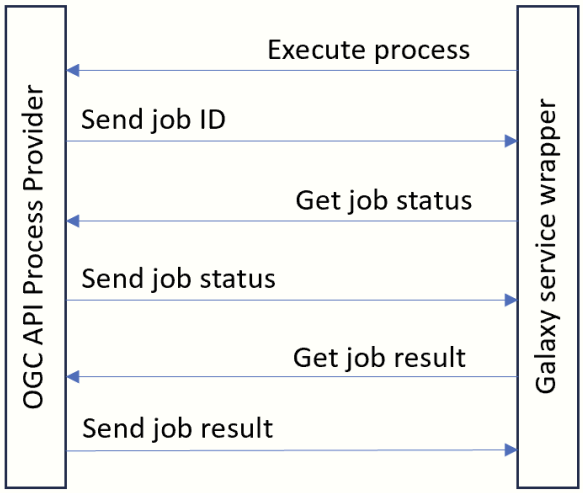

Figure 9 — Communication flow between the wrapper and the OGC API Process.

3.2.2.2. Further integration options

Re-implementation: Another option to integrate an OGC API Processes in Galaxy as a tool is to re-implement it. The advantage of this option is it does not require a remote server, relevant if a server is not available or could be shut down, or if maintaining an independent external platform is beyond the capacity or needs of the organization that created the tool. This option can also be used to check the tool integration process for robustness. Disadvantages of this option are that re-implementing a process in Galaxy is time-consuming and not useful if service providers want to see their own infrastructure in use, and that transferring larger datasets to Galaxy might be necessary. We anticipate that this option would be pursued only by organizations not in a position to maintain their own platform and that, in practice, it will be rarely used for the OSPD.

Containerization: The final option is to integrate the Application Package directly in Galaxy. The Application Package includes the service containerized application and all the necessary service's metadata. This might provide a useful and easy-to-maintain option in cases where it is already containerized, but otherwise this option requires additional effort to create a dockerfile. We anticipate that some platforms may choose to pursue this option in future phases of the OSPD.

3.2.3. SWOT Analysis

Strengths: The Wrapper option comes with a number of benefits. First, the service is executed in Galaxy but the computations run on the provider's infrastructure. This is particularly important if the OSPD infrastructure is based on a shared Galaxy instance (e.g., the European Galaxy server)

[1] https://github.com/AquaINFRA/galaxy/tree/ogc_process_otb_bandmath

which has limited resources because they are managed centrally and allocated across multiple projects and users. The providers have more control over the server hosting the process and can add further resources if needed. Second, to avoid heavy data transfer (e.g., in the case of big data), in the example implementation of the Wrapper option the URL to the result is stored in a .txt file, which can be used as an input to another wrapper in a workflow. Since we only need to send a list of URLs linking to the input datasets from one process to the other, a .txt is a suitable data format. Finally, the wrapper is a lightweight implementation and the service remains untouched.

Weaknesses: A limitation of the Wrapper option is its scalability to several hundreds or even thousands of services. While it is possible to create a template that simply needs to be completed with the corresponding process information resulting in one Galaxy tool per process, there is still too much manual work required. Notably, beyond completing the template, one would also need to onboard every tool (i.e., the wrapper of one service) to Galaxy, which requires a set of pull requests and human review by Galaxy developers which cannot be automated. To mitigate this issue, we implemented a generalized solution, the OGCProcess2Galaxy tool, to wrap a set of processes into one Galaxy tool. For each server, the tool fetches all processes via GetCapabilities from the OGC API Processes Provider (see Figure 2). If not all processes are needed, it is possible to indicate which processes should be included or excluded using a configuration file.

Then, for each process, all necessary information including metadata, inputs, and outputs can be requested via ProcessDescription to build the Galaxy tool. First tests based on the servers provided in the OSPD project showed that the tool can generate a template which requires additional manual work in relation to the variable and parameter attributes, and the communication processes between Galaxy and the workflow. This is because although the processes are standardized they can be implemented in different ways and may have aspects (edge cases) which are difficult to implement generically. The final tool will be reviewed by the Galaxy community for security and operational checks and, if the review is successful, made publicly available. Updates in the list of servers would require an update of the Galaxy tool. However, such minor updates do not require the same review process as it is needed for a new tool.

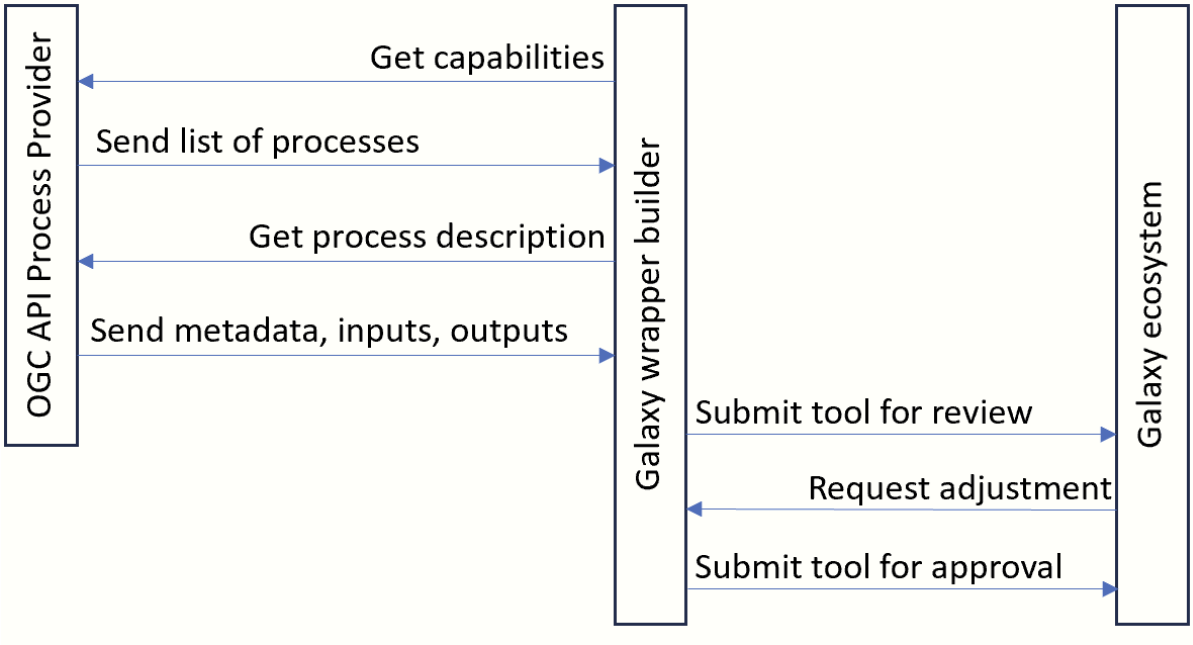

Figure 10 — Figure Description of how to create a generic wrapper.

Opportunities: The Galaxy platform originated in the life sciences community but the Geo user community on Galaxy is growing. The integration of OGC API Processes in Galaxy can increase the awareness of OGC APIs and standards within and beyond the Geo domain. A successful integration can demonstrate the interoperability of OGC API Processes across platforms and, critically, across scientific domains. Moreover, users who are unfamiliar with OGC API Processes or do not have the skills to work with the API receive a ready-to-use application as an entry point to OGC API Processes.

Threats: One risk is the frequently occurring issue of maintenance and the question of who has responsibility for maintaining different elements of the OSPD. Updates in the OGC API can break the Galaxy tool and make it unusable until it receives an update. However, this issue is not specific to the OGC API — Galaxy integration but a general problem in software development. Furthermore, it is debatable whether it is better to have a "stable" tool where changes in the service do not result in a broken but potentially irreproducible tool, or to have a tool that breaks and, hence, does not provide potentially irreproducible results.

The generalized approach has some risks. Notably, it may prove difficult to spot incorrectly implemented edge cases in an operational tool. While creating tests in a Galaxy tool similar to unit tests might address common errors, some issues in a certain process will only become visible upon usage. Another potential risk comes from the quality of the descriptions of the OGC API Processes. A generic tool will strongly rely on high quality process descriptions, which must be created and maintained by platform and service providers.

3.2.4. Open Design questions

How to apply the Wrapper option to OGC API Features?

While further research is needed to investigate how OGC API Features can be integrated in Galaxy, it is broadly expected that the challenges of integration of further OGC APIs will reflect those encountered in integrating OGC API Processes. Testing of an integration workflow for OGC API Features will commence in the second part of the current phase of the OSPD.

How to inform users about service updates?

Updates in the services may or may not break the Wrapper tool. Regardless, to meet the OSPD's Reproducibility requirement, users need to be made aware of any updates in the services and these changes must be reflected in the services OSF.io documentation. Options for displaying and transmitting this information require further exploration.

3.3. Platforms Overview

The third part of the OSPD infrastructure is the community of platforms contributing OGC standards conformant services to the OSPD. Details on each platform's contributions and work are summarized in the OSPD Community Platforms — Detail section.

Platforms play a key role in the OSPD by providing the data, data cataloging, processing, analysis, and modeling services as components which can be incorporated into workflows on Galaxy. They remain independent of Galaxy and provide compute infrastructure and storage, enabling the OSPD to adhere to the 'data at rest' principle, minimizing the computational costs of large data transactions. Platforms are responsible for maintaining or hosting services conformant to OGC standards and any additional requirements for integration into the OSPD, notably minimal standards for metadata and registration of services and platforms in the OSPD osf registry. The OSPD is an open infrastructure and further platforms will be able to join and add services that are conformant to the OGC API standards supporting the system.

Descriptions of the developments by individual platforms participating in the current phase of the OSPD are included in an annex.

3.4. Design Considerations

Some design or implementation choices have implications across multiple components and require coordination beyond the conformance to OGC standards that otherwise enables interoperability across the OSPD.

- Platform Capabilities and Application Reproducibility

- Authentication
- openEO

3.4.1. Platform Capabilities and Application Reproducibility

As seen above, platforms serve as comprehensive environments offering interfaces and tools for processing and utilizing EO data. These platforms enable developers to not only test and execute their applications but also to deploy and share them with others for individual or integrated workflow usage.

At their core, these platforms facilitate the deployment and execution of Application Packages formatted in the Common Workflow Language (CWL) as defined by OGC OGC 20-089, each defined by unique parameters and process descriptions. The use of Application Packages ensures the portability and reproducibility of workflows across different execution platforms as it encapsulates the entire workflow environment, including all necessary software components and dependencies. These platforms, when integrated with cloud computing resources, efficiently manage and execute user-requested data processing tasks, eventually returning the processed information.

With the introduction of Reproducible FAIR Workflows, there is a need to adapt these platforms to fully address the reproducibility of the application requests and their parameters with retrospective provenance. This includes detailed documentation of each process executed and the environment it operated in. CWLProv emerges as a solution to represent workflow-based computational analysis and its provenance at various levels. CWLProv, alongside the structured provenance from the W3C PROV Model, facilitates the creation of a workflow-centric Research Object (RO) that aggregates and shares resources. This object encompasses all aspects of a workflow from its initiation to final outputs. Its structure adheres to the BagIt format, which is a set of hierarchical file layout conventions that ensure reliable storage and transfer of digital content.

The BagIt-compliant CWLProv format includes not only the data generated by the workflow but also metadata detailing the workflow's creation. This allows for the verification and replication of results, fostering an environment where applications are not only deployable but also transparent and reproducible.

3.4.2. Authentication

Many platforms restrict access to their resources to eligible users or have use constraints. Hence, authentication will be an issue not only when using one of the platforms but particularly whenever multiple independent platforms interact with each other. Users would need to authenticate at each platform separately before they can run a workflow which is a cumbersome process and in some cases difficult to manage, for example, if authentication information (e.g., cookies) expire after a few minutes. Furthermore, passing such information via tools like Galaxy can come with security issues.

The optimal approach to coordinating authentication requires further investigation and alignment between the platforms. OpenID Connect (based on OAuth 2) emerges as the option

that is supported by most platforms, but is relatively complex to implement for platforms that don't support it yet. Although Galaxy also provides [OpenID Connect](https://galaxyproject.org/authnz/config/oidc/), it is not yet clear how it can be used to authenticate at remote platforms.

