# D4.1 User feedback report on the functional design of the federation

| | | | |
|---|---|---|---|
| **Status:** | Under EC Review | **Planned due date:** | 30/06/2022 |
| **Version:** | 1.0 | **Submission date:** | 30/06/2022 |
| **Lead Participant:** | Deltares | **Lead Author:** | Björn Backeberg |
| **Related WP:** | WP4 | **Document Ref:** | D4.1 |
| **Dissemination Level:** | Public (PU) | | |
| **Document Link:** | https://doi.org/10.5281/zenodo.6784402 | | |


**Deliverable Abstract**

At the C-SCALE project's inception, six use cases were identified. Their objective was to deploy applications on the C-SCALE federated infrastructure and provide feedback on their experiences working with the providers in co-designing the C-SCALE federation. This report provides information about the use case dependencies and who implemented these. Additionally, the use cases provide feedback about speed of access to resources and data, the ease of use of the resources, the appropriateness of the technology used to implement the use case, the resultant usability of the application running on the federated infrastructure, the missing functionality/resources, the effectiveness of support from the providers and the overall satisfaction of the service/resource. Additionally, all use cases make improvement suggestions which in summary are centred around simplifying and

harmonising resources provisioning and access, improving documentation and providing examples and training and finding mechanisms to ensure that data and tooling are readily available for users so that users can focus on science. In general, the feedback from the use cases is positive. Users recognise the value of developing customised solutions in collaboration with the federation providers and appreciate the flexibility of the resources being provided.

## COPYRIGHT NOTICE

This work by parties of the C-SCALE consortium is licensed under a Creative Commons Attribution 4.0 International License. (http://creativecommons.org/licenses/by/4.0/).

C-SCALE receives funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 101017529.

## DELIVERY SLIP

| Date | Name | Partner/Activity |
|---|---|---|
| **Lead Author:** | Björn Backeberg (BB) | Deltares/WP4 lead |
| **Contributors:** | Gennadii Donchyts (GD) | Deltares/use case |
| | Lorinc Meszaros (LM) | Deltares/use case |
| | Jaap Langemeijer (JL) | Deltares/use case |
| | Frederiek Sperna Weiland (FSW) | Deltares/use case |
| | Joost Buitink (JB) | Deltares/use case |
| | Milutin Milenković (MM) | WUR/use case lead |
| | Matthias Schramm (MS) | TU Wien/use case |
| | Bernhard Raml (BR) | TU Wien/use case participant |
| **Moderated by:** | Björn Backeberg (BB) | Deltares/WP4 lead |
| **Reviewed by:** | Guido Lemoine (GL) | JRC |
| | Zdeněk Šustr (ZS) | CESNET/WP2 lead |
| **Approved by:** | C-SCALE Activity Management Board (AMB):<br>Christian Briese (CB), Diego Scardaci (DS), Charis Chatzikyriakou (CC), Zdeněk Šustr (ZS), Enol Fernández (EF), Björn Backeberg (BB), Sebastian Luna-Valero (SLV) | EODC, EGI Foundation, Deltares, CESNET |

## DOCUMENT LOG

| Issue | Date | Comment | Author(s) |
|---|---|---|---|
| **V0.1** | 10/06/2022 | First draft | BB, GD, LM, JL, FSW, JB, MM, MS, BR |
| **V0.2** | 23/06/2022 | Incorporated feedback from reviewers | BB |
| **V1.0** | 30/06/2022 | Final version approved by AMB | BB |

# Table of Contents

List of Images ........ 6
List of Tables ........ 7
List of Acronyms ........ 8
Executive Summary ........ 9
1 Introduction ........ 10
1.1 Approach ........ 10
1.2 Report structure ........ 13
2 Use cases ........ 14
2.1 Land surface water change monitoring ........ 14
2.1.1 Functionality ........ 14
2.1.2 Dependencies ........ 14
2.1.3 Feedback ........ 15
2.2 Coastal hydrodynamic and water quality modelling ........ 16
2.2.1 Functionality ........ 16
2.2.2 Dependencies ........ 16
2.2.3 Feedback ........ 18
2.3 Seasonal river discharge ensemble forecasting ........ 19
2.3.1 Functionality ........ 19
2.3.2 Dependencies ........ 19
2.3.3 Feedback ........ 20
2.4 Monitoring tropical forest recovery capacity ........ 20
2.4.1 Functionality ........ 20
2.4.2 Dependencies ........ 21
2.4.3 Feedback ........ 21
2.5 Real-time reservoir surface water area monitoring ........ 22
2.5.1 Functionality ........ 22
2.5.2 Dependencies ........ 22
2.5.3 Feedback ........ 23
2.6 Wetland Water Stress Analysis ........ 24
2.6.1 Functionality ........ 24
2.6.2 Dependencies ........ 25