3.4.3. openEO

The first major goal for OSPD is to enable users to interact with platforms implementing OGC API — Processes via Galaxy, which acts as a workflow builder and orchestrator. OSPD is meant to be modular, a design principle which is already expressed by supporting multiple platforms. However, there is only one instance of the communication protocol implemented (OGC API — Processes) and the ubuilder/orchestrator for workflows (Galaxy) in the initial design of the OSPD alternatives. To create a path for alternatives for these components, we are exploring the use of openEO in the first year of the OSPD project. This alternative path defines the **openEO API** and **openEO processes** as the communication protocol, both of which are an OGC Community Standard candidate. Similarly to the OGC API — Processes integration into Galaxy, the openEO API could also be integrated into Galaxy in future years. To provide an alternative module to Galaxy for the user interface/aggregator, the **openEO Web Editor** is being evaluated.

openEO is a project aimed at developing an open, standardized framework to connect various clients to big Earth observation cloud backends in a simple and unified way. By providing the standardized openEO API for workflows with pre-defined processes, openEO allows users to perform operations on Earth observation data across different cloud backends with minor changes to their code and algorithms and without needing to worry about the underlying complexity or specificities of each cloud provider. The openEO API aligns with the OGC APIs Standard baseline (especially Common) and STAC. openEO has implementations for various cloud backends, for example Copernicus Data Space Ecosystem (CDSE), openEO Platform, Sentinel Hub, VITO, and Google Earth Engine.

The openEO Web Editor is a browser-based graphical interface to connect to services that implement the openEO API. It allows a user to discover the offerings of the service, to create openEO workflows through a no-code environment, to manage the user-specific offerings of the service, and to visualize processing results (for D130). While it originates from the openEO community, it recently added basic support for OGC API — Processes.

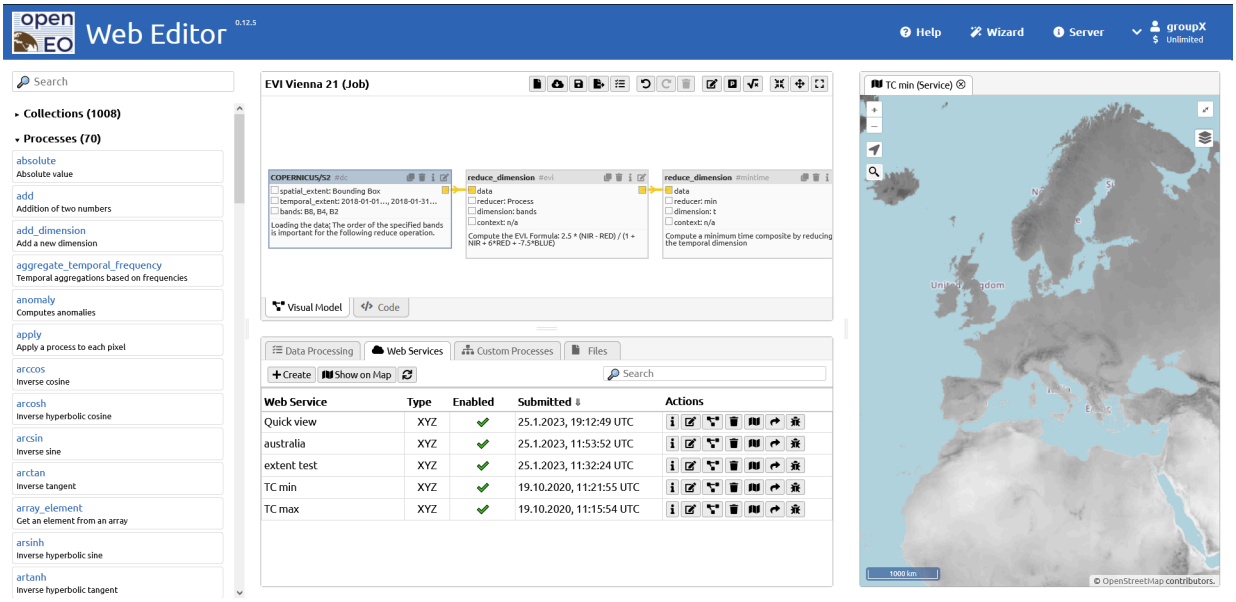

Figure 11 — The openEO Web Editor connected to the openEO Google Earth Engine implementation.

The openEO integration with OSPD was initially explored as part of three different activities:

- An instance of the openEO Web Editor will be deployed for OSPD, D110. The interface will be improved to be able to visualize results of multiple openEO instances or OGC API — Processes instances at the same time. The interface only works with cloud-native file formats that can be rendered in a browser. This provides an easy way to compare results of different cloud service providers through a single map-based interface. It also allows for the connection of various openEO instances to the OSPD, which, in turn, allows a user to run the workflows developed against the openEO implementations mentioned above. The idea to enrich the Web Editor to fully connect to multiple backends at the same time (especially OGC API — Processes) was discarded for now due to the issues encountered with the different authentication mechanisms.

- Based on the openEO Web Editor deployed for D110, the user interaction with Cloud-Optimized GeoTiff (COG) and various tiling services (OGC API — Tiles, WMTS, XYZ) has been improved in the user interface for OSPD D130. It improves the map interface to support scaling of data, free combination of bands for creating visuals like false-color composites, and perform all visualizations client-side to minimize server interaction. Additionally, it handles basic file types such as CSV, TSV, GeoJSON, PNG, and JPEG, with a modular design for future expansion to accommodate more formats like GeoParquet. Unsupported formats are made available for download.

- An openEO instance has been deployed and improved that offers an openEO API interface to parts of the Google Earth Engine (see chapter “openEO / Google Earth Engine”). It enables users to leverage GEE’s Earth observation data and computing capabilities through the openEO interface and its tooling such as the openEO Web Editor. This driver acts as a bridge, allowing for seamless access to GEE’s vast satellite imagery and geospatial

data collections for analysis and processing within the openEO ecosystem. By doing so, it allows to create comparable and reproducible EO workflows.

4 TRAINING MATERIALS

4 TRAINING MATERIALS

To enable potential users to learn to use the OSPD to support their work, either providing data and services necessary for applications that require portability, reproducibility and transparency, or building and executing workflows for these applications, training materials designed for Continuing Professional Development (CPD) and University teaching are being produced.

4.1. CPD Curriculum

4.1.1. Intended audience

The intended audience includes two user groups: the platform providers who intend to integrate their own platforms, algorithms, or datasets into the demonstrator; and the developers, researchers, or other interested people, who wish to use the OSPD's platforms, algorithms, or datasets.

4.1.2. Aims & Intended Learning Outcomes

The aim is to provide direct users of the OSPD with a self-paced, self-study approach to how the OSPD operates and supports open science. The training materials will provide an overview and key details within each module, together with signposting to further detailed information. The intended learning outcomes for each user groups are as follows.

- By completing the CPD curriculum, Platform Provider users will be able to:
    - Explain the principles of open science and why it's a positive approach to collaborative research and development.
    - Demonstrate how to create and register an open science workflow, including the key elements of accuracy, reproducibility, licensing, metadata, and scope of applicability.
    - Understand why using OGC Standards are critical to giving the OSPD a unique distinctive advantage over similar solutions.
    - Demonstrate the process for finding existing science, algorithms, and code within open science repositories.
    - Describe the processes and steps to create a workflow on their own platform, and convert this into a tool available to other users of the Galaxy system.

- By completing the CPD curriculum, OSPD Research users will be able to:

- Explain the principles of open science and why it is a positive approach to collaborative research and development.
- Outline what makes the OSPD demonstrator approach unique, and how it can benefit working practices within academic, commercial, or third sector organizations.
- Understand the process for finding existing science, algorithms, and code within open science repositories.
- Demonstrate how to create new projects on an open science repository, explaining the importance of accuracy, reproducibility, licensing, metadata, and scope of applicability.
- Apply the Galaxy Workflow Builder to create workflows from OSPD algorithms and data.

4.1.3. Outline of Curriculum

The training materials will comprise a blended combination of written text, worked examples, suggested exercises, and videos, depending on the module, offering multiple learning approaches and styles for each learner. The curriculum begins with two modules common to both user groups.

- Module 1 : Introduction to the OGC Open Science Persistent Demonstrator
- Module 2 : Introduction to Open Science

The remaining modules will focus on the specific needs of each user group.

Modules Specific To Platform Provider Users

- Module 3 : Developing Open Science Workflows
- Module 4 : Registering An Open Science Workflow
- Module 5 : Developing a Tool on a Community Platform
- Module 6: Summary & Conclusions

Modules Specific To OSPD Users

- Module 3 : Using Open Science Workflows
- Module 4 : Developing & Using an Open Science Workflow on Galaxy
- Module 5: Benefits of Using OSPD
- Module 6: Summary & Conclusion

4.2. University Curriculum

4.2.1. Intended audience

In addition to the self-paced learning content designed primarily for professionals, learning materials designed for a University level course are in development.

Target Audience: Advanced students (graduate students/senior undergraduate students) in university, participating in the D120 training program to gain advanced skills in open science research using OSPD, Galaxy workflows, VEDA, and I-GUIDE.

The high curriculum is outlined below.

4.2.2. Aims & Intended Learning Outcomes

Learning Objectives:

- Master open science principles and practices:
  - Understand the core values and benefits of open science in various research disciplines.
  - Explore ethical considerations and responsible data sharing strategies.
  - Learn about research reproducibility and its importance in open science.
- Become proficient in the OSPD platform:
  - Gain comprehensive knowledge of OSPD functionalities and capabilities.
  - Utilize various data access methods (WMS, WCS, API) and explore data effectively.
  - Integrate OSPD with other relevant open science platforms and tools.
- Master Galaxy workflows for data analysis:
  - Build and execute complex workflows for data analysis and processing.
  - Utilize advanced Galaxy tools and scripting for custom analysis tasks.
  - Integrate Galaxy workflows with other platforms like OSPD and I-GUIDE.
- Leverage VEDA data for research:
  - Explore and access VEDA datasets relevant to your research field.

    - Utilize OGC standards and tools for seamless VEDA data integration.
    - Apply VEDA data for research projects and analysis within Galaxy workflows.
- Optimize workflows with I-GUIDE:
    - Understand the benefits and capabilities of the I-GUIDE HPC platform.
    - Optimize computationally intensive workflows for parallel execution on I-GUIDE.
    - Utilize I-GUIDE resources to enhance research efficiency and scalability.

4.2.3. Outline of Curriculum

Training Course Structure:

- Module 1: Foundations of Open Science Research
- Module 2: Deep Dive into OSPD and Galaxy Workflows
- Module 3: Advanced Data Analysis with VEDA and Galaxy
- Module 4: Scaling Research with I-GUIDE
- Module 5: Open Science Research Use Cases with OSPD

5

OUTLOOK

5 OUTLOOK

5.1. Work in Progress

The OSPD communication activities aim to engage a broad target audience through a multi-faceted communication approach. This includes drafting a detailed communication plan leveraging visual communication, participation in relevant conferences, organization of targeted workshops, and the establishment of a dynamic online presence through a dedicated website, as well as social media posts through OGC channels. The objectives are to effectively disseminate information about OSPD's activities, achievements, and benefits, and to foster community engagement and collaboration. The plan emphasizes the importance of clear and consistent messaging to convey the initiative's commitment to principles of reusability, portability, and transparency in Earth Science research.

The OSPD result representation will employ storytelling as a powerful tool to represent results and communicate complex information in an accessible and engaging manner. This approach involves crafting narratives around the two use cases and the impact of OSPD's work on society. These stories will be designed to resonate with a diverse audience, from researchers and policymakers to the general public, illustrating the practical applications and societal benefits of open science and interoperable Earth Observation technologies.

5.2. Next Steps

Initial work on the OSPD focuses on defining use cases where there is a genuine need for open science workflows that leverage EO data and analyses, implementation or enhancement of OGC API Processes and Features to support access to analysis and data components on individual platforms, implementing OGC API Processes and Features in Galaxy to enable reuse of components in workflows, the development of templates to document components and workflows in osf.io, and creation of learning and advocacy materials. Initial use cases focused on the needs of public health professionals in advisory or decision making roles who would use EO data and analysis together with non-EO data sources and models to support recommendations or decision making. In this context, based on input from use case leads, the OSPD assumed strong requirements for transparency and reproducibility and a secondary requirement for portability, due to the potential for public scrutiny of recommendations and decisions and a desire to reuse established tools in novel contexts. Future phases of the OSPD may explore further use cases with similar high-level open science needs.

5.2.1. User Testing

The OSPD foresees serving two core user communities: providers of data, processing and analytical services and scientific and research users. The OSPD development team effectively assembles representatives of the first community. To test design assumptions and their implementation, the OSPD team plans to hold workshops to engage with the science user community. Current planned engagements include workshops at: FedGeoDay and the iGuide Annual Meeting.

Ongoing work in the current phase will address unmet requirements and open questions, focused on those outlined below.

5.2.2. Unmet Requirements

Work to date highlights challenges to be addressed in the current and subsequent phases of work including:

- implementation of different parts of the API Processes standard [current and subsequent phase]
- extension of support to other OGC standards based APIs, e.g. openEO, in Galaxy [subsequent phase].

5.2.3. Open Applied Research Questions

The initial design and prototyping phase of the OSPD generated applied research questions including the following.

- What level of granularity is most useful for modules (components of a workflow) integrated into Galaxy and documented as individual units in osf.io?
- To what extent is it possible to implement generalized OGC API processes in workflow builder platforms like Galaxy where a GUI for workflow composition is implemented?
- What are the minimum metadata requirements for workflows that leverage OGC standards in open science EO workflows?

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Ben Domenico: OGC 10-092r3, *NetCDF Binary Encoding Extension Standard: NetCDF Classic and 64-bit Offset Format*. Open Geospatial Consortium (2011). https://portal.ogc.org/files/?artifact_id=43734.

[2] Akinori Asahara, Ryosuke Shibasaki, Nobuhiro Ishimaru, David Burggraf: OGC 14-084r2, *OGC® Moving Features Encoding Extension: Simple Comma Separated Values (CSV)*. Open Geospatial Consortium (2015). http://www.opengis.net/doc/IS/movingfeatures/csv-extension/1.0.0.

[3] Akinori Asahara, Ryosuke Shibasaki, Nobuhiro Ishimaru, David Burggraf: OGC 14-083r2, *OGC® Moving Features Encoding Part I: XML Core*. Open Geospatial Consortium (2015). http://www.opengis.net/doc/IS/movingfeatures/xmlcore/1.0.0.

[4] OGC: OGC 11-165r2: CF-netCDF3 Data Model Extension standard, 2012

[5] Standardized Big Data Processing in Hybrid Clouds. In: Proceedings of the 4th International Conference on Geographical Information Systems Theory, Applications and Management — Volume 1: GISTAM, pp. 205–210. SciTePress (2018).

[6] Lawrence Livermore National Laboratory: NetCDF CF Metadata Conventions – http://cfconventions.org/

[7] ESIP: Attribute Convention for Data Discovery (ACDD) – http://wiki.esipfed.org/index.php/

A

ANNEX A (INFORMATIVE) OSPD COMMUNITY PLATFORMS

A ANNEX A (INFORMATIVE) OSPD COMMUNITY PLATFORMS

A.1. CRIM

A.1.1. Key features

The Data Analytics for Canadian Climate Services (DACCS) (OSF entry) project formed a multidisciplinary working group composed of Universities (Toronto, McGill, Concordia, Victoria, Alberta, Sherbrooke), and innovation hubs (CRIM, Ouranos) that lead to the development and maintenance of a federated server node network (Marble project), which aims to provide platforms for climate data analysis. Amongst this network, CRIM hosts the Hirondelle node (OSF entry), Ouranos hosts the PAVICS node (OSF entry), and University of Toronto hosts the Red Oak node. Each of these nodes, implemented using the birdhouse-deploy infrastructure as code (OSF entry), provide interoperable capabilities, but offering different combinations of processing components and datasets, according to the respective domains for which each organization focuses their work. Across these platforms, the following capabilities are provided.

- A GeoServer instance that provides storage and indexing of features, tile and coverage retrieval, map rendering, and other capabilities to view and edit geospatial data using OWS and OGC API services.
- A THREDDS catalog for scientific data and metadata cataloging.
- STAC API and STAC browser for Spatio-Temporal metadata cataloging, search and retrieval of both climate and Earth Observation datasets.
- Per-user Jupyter instances to allow interactions with the services, processing capabilities and dataset offered by the platforms.
- Multiple WPS services (finch, raven, hummingbird, etc.) that provide processing capabilities for climate indicators, hydrological modeling and analyses around climate science.
- An OGC API — Processes implementation, called Weaver (OSF entry), that supports deployment of OGC Application Packages using Common Workflow Language (CWL) and their integration in workflows with existing WPS and ESGF processing services.

- Authentication and authorization of users, groups, workspaces and access control synchronization of the contents offered by the various services, in a unified manner, using a combination of Magpie, Twitcher and Cowbird services.

With these capabilities, researchers and developers are able to interact with the processes and datasets at different levels of expertise, whether through Python scripts in Jupyter instances, or using more user-friendly interfaces as provided by the STAC browser. The common service interfaces allow distinct data and processing to be accessed by users in a standardized manner, although the underlying operations can differ greatly. For example, CRIM processes tend to focus more around dynamic applications for AI/ML development, while Ouranos's efforts focus on long-lived hydrological and climate modeling and simulations.

A.1.2. Development Roadmap

The development roadmap is two-fold:

- **Target Date**: February–March 2024

Description: Define use cases to analyze and develop an initial structure that establishes the foundations of the applications that can be reused across participants and platforms to evaluate interoperability and capabilities relevant for open-science. Planned work includes the deployment of a land-cover mapping ML algorithm (OSF entry) as well as the implementation of the Algae Bloom use case for deployment using CWL.

- **Target Date**: March–July 2024

Description: Development on the platform architecture offered by CRIM to address necessary adjustments to support open-science between participants using common interfaces, such as OGC API — Processes, CWL and CFF, and including the necessary definitions in OSF to share references to relevant platforms, API implementations and deployed OGC Application Packages using CWL that reproduce the open-science work.

A.1.3. Galaxy Integration (Galaxy as a client of my service)

Provide adjustments as needed to the already available OGC API — Processes interface (i.e.: Weaver) to execute the selected use cases, such as the Algal Bloom use case, in order to provide necessary functionalities by the Galaxy tools to interact with the server accordingly. These adjustments could include, amongst other things, the addition of capabilities to provide metadata or response formats to support open-science and workflows across the multiple D100/101 platforms in an interoperable manner. For example, OGC API — Processes allows the definition of metadata in the process description, but no explicit requirements are mandated by the API specification to provide DOIs, CFF citations and other details to ensure open science and data lineage. Further, inputs and outputs of processes allow the specification of generic media-types (e.g.: GeoTiff for the Algal Bloom use case), but there are no explicit requirements enforcing specific sensors (e.g.: the Algal Bloom algorithm requires Sentinel-2 L2A products

due to specific band calculations). Strategies to establish how these requirements can be communicated between Galaxy and OGC API — Processes remain to be defined.

A.1.4. Open design questions

- **How to share the OGC Application Package?**

Current OGC API — Processes specification allow deployment of Application Packages (through Part 2 — DRU) which can be employed for the dynamic creation and integration of open-science applications. However, the specific implementation of supported Application Packages by an OGC API — Processes compliant server are left up to the design choices of each specific server. Furthermore, little to no details are provided regarding how the deployed application can then be shared externally. Only an abstract process description is made available through the core endpoints of the API. In order to make OGC Application Packages interoperable across OGC-compliant platforms and Galaxy, more design discussions must be made to provide details that will allow replicable open science workflows.

- **How to integrate with Galaxy considering OGC API — Processes?**

Current tool definitions in Galaxy expect users to provide the code and undertake the testing necessary to integrate their application using the technical structure ('the wrapper') prescribed by Galaxy. This produces notable constraints that need to be addressed.

First, the tool definition assumes that users that want to share their application as a result of open-science have the knowledge to do so efficiently, due to the high amount of technologies involved, and that they understand the intricate implementation details of Galaxy. In reality, the great complexity surrounding the DevOps requirements in the context of a platform could discourage the integration of open-science work because of this technological and knowledge barrier.

Second, the rapidly evolving nature of AI/ML and open-science resulting from it, asks for highly dynamic and evolving applications, for which the integration overhead in Galaxy could limit adoption.

Third, multiple services and processes are already available through standard interfaces, such as OGC API — Processes, which cannot be directly leveraged in Galaxy. Because of the quantity of existing applications, it is not reasonable to convert all of them to the format required by Galaxy.

Therefore, for all above reasons, more work needs to be performed for Galaxy to support OGC API — Processes, such that the dynamic and evolving ecosystem surrounding applications for Climate, Earth Observation and Disaster response analysis can be quickly and easily integrated by users from multiple domain backgrounds.

- **How to expose services, processes and data through OSF?**

One proposed candidate for sharing open-science work is the OSF platform. While the structure seems promising for building a knowledge archive to share scientific work, referencing documents, code, and relevant material sources to reproduce science, there is still a lack of

details regarding a standard approach to provide metadata that could be reused by different platforms. Potential metadata annotation formats, such as STAC, CFF citations and others have to be considered and evaluated, to determine if they are sufficient to achieve reproducible and licensable open-science.

A.1.5. SWOT analysis

- Strength

    A) All the platform definitions and offered services are completely open-source.

    B) Services have been applied in the context of multiple large-scale projects, OGC testbeds, pilots and related work.

    C) Maintainers of the code base are active participants in the OGC API — Processes specification design and STAC community.

- Weaknesses

    A) The platform is not a central data hub (eg: such as Copernicus, Earth-Search, etc.). With a few exceptions, it usually relies on established external sources to obtain reference imagery products and climate datasets. Catalogs are employed to store derived products resulting from analyses.

- Opportunities

    A) Improve metadata and data lineage capabilities for reproducible science and workflows.

- Threats

    A) Authentication and authorization requirements to access specific capabilities offered by the platform must be supported by Galaxy. How to describe these requirements remains to be established.

    B) Under-evaluation of the complexity involved by workflow definitions and capacities to reproduce the experiments across platforms. Overview of service — iguide

A.2. GeoLabs

A.2.1. Key features

The GeoLabs integration leverages the ZOO-Project, an Open-Source Processing Engine and an OSGeo Project that complies with OGC standards, initially based on WPS 1.0.0 and later on WPS 2.0.0. It serves as a Reference Implementation of the OGC API — Processes — Part 1: Core. The ZOO-Project builds upon the ZOO-Kernel, which can execute local processes implemented in a variety of programming languages, including C/C++, Java, Python, PHP, Perl, and JavaScript (limozjs or Node.js).

ZOO-Project supports the execution of complex geospatial processing through existing libraries like OrfeoToolBox (OTB) and SAGA-GIS. It offers a framework to publish results as standard OGC Web Services or APIs (WMS, WFS, WCS), with minimal modifications required to process metadata. The integration with High-Performance Computing (HPC) environments, as demonstrated in projects like GeoSUD and Phidias-HPC, enables remote task execution using protocols such as Secure Shell (SSH) and workload managers like SLURM. This capability has been further enhanced through the use of Singularity containers for packaging workflows.

During the Testbed19 project, ZOO-Project showcased its ability to deploy workflows or individual applications using the OGC API Processes — Part 2: Deploy, Replace, and Undeploy (DRU) draft specification. The integration with the Earth Observation European Platform for Cloud and Analytics (EOEPCA) supports the Common Workflow Language (CWL) conformance class, providing a comprehensive environment for EO data management. This integration also includes a unified platform with the eoAPI (part of VEDA), supporting full OGC API — Processes (Part 1, Part 2, and partial Part 3) for seamless interoperability.

The ZOO-Project has been successfully integrated with various HPC components, making it suitable for scenarios requiring heavy computational resources and large-scale data processing. This framework enables users to interact with the Triton Inference Server for geospatial inference, providing an API that conforms to the OGC API — Processes — Part 1: Core Standard. In addition, the ZOO-Project-DRU Helm chart simplifies the deployment of the project, supporting both DRU and CWL, which enables easy deployment and management on Kubernetes clusters.

A.2.2. Development Roadmap

- Enhancing CWL File Support: Extend support for handling multiple CWL files, allowing for complex workflow execution and adherence to the full OGC Application Package specification.

*Strengthening Galaxy Integration: Improve the integration with the Galaxy platform by utilizing the ZOO-Project services to access VEDA's processing and data capabilities.

- Unified Authentication and Access Management: Establish a unified authentication service for the platform, leveraging OpenID Connect, to streamline user access across the different integrated systems.
- Securing and Managing Metadata: Implement secure access to results published through STAC collections and manage metadata more effectively to ensure compliance with best practices.

A.2.3. Current Integration Achievements:

- Successful Integration of Basic ZOO-Project Version with Galaxy: The basic ZOO-Project version was integrated with Galaxy, enabling access to its processing capabilities through Galaxy's user interface.
- Unified OGC API Entry Point: ZOO-Project-DRU combined with eoAPI provides a single entry point for the full OGC API — Processes — Part 1: Core and Part 2: DRU, supporting seamless interaction and workflow management.
- Deployment and Execution of Algae Use Case: CWL-based workflows for the algae use case, provided by CRIM, are deployable and executable with minor modifications, showcasing the platform's versatility.
- Kubernetes Deployment for Scalability: ZOO-Project-DRU and eoAPI components have been successfully deployed on Kubernetes clusters using Helm charts, enhancing the platform's scalability and operational efficiency.
- Publication of STAC Collections: eoAPI endpoints are utilized to publish STAC collections resulting from process executions, simplifying data discovery and access for users.