2.6.3 Feedback ........ 26
3 Summary of improvement suggestions from users ........ 28
4 Conclusion ........ 30

# List of Images

Figure 1. Proposed workflow towards automated provisioning of compute and data resources for users ........ 29

# List of Tables

Table 1: Summary of use cases, their scope and providers supporting their deployment ............. 12
Table 2. Summary of improvement suggestions .......................................................................... 28

# List of Acronyms

| Acronym | Description |
|---|---|
| AAI | Authentication and Authorisation Infrastructure |
| AppDB | EGI Applications Database |
| API | Application Programming Interface |
| ARD | Application Ready Data |
| CDS | Climate Data Store |
| CI/CD | Continuous Integration / Continuous Delivery |
| CMEMS | Copernicus Marine Service |
| C-SCALE | Copernicus – eoSC AnaLytics Engine |
| ECMWF | European Centre for Medium-Range Weather Forecasts |
| EODC | Earth Observation Data Centre |
| EOSC | European Open Science Cloud |
| EO | Earth Observation |
| ESA | European Space Agency |
| GEE | Google Earth Engine |
| GFS | Global Forecast System |
| GPU | Graphics Processing Unit |
| HPC | High Performance Computing |
| HTC | High Throughput Computing |
| WP | Work Package |
| WUR | Wageningen University & Research |
| REST | REpresentational State Transfer |
| SAR | Synthetic Aperture Radar |
| SRAM | SURF Research Access Management |
| STAC | SpatioTemporal Asset Catalog |
| SURF | collaborative organisation for IT in Dutch education and research |
| TU Wien | Vienna University of Technology |
| TRL | Technology Readiness Level |
| VM | Virtual Machine |
| VO | Virtual Organisation |

# Executive Summary

The C-SCALE project plans to deliver a federated compute and data infrastructure unifying the resource tier by connecting Copernicus resources and EOSC compute and storage providers. In so doing, the infrastructure will offer homogenous access to compute and data resources including Copernicus and Earth Observation (EO) data, with a seamless user experience where the complexity of resource provisioning and orchestration is abstracted away from the end-user.

The above will be achieved by working closely with research communities who, through use cases, will co-design, test, pilot, refine and ultimately help create a federated infrastructure that delivers data and platform services that are useful for the community.

Here the six use cases identified at the project's inception report on their experiences working with the providers evaluating

- speed of access to resources and data
- ease of use of the resources
- appropriateness of the technology used to implement the use case
- resultant usability of the application running on the federated infrastructure
- missing functionality/resources
- effectiveness of support from the providers
- the overall satisfaction of the service/resource.

The main improvements proposed by the use cases centre around:

- Simplifying and harmonising resources provisioning and access
- Improving documentation and providing examples and training
- Finding mechanisms to ensure that data and tooling are readily available for users so that they can focus on science

However, in general, the feedback from the use cases is positive. Users recognise the value of developing customised solutions in collaboration with the federation providers and appreciate the flexibility of the resources being provided.

# 1 Introduction

The C-SCALE project plans to deliver a federated compute and data infrastructure unifying the resource tier by connecting Copernicus resources and EOSC compute and storage providers. In so doing, the infrastructure will offer homogenous access to compute and data resources including Copernicus and Earth Observation (EO) data, with a seamless user experience where the complexity of resource provisioning and orchestration is abstracted away from the end-user.

The above will be achieved by working closely with research communities who, through use cases, will co-design, test, pilot, refine and ultimately help create a federated infrastructure that delivers data and platform services that are useful for the community.

In this context, the primary objective of Work Package 4 (WP4) is to support the user-driven co-design by iteratively deploying use cases on the infrastructure to test the usability and functional design of the federation components and services.

## 1.1 Approach

The integration activities to deliver the C-SCALE Federation are driven by user requirements derived from the use cases to guarantee the delivery of an easy to use environment that satisfies needs from research communities wishing to exploit EO and Copernicus data.

To achieve the above, six use cases have been deployed on the different providers participating in the project (Table 1). Use cases with TRLs ≥ 8 were selected to test performance and scaling in a federated infrastructure. Additionally, these use cases assist the federation providers in setting up services to ensure that future use cases with potentially lower TRLs, focussing on research and science, are adequately supported. The current set of use cases resource dependencies range from complex multi-CPUs to simple JupyterLab instances. Note that through the project's open call (https://c-scale.eu/call-for-use-cases/) four additional use cases[1] have been identified that will contribute to future iterations of the federation co-design, but these are not included in this feedback report.