A.2.4. Open Design Questions:

- Authentication and Privilege Sharing: How will user authentication and privileges be maintained and shared among different platforms? Can a standard OpenID Connect authentication mechanism handle the necessary interactions?
- Data and Metadata Management: How will datasets and results be referenced and managed in the OSF catalog? What is the process for ensuring the reproducibility of shared workflows?
- Workflow Versioning and Repository Management: What is the best practice for maintaining, versioning, and publishing the workflow files and Docker images needed for CWL-based workflows?

A.2.5. SWOT Analysis:

- Strengths:
    A) Supports a wide range of processes (700+) through the ZOO-Kernel.
    B) Robust integration with HPC resources and geospatial libraries like OTB and SAGA-GIS.
    C) Standardized OGC API support enhances interoperability with external platforms.

- Weaknesses:
    A) Limited data access services beyond what MapServer provides.
    B) High resource requirements for processing complex workflows.

- Opportunities:
    A) Potential to build stronger bridges between ZOO-Project and VEDA.
    B) Ability to leverage VEDA's capabilities to enhance the ZOO-Project's processing power.

- Threats:
    A) Authentication and authorization requirements may complicate integration.
    B) Resource constraints could limit the scalability of demonstrations on the ZOO-Project platform.

A.3. Terradue

A.3.1. Key features

Terradue brings an on-demand processing service for flood mapping using Sentinel-1 satellite data. This service simplifies flood extent mapping by utilizing pre- and post-event co-polarized Sentinel-1 Sigma Nought imagery. The process involves a user-defined dB threshold to binarize backscatter, effectively distinguishing water from other surfaces. The service masks out areas classified as water in both pre- and post-event images to accurately depict flooded regions.

The processing service requires specific parameters like pre- and post-event Sentinel-1 acquisitions, the AOI, backscatter thresholds, and the HAND (Height Above the Nearest Drainage) asset threshold. It's designed to be a straightforward solution, with potential future enhancements like automatic image binarization, the use of permanent water masks, and extension to support optical data. The service follows a workflow with key steps for:

- **Calibration of Sentinel-1 Data**: Pre- and post-event Sentinel-1 datasets are calibrated upon user upload. An optional step involves generating the HAND auxiliary dataset for the image area.
- **Image Processing and Binarization**: The service performs co-located stacking of images within the specified Area of Interest (AOI). It then binarizes the two co-polarized single-band images using a user-defined backscatter threshold in dB, resulting in pre- and post-event water masks.
- **Masking and Flood Map Generation**: The service generates a flood map by identifying pixels classified as water only in the post-event imagery. It involves mathematical operations on the water masks to differentiate pre-existing water bodies from flood water.
- **Vectorization and Filtering**: The final step involves converting the filtered flood mask into a polygon vector format, applying a sieve filter to remove smaller clusters of pixels. The service aims to provide a rapid and efficient means of flood mapping, crucial for disaster response and environmental monitoring. Its capacity to utilize satellite data effectively can be invaluable in timely and accurate flood assessment, crucial for planning and mitigation in flood-prone areas.

Terradue's flood mapping service adheres to the OGC API Processes standards and uses the Common Workflow Language (CWL) for packaging and deployment. The service has been integrated with the Galaxy platform, allowing users to access its capabilities directly from Galaxy through the OGC API Processes endpoints. This integration offers flexibility and customization, enabling users to modify parameters for tailored flood mapping solutions.

The service's architecture supports potential enhancements, such as the inclusion of optical data and automated image binarization, to broaden its applicability and improve processing efficiency. The combination of advanced technology, user-defined settings, and adherence to standards positions Terradue's flood mapping service as a robust tool for EO data analysis and flood management.

A.3.2. Development Roadmap

- Development of Core Algorithms: Implement the core flood mapping algorithms, focusing on Sentinel-1 data processing, binarization of backscatter, and integration of HAND datasets.
- Application Packaging: Package the application ensuring all dependencies and resources are included for seamless deployment and execution.
- Application Deployment and Integration: Deploy the packaged application on the server. Integrate it with the server's existing OGC API Processes infrastructure.

A.3.3. Galaxy Integration

Galaxy interacts with the flood mapping service via the OGC API Processes endpoint, allowing users to configure and execute the flood mapping workflow from the Galaxy interface. This setup leverages Galaxy's existing tools for processing and visualizing EO data while using the OGC API Processes standard to manage workflow execution. The integration has successfully demonstrated the execution of flood mapping workflows, providing a seamless interface for users to access and utilize the service.

A.3.4. Open design questions

An open issue for the deployment of the flood mapping service is the management of access control from Galaxy with user credentials delegation. As a solution, implementing signed URLs could be a viable approach. Signed URLs provide a secure way to grant temporary access to resources. By using unique tokens embedded in these URLs, the service can authenticate and authorize users without the need for more complex authentication mechanisms. This method simplifies the access control process, making it more efficient and user-friendly, while still maintaining a high level of security and control over who can access the service and when.

A.3.5. SWOT analysis

Strengths:

- **Advanced Technology**: Leverages sophisticated algorithms and Sentinel-1 satellite data for accurate flood mapping.
- **User-Centric Design**: Intuitive interface and user-defined settings enhance accessibility.
- **Compliance with Standards**: Adheres to OGC API Processes Part 2 and uses Common Workflow Language (CWL) for packaging, ensuring portability and efficient deployment across various platforms.
- **Flexibility and Customization**: Allows for diverse user inputs, enhancing service applicability.

Weaknesses

- **Dependency on Satellite Data**: Reliant on Sentinel-1 data availability and quality.
- **Limited Scope**: Focused primarily on flood mapping, potentially restricting broader environmental applications.
- **Resource Intensive**: The service may require significant CPU resources for data processing, which could lead to scalability and efficiency issues in deployment.

Opportunities:

- **Expansion to Optical Data**: Future integration with optical data can broaden the service's applicability and accuracy.
- **Collaboration Potential**: Opportunities to collaborate with governmental, academic, and non-profit organizations in disaster management and environmental studies.

Threats

- **Operational Challenges**: Managing large datasets and maintaining consistent service quality.
- **Sentinel-1 Data Availability**: Risk of limited or interrupted access to Sentinel-1 data, which is crucial for the service's functionality.
- **Resource Constraints**: Limited CPU resources and potential deployment challenges.

A.4. Ellipsis Drive

Ellipsis Drive provides a storage solution in which users can host arbitrary spatial datasets (ranging from 1mb to 10tb). Regardless of the original input files/data Ellipsis Drive always ensures scalability, access management and compliance to all relevant standards. You could compare Ellipsis Drive to a managed geoserver with a file system like architecture built on top of it.

A.4.1. Key features

Ellipsis Drive provides a storage for spatial data with the following key features:
1 input flexibility: supports any common spatial format for raster, vector and point cloud. (from netcdf, to geodatabase. From las files to ECW's). See the full list of file types:
2 output flexibility. When adding a number of files to a layer you can use the layer via any OGC protocol. Think of WMTS, WFS, WCS etc.
3 mosaicing. You can place up to 10 million files within the same layer
4 scalability. No matter the size of the original content the layer always performs the same way.
5 Access Management. You can easily set permission to content and share it. Many authentication methods are supported out of the box.
6 Embedded in an intuitive file system.
7 Equipped with a powerful spatial search based on relevant metadata

A.4.2. Development Roadmap

1 Exposing coast line information and building footprints to the Galaxy platform..

2 Exposing a STAC endpoint for Galaxy to search content with

A.4.3. Galaxy Integration (Galaxy as a client of my service)

Exposing layers as WFS/WCS services
Exposing a STAC core and STAC search endpoint

A.4.4. Open design questions

There are four ways vector data can be fed to the Galaxy platform. WFS, as a batch in a compressed list, via a Features API or as vector tiles. Feeding data to Galaxy via a Features API would be preferable. It is still unclear which of these protocols would be most suitable for Ellipsis Drive.

Executing processes in Ellipsis Drive has been newly added and it is still an open design question how to effectively distribute recources and charge costs without complicating the UX too much.

A.4.5. SWOT analysis

Strengths

- Ellipsis Drive works and performs on uncurated user input data relieving pressure on the user.
- Ellipsis Drive can serve a hosted dataset in multiple OGC protocols without the need to duplicate data.

Weaknesses

- The STAC search is implemented for collections not assets due to its internal definitions.
- Ellipsis Drive only supports query, with no analytical support planned.

Opportunities

- To further improve dataset metadata and findability.

Threats

- The chosen strategy is dependent on Galaxy's ability to query external datasets using OGC protocols.

A.4.6. Activities performed

Under D101 — Open Science Platform, Ellipsis Drive has contributed to co-design and requirements scoping, and has used its pre-existing platform (Ellipsis Drive) as well as the newly developed Information Factory, its assets and its capabilities to expand on existing endpoints, protocols, packages and plugins to facilitate cross-platform workflows that have been collaboratively created during the OSPD initiative.

A.5. PolarTEP

A.5.1. Key features

The Polar Thematic Exploitation Platform (Polar TEP) provides a complete working environment where users can access algorithms and data remotely, obtain computing resources and tools that they might not otherwise have, and avoid the need to download and manage large volumes of data.

The following are key components of Polar TEP:

- **Data Discovery, Visualization, and Interaction**: Earth Observation and related data is easily accessible for visualization or simple analysis. (Link: https://viewer.polarview.org)
- **Interactive Development Environment:** Users can develop their own applications in Python using Jupyter Notebooks. (Link: https://polartep.hub.eox.at/)
- **Machine Learning Platform**: Polar TEP provides MLflow to manage all stages of the ML lifecycle, including experimentation, reproducibility, deployment, and model registry.
- **Execution Environment**: Headless execution or Conda environments are used to provide processing with minimal execution overhead and computing resources scaled to the current demand.

Polar TEP Support of Open Science

Polar TEP provides tools to assist researchers in participating in all facets of open science (see Figure 1), including:

- **Input Data**: with data discovery and visualization.
- **Analysis**: with an Interactive Development Environment and Machine Learning.
- **Collaboration**: with code sharing, commenting, and discussion forums.

- **Results**: through the packaging of data, algorithms, methods, and conclusions so that research can be reproduced and extended.
- **Dissemination**: by telling the story of the research to influence decisions and foster further research.

Figure A.1 — Polar TEP support of Open Science

A.5.2. Development process

Within the OGC OSPD pilot, Polar TEP supports the algal bloom use case. The figure shows the OSF open science process cycle that begins with a research idea, moves through data collection and analysis, and continues to the publishing of results, before repeating again.

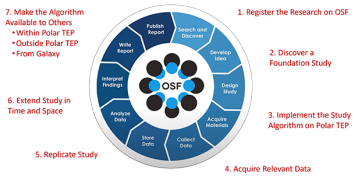

Figure A.2 — OSPD Open Sceince Process

Around the outside of the figure are the example steps taken within the OSPD project. These consist of:

1. Registering the Research on OSF
2. Discovering a Foundation Study
3. Implementing the Study Algorithm on Polar TEP
4. Acquiring Relevant Data
5. Replicating the Study

6. Extending the Study in Time and Space

7. Making the Algorithm Available to Others

1. Registering the Research on OSF

By registering on the Open Science Framework, research can be managed and shared throughout the entire project lifecycle.

2. Discovering a Foundation Study

A literature review resulted in a foundation study for the OSPD project. It concerns the use of Sentinel 2 satellite data to determine water quality. The study algorithms calculate measures of Chlorophyll a, Cyanobacteria and Turbidity. The foundation study used data from the Alqueva reservoir in Portugal in October 2017.

3. Implementing the Study Algorithm on Polar TEP

The study algorithms had already been coded in Java Script. The code was then translated into Python and implemented in Polar TEP as a Jupyter Notebook.

4. Acquiring Relevant Data

Polar TEP was then used to find the Sentinel 2 images that were used in the foundation study. Here is the Sentinel 2 image over the Alqueva reservoir on the foundation study date of October 12th, 2017.

Figure A.3 — Sentinel 2 Image of the Study Area

5. Replicating the Study

The algorithms were applied to the Sentinel 2 data. The results from the foundation study paper are shown on the left. The results from the Polar TEP replication of the study are shown on the right. The differences are due to variations in colour reproduction.

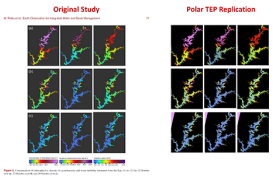

Figure A.4 — Study Replication

6. Extending the Study in Time and Space

Polar TEP was then used to extend the use of the algorithms in time and space. Here the three algorithms have been used to analyze an area of Lake Erie in Canada in June 2023.

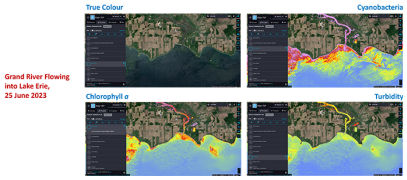

Figure A.5 — Study Extension

7. Making the Algorithm Available to Others

To support the work of others in replicating and extending the research, the algorithms are made available within and outside of Polar TEP. Within Polar TEP, they are available in four forms:

- As On-the-fly Java Script
- As a Data on Demand service
- In a Jupyter Notebook, and
- Through Headless Execution

Outside of Polar TEP, they are available in two ways:

- As a Python Notebook, stored on OSF, and
- Accessed from the Galaxy platform (see the next section)

Use on Polar TEP or from Galaxy requires a Polar TEP account. The code on OSF is freely available.

A.5.3. Galaxy integration

Within the OSPD project, Polar TEP will utilize the Galaxy Platform as a web-based system that allows users to develop and manage complex data processing workflows. For example, the algae bloom algorithm will be triggered from the Galaxy platform but executed on Polar TEP.

The Galaxy tools will act as intermediaries, communicating through the OGC API Processes endpoint to submit and monitor jobs, and retrieve results and metadata. This integration will allow users to execute sophisticated workflows without needing to manage computing infrastructure across multiple platforms.

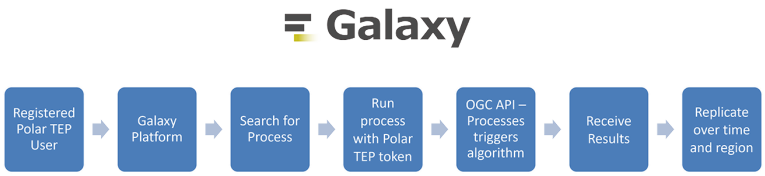

Figure A.6 — Polar TEP / Galaxy Integration

A.5.4. Open design questions

How will cloud credits for data processing on Polar TEP be provided? Options: i) A Galaxy account is created on Polar TEP so that users coming from Galaxy can invoke processes; ii) A user has an account on Polar TEP and user authentication is passed from Galaxy. In either case, there needs to be communication between Galaxy and Polar TEP to provide the estimated cost of processing for approval by the user before processing is triggered.

A.5.5. Outlook

Within the first phase of OSPD, Galaxy did not have time to integrate Polar TEP. It is expected that integration will be accomplished in a manner identical to that used for the Terrabyte platform (see Section A.6).

A.6. Terrabyte — DLR

Terrabyte is a high-performance data analytics (HPDA) platform currently in operation as a collaborative effort between the Earth Observation Center (EOC) of the German Aerospace Center (DLR) in Oberpfaffenhofen and the Leibniz Supercomputing Centre (LRZ) in Garching near Munich. Terrabyte is openly accessible to DLR and LRZ scientists, as well as national and international collaborative partners of DLR engaged in joint projects.

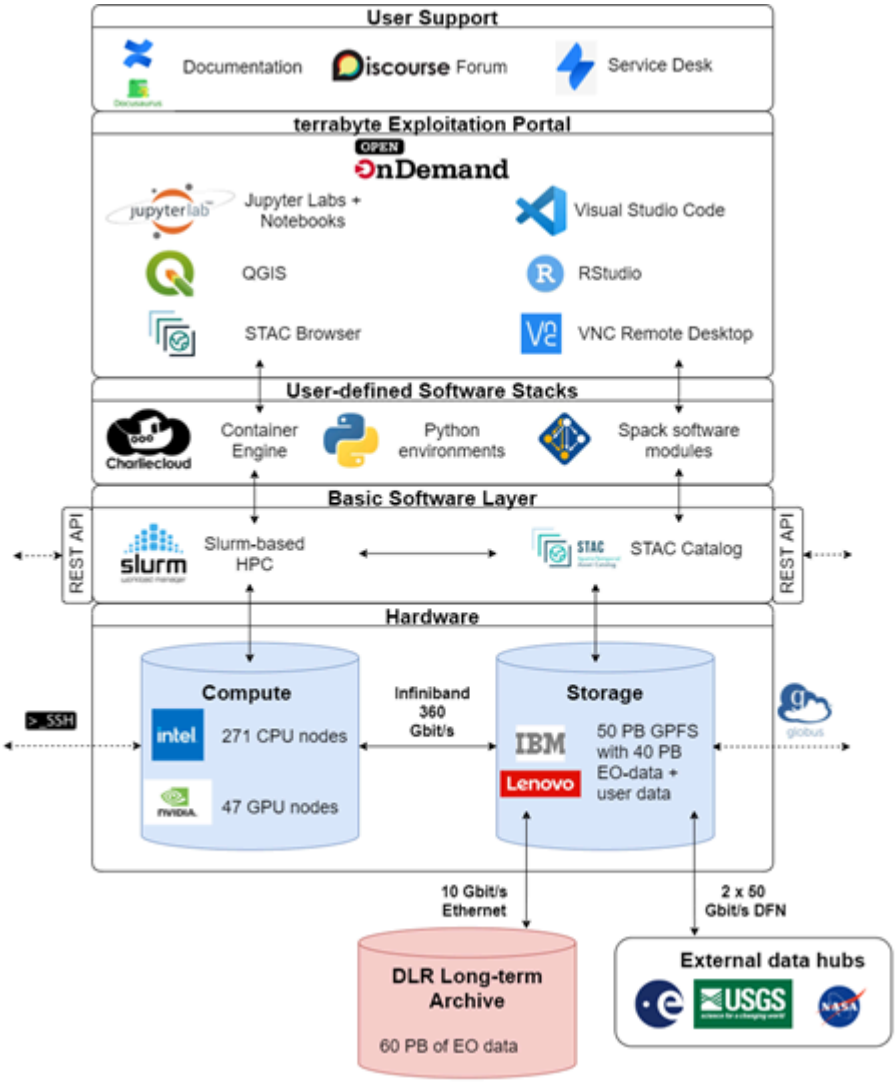

Figure A.7 — Key features and services of terrabyte

At the core of terrabyte is a robust Slurm-based High-Performance Computing (HPC) cluster, comprising 271 CPU nodes, each equipped with two 40-core Intel Xeon Platinum processors, and 47 GPU nodes, each featuring four NVIDIA A100 accelerators. Additionally, terrabyte boasts a high-speed Infiniband-connected GPFS (General Parallel File System) storage system with a substantial capacity of 50 PB, including 40 PB dedicated to Earth Observation (EO) data. EO data is constantly being updated through downloads from external data hubs or reloads from DLR's long-term archive. This infrastructure, combined with its user-oriented software stack consisting of an Open OnDemand-based exploitation portal, various options for user-defined software stacks, REST APIs, and accessible user support resources, empowers researchers to effortlessly deploy processing workflows ranging from small to very large-scale in a powerful computing environment, eliminating the need to download massive amounts of Earth Observation (EO) data.

Terrabyte exposes the Slurm- and STAC REST APIs to OGC OSPD. Additionally, an OGC API Processes endpoint has been deployed as part of the first phase of OSPD. Through these APIs,

OGC OSPD is able to make use of terrabyte's Slurm-based HPC capabilities, user-defined software stacks, and the large amount of EO data.

Table xx gives an overview of links to existing terrabyte tools. Please note that some of these tools, including the OGC API Processes endpoint, are only accessible for registered terrabyte users or from the DLR network.

Table A.1

TOOL	URL
Documentation	https://docs.terrabyte.lrz.de
Support Forum	https://forum.terrabyte.lrz.de
Service Desk	https://servicedesk.terrabyte.lrz.de
Self-Registration Service	https://register.terrabyte.lrz.de/
Compute Portal	https://portal.terrabyte.lrz.de
HPC Login node	ssh://login.terrabyte.lrz.de
Slurm REST API Endpoint (not publicly available)	https://slurmrest.terrabyte.lrz.de/
STAC Browser	https://stac.terrabyte.lrz.de/browser
STAC REST API Endpoint	https://stac.terrabyte.lrz.de/public/api
OGC API Processes endpoint	**https://processing.terrabyte.lrz.de/**

Table xx: Links to existing terrabyte tools

A.6.1. Developments

After setting up the OGC API Processes endpoint using the open source Python package **`pygeoapi`**, the algae bloom use case was implemented as a containerized EO application package based on OGC EO Application Package including Common Workflow Language (CWL). The workflow can be run on terrabyte either directly through the OGC API Processes endpoint or from Galaxy (see next section).

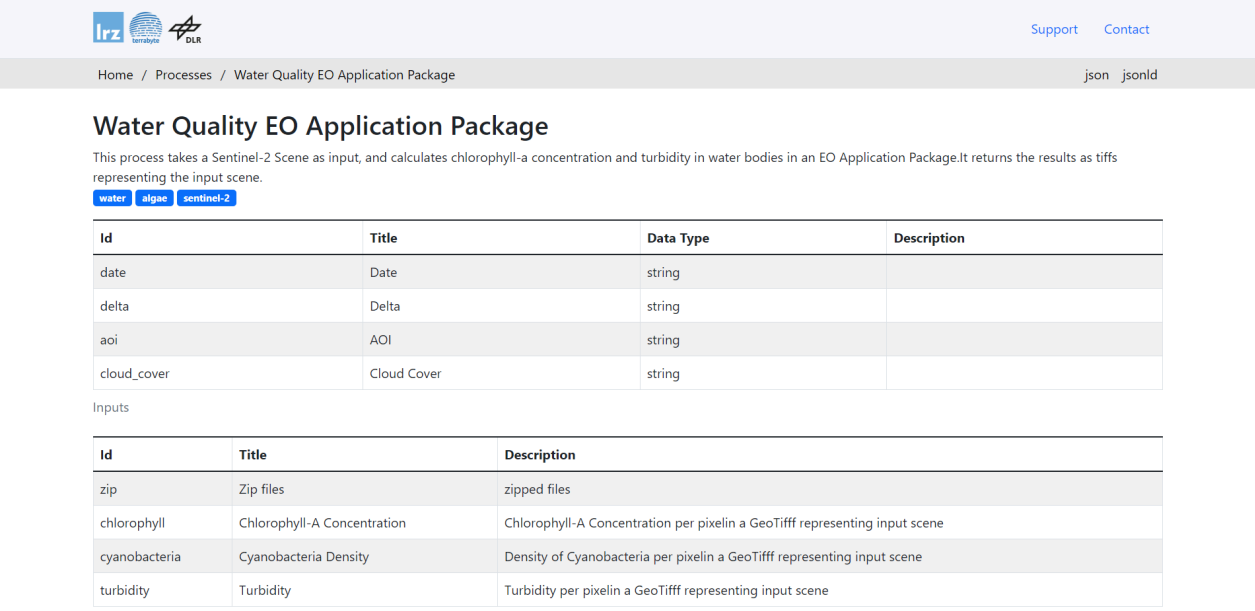

Figure A.8 — Screenshot of the OGC API Processes web page of terrabyte

A.6.2. Galaxy Integration (Galaxy as a client of my service)

Galaxy serves as a web-based workflow management platform. The algae bloom workflow was deployed as a tool within Galaxy based on terrabyte's OGC API Processes endpoint. Thus, processing does not occur on Galaxy itself, but on terrabyte. In the established setup, the Galaxy tool acts merely as a client that communicates with the OGC API Processes endpoint to submit and monitor processing jobs, as well as receive processing results. Since terrabyte is only accessible for registered users, access to the OGC API Processes endpoint is controlled by OpenID Connect (OIDC) bearer tokens that have to be obtained through a two-factor authentication (2FA)-token service and submitted as part of the Galaxy workflow.