Feedback is provided via a User Forum (https://github.com/c-scale-community/discussions) which facilitates collaboration between the researchers deploying the use cases and the infrastructure providers supporting the use cases.

For each use case a Github project is set up (https://github.com/orgs/c-scale-community/projects). The projects and repositories are private and require users to be members of the C-SCALE Github organisation. Eventually the project will make the repositories public and register associated services in the EOSC Portal.

[1] Four additional use cases from the open call:

1. On-demand semantic EO data cubes
2. Data-driven forecasting of global shorelines
3. Development and application of advanced processing chains and data standard for exploitation of Sentinel-1 and other SAR data
4. Benchmarking GPU accelerated SAR ARD generation on cloud infrastructure

Each Github project contains a Kanban-style board, which is used to manage, coordinate and communicate on activities towards deploying each use case. In this way the project can leverage Github's issue tracking capabilities to track activities and record how effectively these were resolved, which in turn will facilitate providing feedback to the providers.

*Table 1: Summary of use cases, their scope and providers supporting their deployment*

| Use case | Lead organisation | Scope | Providers |
|---|---|---|---|
| **Land surface water change monitoring** | Deltares | This use case compares the performance of accessing and processing Sentinel-2 data on remote object storage and local object storage. | INCD, INFN |
| **Coastal hydrodynamic and water quality modelling** | Deltares | This use case tests the interoperability, scalability and performance of combining cloud and HPC compute and data resources in its workflow | GRNET, CloudFerro/CREODIAS |
| **Seasonal river discharge ensemble forecasting** | Deltares | This use case compares the performance and scalability of ensemble forecasting on HTC and cloud compute and data resources. | SURF, EODC |
| **Monitoring tropical forest recovery capacity** | WUR | This use case explores methodologies to create interoperable and scalable analysis ready data cubes for Landsat and Sentinel-1, including assessing the quality of the Sentinel-1 GRD application ready data to detect magnitudes and recovery times of the signal disturbances. | SURF, EODC |
| **Real-time reservoir surface water area monitoring** | Deltares | In a real-time application, this use case quantifies the redistribution delays (latency) of Sentinel data offered via the ESA DataHub Relays that are part of the federated infrastructure. | CESNET, VITO |
| **Wetland Water Stress Analysis** | TU Wien | This use case tests the performance of connecting data cubes hosted on distributed infrastructures in a single workflow compared to using data cubes hosted on a single infrastructure. | EODC, CloudFerro/CREODIAS |

Additionally, the User Forum includes an open discussion forum (https://github.com/c-scale-community/discussions) where anyone with an interest in Copernicus and EO data, tools, resources and services can discuss technical issues, ask questions, collaborate, and share ideas.

It should be noted that whilst the use cases have made significant progress in deploying their applications, thanks largely to effective collaboration between the users and the providers to understand the use case requirements, not all of them have completed the use case deployment.

The foci for the use cases for the remainder of the project are

- Generalise workflow for easy redeployment
- Redeploy the use cases to different providers part of the C-SCALE federation
- Test the scalability and performance (e.g., larger areas, long time series)
- Provide documentation (including examples) on how to deploy the use case

## 1.2 Report structure

This feedback report is structured as follows: In Section 2,**Error! Reference source not found.** for each use case, the planned or desired functionality is described, including its software and data dependencies and how these were made available to the use case by the provider, followed by an evaluation of, amongst others:

- speed of access to resources and data
- ease of use of the resources
- appropriateness of the technology used to implement the use case
- resultant usability of the application running on the federated infrastructure
- missing functionality/resources
- effectiveness of support from the providers
- overall satisfaction of the service/resource

For each use case a section on improvement suggestions is included.

The report concludes with a summary of our findings and highlights critical capability gaps for the infrastructure providers to address.

# 2 Use cases

## 2.1 Land surface water change monitoring

### 2.1.1 Functionality

A user can easily deploy a workflow that produces satellite derived surface water changes for a geographic area of interest. The resultant output of the workflow visualises where surface water has become land (due to accretion, land reclamation, droughts) or vice versa where land has become surface water (due to erosion, reservoir construction).

The use case is based on the paper by Donchyts et al. (2016): "Earth's surface water change over the past 30 years". More information about the application can be found at https://www.deltares.nl/en/software/aqua-monitor/. And a Google Earth Engine implementation of the application can be found at https://aqua-monitor.appspot.com/.

Here the application is ported from Google Earth Engine to an open source workflow, leveraging openEO (https://openeo.org/) and tooling available in the C-SCALE federation.