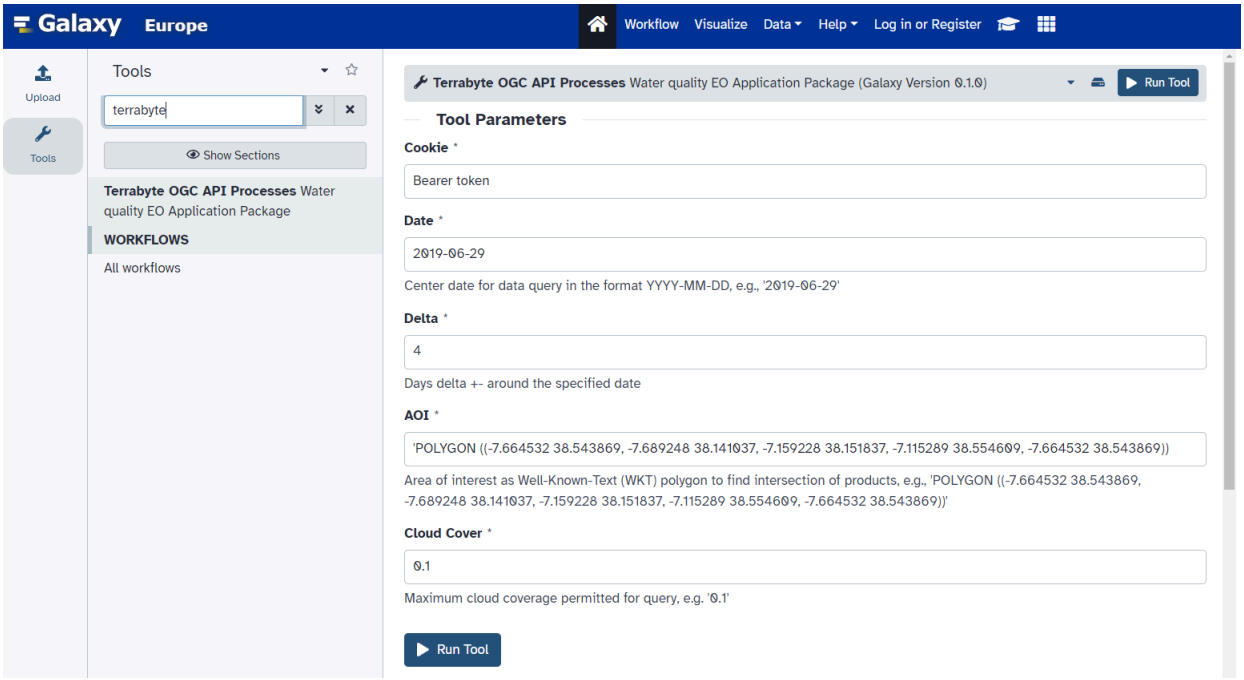

Figure A.9 — Screenshot of the terrabyte Galaxy tool

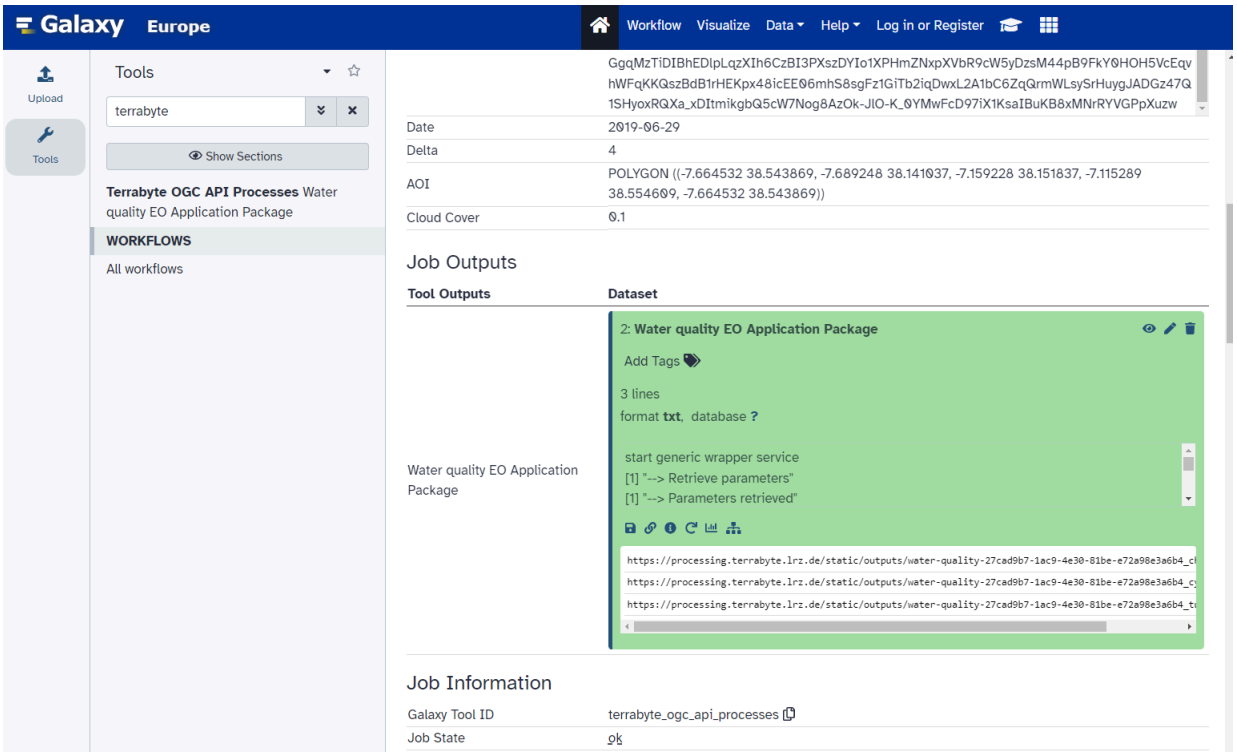

Figure A.10 — Screenshot of the terrabyte process execution in Galaxy

A.6.3. Restrictions

As is common with many platforms, access to terrabyte is restricted to specific user groups. Accessing terrabyte's resources and APIs requires authentication procedures that include 2FA and the generation of OIDC access tokens. These requirements limit the interoperability of workflows in Galaxy across different platforms and hinder the sharing of intermediate data among them.

A.6.4. Summary & Outlook

As part of the OGC OSPD Phase 1, terrabyte has been extended with an OGC API Processes endpoint to execute OGC EO Application Packages on the terrabyte infrastructure. Similar to the other OSPD platforms, the endpoint has been integrated into Galaxy, which acts as a central execution platform.

While the current setup focuses solely on OGC EO Application Packages, another standardized approach for providing reproducible algorithms, OpenEO, should also be considered in future phases of OSPD. terrabyte will continue to work on deploying and operating services for the standardized execution of processes via both OGC API Processes and OpenEO API. The prototype OGC API Processes service deployed within OSPD will serve as a foundation to transition this service into full operations.

A.7. Development Seed

A.7.1. Key features

VEDA Deployment using Kubernetes: A unique instance of the Visualization, Exploration, and Data Analysis (VEDA) platform was deployed using kubernetes. This demonstration is the first test of a full VEDA deployment using kubernetes, and provide a blueprint for users hoping to implement their own instance of VEDA outside of NASA supported instances. Lessons learned were used to improve the source code for VEDA, and documented to encourage others to deploy VEDA in their own cloud environments.

Dashboard: The primary frontend component of the VEDA platform is the Dashboard. We deployed a Dashboard instance that has content relevant to the use cases. This Dashboard visualizes EO data, showcasing the abilities of Analysis-Ready, Cloud-Optimized (ARCO) data, and is accessible from osf.io.

Data Services: We created a unique instance of the data ingestion services and STAC catalog developed on VEDA. We use Airflow to ingest data through an OGC Processes API, and there are OGC-compliant APIs to make the STAC records available to Galaxy. Users are be able to

search the STAC catalog via API, through the Galaxy hub. Documentation for these services are available through osf.io.

A.7.1.1. Links to any extant tools

A) Existing VEDA Dashboard: https://www.earthdata.nasa.gov/dashboard/

B) VEDA Documentation: https://nasa-impact.github.io/veda-docs/

A.7.2. Development Roadmap

1. Further hardening of authentication across platforms
2. Improved integration with ZOO to allow ZOO to directly ingest data into VEDA

A.7.3. Galaxy Integration (Galaxy as a client of my service)

To support OSPD users building workflows using Galaxy, we provide an OGC-compliant API for Galaxy to conduct a simple STAC search on any of the data that has been ingested to match the use cases. This enables users to find comparable ARCO datasets that support their use cases. Through Galaxy, users can provide broad, spatiotemporal search queries, or more precise queries using collection-specific properties such as cloud coverage, or the availability of specific spectral bands. The results will be provided as a GEOJSON "FeatureCollection".

This workflow is useful and viable because the GeoJSON output fits into Galaxy's existing understanding of "datasets" and processes. The GeoJSON output can be used as a "dataset", despite describing dozens or hundreds of matching assets from several collections. Galaxy users can then narrow down and select which assets to include in the rest of their workflow, but this model allows them to continue to do so using this VEDA instance, without leaving the Galaxy environment.

An example of this type of service can be found on VEDA's STAC. This example STAC is not currently OGC-compliant, but the implementation here will be iterated on to be compliant with OGC processes.

A.7.4. Open design questions

1. What is the best way to ensure discoverability of the entire VEDA platform, since large portions of VEDA's value proposition are not a part of the overall Galaxy workflow?
2. Are there improved methods of accessing STAC records through Galaxy or the OGC Processes API?

3. Once you have completed a STAC search, how can results persist for science reproducability?

A.7.5. SWOT analysis

Strengths

1. Most of the code is already tested and deployed in multiple instances
2. Using kubernetes will provide a cloud-agnostic option for those wanting to implement VEDA
3. The planned integration with Galaxy is a function that has been used in many other cases, so it should be stable

Weaknesses

1. There are no substantive analytical capabilities that are a part of this VEDA instance, so it might feel out of touch with the rest of the OSPD.
2. The Dashboard is not an integral part of the workflow.
3. VEDA is not a SaaS tool and is not a fully custom implementation. Thus, users might struggle to understand what they are looking at when they see it in the OSPD

Opportunities

1. To improve the metadata of data ingested in VEDA, as well as the VEDA tools
2. To test the interoperability of the VEDA tools with other tools
3. To test a full VEDA deployment using kubernetes

Threats

1. If we don't have ready access to a COG formatted data to ingest into this VEDA instance, we may run into issues
2. If the data ingested into this VEDA instance is not relevant to the use case developed, users might not understand why they conducted this STAC search as part of the overall workflow
3. Authorization issues dealing with so many systems might cause more development work than anticipated
4. Overview of service — Open Science Studio

A.8. Open Science Studio

A.8.1. Key features

Use the same code in your notebook: APIBaker allows you to seamlessly transition your notebook code into a fully functional API with just a few clicks. You can select specific code cells or functions from your notebook to expose as API endpoints. This feature ensures that the current workflow developed within a jupyter notebook can be reused without the need for code rewriting or duplication.

Custom domain support: With APIBaker, you can easily configure your API to use your own custom domain name. This adds a professional touch to your API endpoints and provides a branded experience for your users.

Version control integration: APIBaker seamlessly integrates with version control systems (e.g. Git) to manage the evolution of your API. Each time you make changes to your notebook and re-generate the API, APIBaker automatically tracks these changes and prompts you to commit them to your version control repository. This ensures a clear history of all modifications to an individual's API codebase and enables collaboration with team members.

Token-based security: APIBaker provides robust security features by allowing you to secure your APIs with tokens. When setting up your API, you can generate unique access tokens that users must include in their requests to authenticate and access the endpoints. This token-based authentication mechanism ensures that only authorized users can interact with your API, protecting sensitive data and preventing unauthorized access.
By incorporating these extended features, APIBaker empowers users to create, customize, and secure APIs from their Jupyter notebooks with ease, flexibility, and confidence.

Strengths:

- **Unique Offering:** APIBaker provides a unique solution for converting Jupyter notebook code into API endpoints with minimal effort, catering to the needs of data scientists, analysts, and developers.
- **Ease of Use:** The extension simplifies the process of creating APIs by using existing code from notebooks, allowing users to generate APIs with just a few clicks within the familiar JupyterLab environment.
- **Customization Options**: APIBaker offers customization features such as domain configuration and token-based security, empowering users to tailor their APIs to specific requirements and preferences.
- **Integration with Version Control:** Integration with version control systems like Git enhances collaboration and facilitates versioning, ensuring that changes to the API codebase are tracked and managed efficiently.

- **Security Features**: Token-based authentication enhances API security, enabling users to control access to endpoints and protect sensitive data from unauthorized access.
- **Efficiency Gains** (in progress): The upcoming shared folder for Argo Workflows and caching features are expected to significantly reduce execution time, enabling faster performance.
- **Enhanced User Experience** (in progress): Planned features like log downloading, a loader for handling large logs, and integration of Conda environments aim to improve usability and reduce setup time for users.
- **Reliability Improvements** (in progress): By separating log queries and execution responses into distinct services, API performance and reliability are anticipated to improve.
- **Customization Options** (in progress): Custom domain setup and token-based security will remain, continuing to provide users control over their API endpoints.

Weaknesses:

- **Limited Functionality**: While APIBaker streamlines the process of creating basic APIs from notebook code, it may lack some advanced features and customization options compared to dedicated API development frameworks.
- **Dependency on JupyterLab**: APIBaker's functionality is tightly coupled with JupyterLab, which may limit its accessibility to users who do not use Jupyter notebooks for their development workflow.
- **Learning Curve:** Users who are not familiar with JupyterLab or web API concepts may require some learning to effectively utilize APIBaker and understand its features and capabilities.
- **Complex Cache Management** (in progress): New caching capabilities might require additional resources and maintenance to ensure accuracy, which could increase operational complexity.
- **Environment Compatibility** (in progress): Automatically copying Conda environments could lead to compatibility issues if certain libraries conflict across projects or users.
- **Dependency on JupyterLab** (in progress): The tool's functionality will still be tied to JupyterLab, limiting accessibility for users who work outside this ecosystem.

Opportunities:

- **Expansion of Features:** There is an opportunity to expand APIBaker's feature set by adding support for additional programming languages, advanced authentication mechanisms, and integration with cloud services for deployment.
- **Integration with Other Tools:** APIBaker could explore partnerships or integrations with other data science and development tools to enhance its functionality and reach a broader user base.

- **Community Engagement:** Building a strong community around APIBaker through documentation, tutorials, and user forums can foster collaboration, encourage contributions, and drive further development and adoption.
- **Enhanced Argo Workflows Integration** (in progress): Further optimization of Argo Workflows could enable even faster processing times, especially beneficial for handling large files.
- **Expanded Community Engagement** (in progress): Offering resources such as detailed logging tutorials and cache management tips could foster user engagement and broaden APIBaker's reach.
- **Cross-Platform Compatibility** (in progress): Extending support to environments outside JupyterLab could attract a wider user base, making APIBaker accessible to non-Jupyter users.

Threats:

- **Competition:** The API development landscape is competitive, with many established frameworks and platforms available for creating APIs. APIBaker may face competition from these alternatives, which offer more comprehensive feature sets and established user groups.
- **Dependency on Specific Technology (Argo)** (in progress): Continued reliance on Argo Workflows may limit flexibility, especially if more adaptable or cost-effective alternatives become available.
- **Competition from Full-Stack API Platforms** (in progress): Competing frameworks with broader capabilities continue to pose a challenge, potentially impacting APIBaker's competitiveness.
- Overall, APIBaker has the potential to address the growing demand for easy-to-use API development tools within the data science community, but it must continue to innovate and evolve to stay competitive and meet the evolving needs of its users.

A.8.2. Recent Development Activities

Following client feedback, the team is actively working on **APIBaker version 2.1**. This version will introduce several enhancements focused on improving workflow efficiency, environment compatibility, and user experience. Planned updates include:

- **Feature: Argo Workflows Shared Folder Access** Argo Workflows will gain access to a shared folder, which will remove the need to copy large files (e.g., data files, images, models) across storage. This update is expected to reduce delays in image creation and subsequent execution, optimizing workflow efficiency and minimizing resource usage.
- **Feature: User's Conda Environment Copying** APIBaker will incorporate a feature to automatically copy the user's CONDA environment into the execution image. This

improvement aims to remove the need for users to manually specify libraries, reducing errors caused by missing dependencies and simplifying the script setup process.

- **Feature: Execution Log Download** Users will be able to download execution logs, making it easier to identify and troubleshoot errors in their runs. This feature will improve user access to detailed log data for in-depth analysis.
- **Fix: Log Loader Addition** To address delays when loading large logs, a loader will be introduced to inform users of the loading status, enhancing the user experience by providing feedback on system activity.
- **Fix: Log Query and Execution Response Separation** The log query and image execution response processes will be separated into two distinct services, aiming to enhance performance and provide more reliable access to logs and execution feedback.
- **Feature: Run Caching** A caching mechanism will be implemented for execution runs, which will improve response times for subsequent queries. This feature will optimize the user experience in scenarios involving recurring executions.

A.8.3. Prospects for Future Activities

- **Optimization of Execution Speed**: Further enhancement of Argo Workflow integration and run caching.
- **Cross-Environment Flexibility**: Expanding compatibility to additional environments beyond JupyterLab and supporting multiple deployment methods.
- **Community Development and Documentation**: Building an active community through resources, tutorials, and forums focused on cache usage, error troubleshooting, and Argo Workflows efficiency tips.
- **Advanced Security and Authentication**: Exploring advanced authentication options beyond token-based security, such as OAuth2, to enhance data protection for more sensitive applications.

A.8.4. Open Design Questions

- **Cache Management Strategy**: Determining the optimal caching strategy for execution runs, including cache expiry times and storage limits, to balance performance with resource management.
- **Conda Environment Isolation**: Exploring how to isolate copied Conda environments effectively to prevent compatibility issues across different execution contexts.
- **Argo Workflows Optimization**: Identifying efficient methods to streamline access to shared folders, especially for extremely large datasets, to avoid bottlenecks in Argo

Workflows. Error Reporting in Logs: Deciding on the level of detail and format for downloadable logs, aiming for a balance between usability and comprehensiveness.

A.9. OSF

A.9.1. Key features

User Flow Diagram Development: A significant milestone was the creation of a potential user flow diagram, illustrating how collaborators could utilize OSF's project and registration services to store and archive research materials. This workflow, critical for enhancing user experience and integration capabilities, was continuously refined to align with grant needs and technical feasibility. We look forward to continuously refining this workflow in year 2.

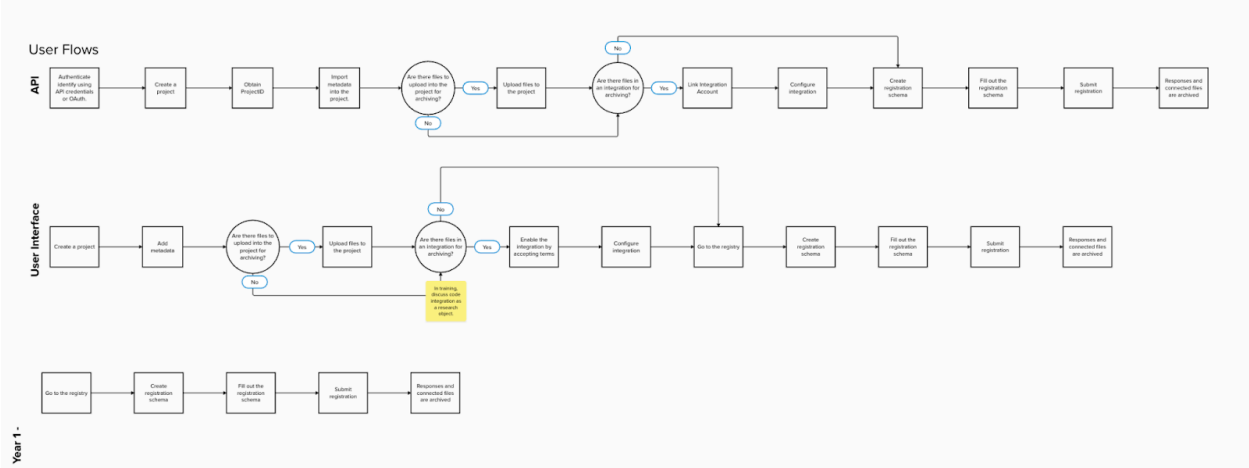

Figure A.11 — User Flow Diagram

OSF Registry for OSPD: We successfully established a publicly available OSF Registry for OSPD, designed to facilitate the archiving of research through tailored registration template for platforms in use with geospatial research. This template captures essential information to enable reproducibility and further research, adhering to branding requirements. Additional templates will be created for other objects (ex. services) in year 2. You can find the registry at osf.io/registries/ogccommunity.

A.9.1.1. Links to any extant tools

- **Open Science Framework (OSF)**: The central platform for the project, available at osf.io.

- **OSPD Test Registry**: The publicly accessible registry specific to OSPD, available at osf.io/registries/ogccommunity.

A.9.2. Development Roadmap

CompletedMilestones:

- Completion of the OSPD-specific registry and development of the platform registration template.
- Hosted an OSF 101 webinar that familiarized collaborators with the platform and its key features.

Challenges and Resources:

- There were no major challenges requiring additional resources. Continuous evaluation remains essential to address any emerging needs.

A.9.3. Galaxy Integration (Galaxy as a client of my service)

- **Galaxy Integration:** While Galaxy and Polar TEP were not integrated in Year 1, prioritization was given to developing workflows for the identified use cases. However, the OSF team is continuing to build out an updated add-on integration marketplace, which will be released in 2025. This will enable further integrations, including those with Galaxy and Polar TEP, and beyond..
- **External Collaborations:** With the new add-on integration marketplace, we plan to continue our collaboration with Polar TEP towards integration, aiming to highlight and promote open science practices within and beyond the project.

A.9.4. Open design questions

Addressing Core Design Principles: TThe OSPD pilot leveraged the OSF's registry service to embody principles of Reusability, Portability, Transparency, and ESG + SDG consciousness. The registration templates were crafted to ensure detailed documentation of research protocols, facilitating reproducibility and adherence to open science practices.

A.9.5. SWOT analysis

Strengths

- **Rigorous and Transparent Research Process:** The OSF registry's specific questioning framework promotes rigor, transparency, and reproducibility in research studies, distinguishing it from other platforms.
- **Archiving and Digital Backup:** Duplication of all submitted information to the Internet Archive ensures long-term preservation, enhancing trust in data reliability.
- **Financial Safeguarding:** A continuously growing $47,000 contingency plan underscores a commitment to data preservation even in the event of organizational closure.
- **Potential for Integration:** The exploration of OSF projects as a bridge for connecting platforms to the registration process, although not fully realized, indicated a forward-thinking approach to platform interoperability.

Weaknesses * None at this time.

Opportunities

- **Enhanced Collaboration and User Base Expansion:** The planned add-ons marketplace offers significant opportunities to foster interdisciplinary collaboration and attract broader user groups by supporting new integrations.
- **Technological Partnerships:** The new integrations with other platforms suggest a path forward for improving data sharing and visibility.
- **Integration and Metadata Synchronization:** The lack of synchronization and integration capabilities for metadata across platforms was a limitation in Year 1. However, this is now an opportunity with the development of the add-ons marketplace to support greater integration.
- **Interdisciplinary Collaboration:** Although this pilot was geospatially-focused, the potential for interdisciplinary collaboration remains an area of future exploration. Further development will include leveraging visualization platforms and ensuring the provenance and credibility of shared data.

Threats

- **External Factors:** While no immediate external threats were identified, ongoing vigilance for technological advancements, competitive platforms, and changes in open science policies remain essential to maintain relevance.
- **Internal Risks:** While no significant internal risks were identified, continuous assessment of technical, operational, and financial health remains critical for sustainable development and adoption.

A.10. openEO / Google Earth Engine

A.10.1. Key features

Google Earth Engine (GEE) is a cloud-based platform for planetary-scale environmental data analysis that combines a multi-petabyte catalog of earth observation and geospatial datasets. GEE has powerful computing capabilities to enable scientists, researchers, and developers to detect changes, map trends, and quantify differences on the Earth's surface. Launched by Google in 2010, this platform provides access to a vast archive of historical and real-time geospatial data, including satellite imagery, climate datasets, and elevation data, among others. By leveraging Google's cloud infrastructure, Earth Engine facilitates the analysis of large datasets, making it a crucial tool for environmental monitoring, natural resource management, agricultural planning, and climate change research. Its user-friendly interface and extensive API support various applications, from deforestation tracking and water resource management to urban planning and disease spread modeling, significantly contributing to global efforts in sustainability and conservation.

openEO is an open API designed to unify the way geospatial processing and analysis are performed across various cloud-based platforms. It aims to make Earth observation data more accessible and easier to analyze for scientists, researchers, and developers by providing a standardized interface for interacting with large Earth observation data repositories. Through openEO, users can connect to multiple data sources, perform complex analyses, and scale their computations without needing to be experts in the underlying cloud computing technologies. This initiative supports a wide range of pre-defined processes for a wide range of use cases.

The **openEO Google Earth Engine driver** is a component that implements the openEO API for Google Earth Engine, enabling users to leverage GEE's Earth observation data and computing capabilities through the openEO standardized API. This driver acts as a bridge, allowing for seamless access to GEE's vast satellite imagery and geospatial data collections for analysis and processing within the openEO ecosystem. By doing so, it allows to create comparable and reproducible EO workflows.

The key features of the openEO Google Earth Engine (GEE) driver include:

- Standardized Access
- Scalable Data Processing
- Cross-Platform Compatibility
- On-demand processing via web services
- Accessibility for Non-Experts
- Community and Ecosystem Integration

A.10.1.1. Links to any extant tools

- https://earthengine.openeo.org
- https://github.com/Open-EO/openeo-earthengine-driver
- https://editor.openeo.org/?server=https%3A%2F%2Fearthengine.openeo.org&discover=1
- https://earthengine.google.com/

A.10.2. Tasks completed

- openEO API updates, especially updating to the latest version of the API
- Authentication with Google Accounts
- Improved, more efficient and more robust architecture, especially for chaining processes
- Implemented new processes and reimplemented existing processes, especially in the categories: apply, reducers, filter, aggregations, math operations, and resampling
- Improved STAC metadata output for batch jobs
- A draft version to run processing as “tasks” on GEE with storage of the results in Google Drive
- Implemented the Algal Bloom use case

A.10.3. Galaxy Integration (Galaxy as a client of my service)

Not planned for OSPD year 1. Instead an instance of the openEO Web Editor will be deployed (see the corresponding section).

A.10.4. Design investigations

- openEO works based on the concept of data cubes. Google Earth Engine doesn’t work on data cubes. As such a mapping between the data cube view and GEEs data types needs to be established, either on the client-side (openEO GEE driver) or on the server-side (GEE). This may lead to bottlenecks for some workflows where intermediate data needs to be retrieved from Google’s processing engine before the actual result has been generated. We could solve many issues by writing code that adapts to more cases, but this was not possible for all issues. I’ve reported all issues to the GEE team and they are currently under investigation.
- Other issues reported in the previous ER have been resolved.

- Handling of no-data values is different in openEO and GEE. This may lead to inconsistencies in the results between openEO instances. We report corresponding values to the users so that they can handle them in their following steps.
- The options to integrate cloud storage are Google Drive and Google Cloud Storage (mostly for batch processing and file uploads). Google Drive provides free storage which we can connect to from the openEO GEE driver, but the Google Cloud Storage needs a credit card assigned to the Google Account, which we don't have and can't assume users to have. As such we should fall back to Google Drive for reliable behavior. Google Drive is limited in storage quota and might be difficult to access and manage though. This could be improved to optionally also support Google Cloud Storage in the future, but it seemed out of scope and not of interest for OSPD yet.
- Integration into Galaxy has been postponed to year 2.
- Use Case 1 (Algal Bloom) can run on GEE without issues. Use Case 2 needs further evaluation with the GEE team with regards to the feasibility and which parts could run on GEE.

A.10.5. SWOT analysis

Strengths

- **Massive Data Archive**: Google Earth Engine provides access to 1000+ collections of historical and real-time satellite imagery and geospatial datasets.
- **Powerful Computing Capabilities**: It leverages Google's cloud computing infrastructure, enabling complex geospatial analyses and processing large datasets very fast.
- **Pre-defined set of processes**: The openEO specification pre-defines a set of processes that allows writing of portable workflows. For OGC API — Processes no pre-defined processes are available and as such a wide variety of proprietary processes evolve.

Weaknesses

- **Closed Source**: While the openEO GEE driver is open-source, the Google Earth Engine itself is closed-source and it can't be reviewed for potential issues.
- **Limited features**: The openEO GEE driver will not provide the full GEE features through the openEO interface.
- **No custom code (UDFs)**: GEE doesn't allow to run custom code (in openEO called *UDFs*) or all kinds of ML models, as such it may not be able to run all use cases.

Opportunities

- **Stable and more feature-rich openEO interface for GEE**: The existing openEO GEE interface was a proof-of-concept and was limited in scope. The new implementation is

more stable and feature-rich so that it becomes a viable alternative to access GEE via its proprietary interfaces. It is also easier to extend and maintain.

- **Portable workflows**: Workflows can be written in a portable manner so that they can be executed on GEE and other openEO-based implementations such as CDSE or openEO Platform.
- **Batch jobs**: The Google batch job interface can be made available via the standardized openEO API.
- **Authentication via Google or custom user accounts**: Authentication is possible for Google accounts via OpenID Connect. If no access to GEE is available for certain users, custom user accounts can be made available via HTTP Basic.
- **Integration into openEO ecosystem**: Google Earth Engine is usable via the user-friendly openEO tooling (e.g. Python, R, Julia, JS, Web Editor),

Threats

- **Dependency on Google**: Google may change the pricing model or the availability of the service at any time (which has happened in the past).
- **Technical difficulties with data cubes**: There are various technical challenges to solve that occur from making GEE processing available via a data cube interface. Performance implications apply.
- **Cloud Storage limitations**: Access to Google Cloud Storage is paid only and a credit card is needed. The alternative Google Drive has a restricted storage quota and may mix data from different sources in a single interface (e.g. if you also store private data in the Google Drive).

A.11. I-GUIDE

A.11.1. Key features

Institute for Geospatial Understanding through Integrative Discovery Environment (I-GUIDE) is a 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 [1] enables seamless integration of advanced cyberinfrastructure and cyberGIS capabilities to empower users to undertake computationally reproducible and data-intensive geospatial analytics at scale.
The most relevant feature of the I-GUIDE Platform to OSPD is the CyberGIS-Compute framework [2] which provides transparent access to HPC resources, while enabling users to manage containerized computational models and tools that can be configured and launched

on HPC through an interactive interface. The following figure illustrates the architectural components of the CyberGIS-Compute framework.

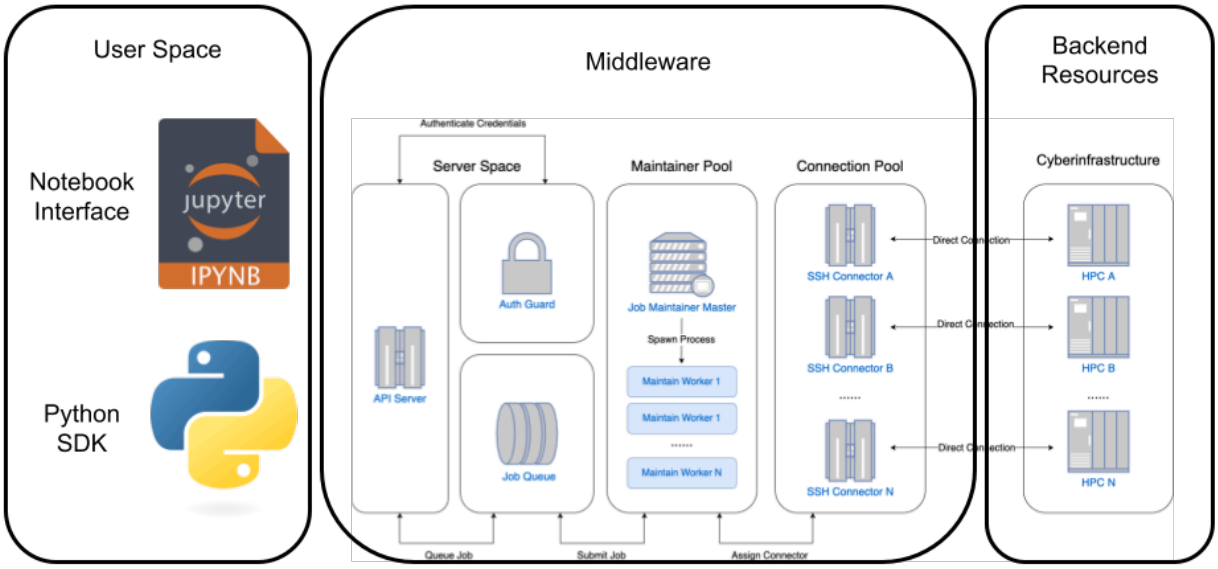

Figure A.12 — The CyberGIS-Compute middleware connects python and jupyter notebook interfaces with diverse cyberinfrastructure back end architectures.

CyberGIS-Compute middleware is not tied to a specific HPC backend. It can provide access to any HPC backend given certain configuration requirements. At this time, CyberGIS-Compute allows HPC jobs to be scheduled on following HPC backends

- Keeling hosted at University of Illinois Urbana Champaign (https://cybergis.illinois.edu/infrastructure/hpc-user-guide/hpc-a-quick-start-guide/)
- Anvil hosted at the Purdue University (https://www.rcac.purdue.edu/anvil)
- ACES hosted at the Texas A&M University (https://hprc.tamu.edu/aces/)

    a) Development Roadmap

    b) Galaxy Integration (Galaxy as a client of my service) As discussed in the section, there are four options suggested by the Galaxy [4] team for integrating tools and processes with the Galaxy framework. Option-2 seems to be the best to move forward with integrating the CyberGIS-Compute framework with the Galaxy platform due to the following reasons

- CyberGIS-Compute middleware is already implemented and available as a service on a publicly accessible server
- CyberGIS-Compute provides a Python SDK to interact with functions deployed on the server. These functions are accessible via the SDK and execute on the server Considering CyberGIS-Compute as a black box, we can integrate its functionality with the Galaxy framework via the SDK. We will create a module having self-contained functions, which

can accept data input, accept code snippets to invoke processing on the data on remote HPC resources and get output results from the HPC back to the Galaxy platform.

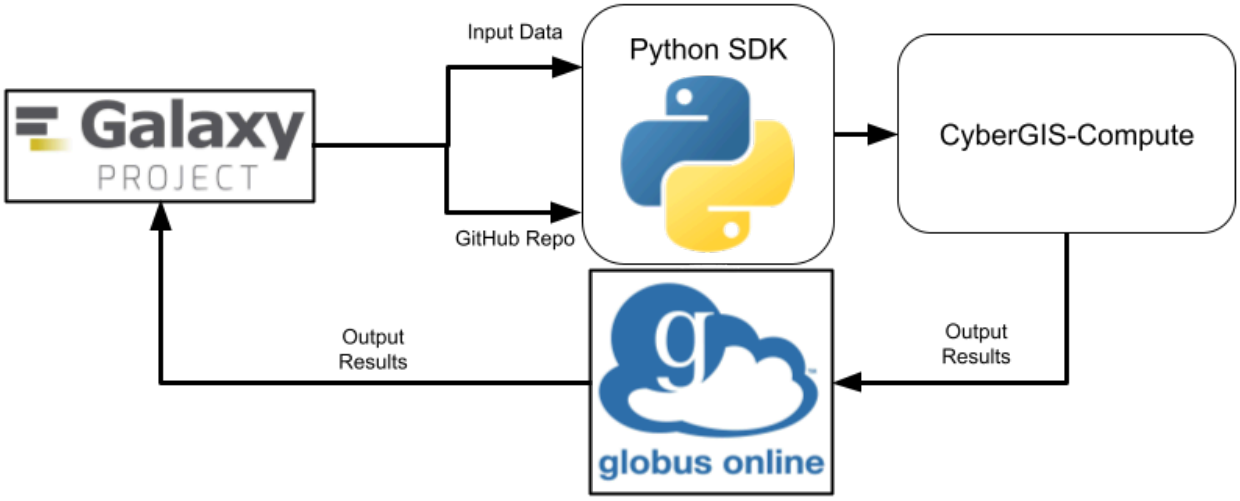

Figure A.13 — *Globus* provides a facility for the transfer of data between the Galaxy platform and HPC environments.

CyberGIS-Compute has an option which allows for interacting with the service from anywhere. In this case, all interactions with the CyberGIS-Compute is managed by an automated agent deployed over CyberGISX. While this can be a good starting point, this approach has serious drawbacks in terms of data management.

A.11.1.1. Data Management

As figure # shows, CyberGIS-Compute relies mostly on Globus online service to move data in and out of the backend HPC environment. At this time, we are not confident about Galaxy's capability to work with Globus services. However, prior work [3] can be used as a starting point.

A.11.1.2. Input Data

Input data will be passed directly to the CyberGIS-Compute SDK. This will be useful when data is already in the Galaxy framework or is generated as an output of another Galaxy tool. This option will usually be fine for small size datasets. CyberGIS-Compute is fully capable of ingesting local data and copying it over to the backend HPC for processing. Alternatively, for larger input data, Globus endpoint can be used to move data to the HPC backend.

A.11.1.3. Output Data

Once a submitted job finishes execution on the HPC backend, CyberGIS-Compute allows accessing generated output data via Globus endpoint. This allows for handling any dataset size generated from the job processing on HPC.

A.11.1.4. Interim Data

Most of the data management in CyberGIS-Compute is handled via Globus online services. Interim data can also be managed by Globus. However, instead of moving data back to the Galaxy server after job completion, it will be more efficient to return only _Globus links _to the output interim data. The links can be used to directly copy data over to the next processing service/tool avoiding any extra data movement.

A.11.1.5. Data Format

CyberGIS-Compute jobs are mostly format agnostic and can work with any data format relevant to the job being submitted to the HPC. The output data can either be original data, in any format, or a link to the data on Globus endpoint.

A.11.2. References

[1] I-GUIDE Platform
[2] CyberGIS-Compute
[3] Galaxy-Globus Integration