The use case provides the following:

- A Jupyter Notebook containing the openEO-based workflow to derive land-surface changes
- A docker container image to build and run the Notebook on a C-SCALE Cloud IaaS provider

### 2.1.2 Dependencies

This use case is being developed at https://github.com/c-scale-community/use-case-aquamonitor

| ***Dependency*** | ***Implemented by*** | ***Status*** |
|---|---|---|
| *EGI Check-in to access cloud IaaS* | *Provider* | *Users can access the providers' OpenStack interface via EGI Check-in and deploy VMs for their use case.* |
| *Sentinel-2 L1C data* | *Provider* | *Provider has the necessary data available at their site.*<br>***Outstanding issue**: Register metadata in a STAC catalogue to use the openEO backend to process the satellite imagery.* |
| *openEO backend to process satellite data* | *Provider* | *openEO backend has been installed at the provider.*<br>***Outstanding issue**: Register metadata in a STAC catalogue to use the openEO backend to process the satellite imagery.* |
| *STAC catalogue to register metadata (openEO dependency)* | *Provider* | *A prototype STAC catalogue as a Service has been deployed for providers of compute resources in the C-SCALE federation to register their EO metadata.*<br>***Outstanding issue:** Functionality to register metadata in the STAC catalogue is in development: https://github.com/c-scale-community/stac-ingestion* |
| *Docker* | *Provider* | *Cloud container compute, including Docker, is available for users.* |
| *Python libraries and JupyterHub* | *Use case* | *Docker container including dependencies and instructions to build the docker image have been implemented by the use case (https://github.com/c-scale-community/use-case-aquamonitor).*<br>***Outstanding issue**: Publish docker image to AppDB (C-SCALE's software distribution platform) for reuse by other users.* |

### 2.1.3 Feedback

| *Category* | *Feedback* | *Improvement suggestions from users* |
|---|---|---|
| *Access to resources and data* | *1. Access to resources is generally slow, since it requires a relatively complex workflow to register users in EGI Check-in, set up a VO and enable it in Perun.*<br>*2. The different access mechanisms (EGI Check-in for cloud and SRAM for HTC/HPC) add complexity.*<br>*3. Access to data is also slow. There are typically discussions around provider/user responsibilities around sourcing data not available on the federation.* | *1. Make the VO creation and Perun enabling a provider action.*<br>*2. Integrate EGI Check-in and SRAM for harmonised access to heterogenous IaaS.*<br>*3. Data provisioning, including data not yet on the federation, should be a provider action.*<br>*4. Access to resources should be automated and programmable.* |
| *Ease of use of the resources* | *1. OpenStack is a bit complicated to use for users not familiar with it. The process to deploy VMs in OpenStack is different across different providers.* | *1. Implement a simplified and harmonised interface to deploy VMs in the cloud, e.g., SURF's Research Cloud. Or try to harmonise OpenStack VM deployment procedures or provide dedicated trainings.* |
| *Appropriateness of the technology used to implement the use case* | *1. Kubernetes is perhaps the best container orchestrator and can be installed and used on any infrastructure in a reusable manner.*<br>*2. OpenEO is the logical choice for replicating Google Earth Engine (GEE) functionality for Petabyte-scale EO data processing platform. openEO supports parallelization and abstracts infrastructure management away from the user. It is less mature than GEE and during the development of the use case, requests for new features or bug fixes were needed.* | *1. In combination with the mentioned programmable services as described above, having Kubernetes as a Service would allow use cases to easily host web applications and workflows.*<br>*2. A roadmap for the development and support for openEO for the coming years is needed. Investing in openEO in large projects without knowledge on its continuity poses a risk and hinders uptake.* |
| *Resultant usability of the application running on the federated infrastructure* | *The use case will be able to run at any openEO provider.* | *With openEO, porting use cases between providers can be as simple as switching a URL, given that the metadata of the data products is equal. Having an integrated view on metadata of federation datasets will help avoid portability issues.* |
| *Missing functionality / resources* | *Missing functionality within openEO is continuously being communicated at https://discuss.eodc.eu/. Most of the functionality requested are improvements on openEO processes. Custom developments are possible through openEO's user defined functions using python.* | *The documentation on how to work with user defined functions is still sparse. For example, how the DataCube will be represented within your function is unclear.* |

| | | |
|---|---|---|
| *Effectiveness of support from the providers* | *1. The use case is still waiting for the deployment of a STAC catalogue for the data.*<br>*2. The providers have been very proactive in providing support to implement the necessary tooling, e.g., the openEO backend.* | *Automation of requesting compute resources, tools and data is a logical next step.* |
| *Overall satisfaction of the service/resource* | *Using regular meetings and staying in contact, the use case made steady progress. Some of the software was clearly new to the providers, but they were open for experimentation. Continued co-development is encouraged, so that both the users and providers are aware what software and workflows are being used in practice.* | |
| ***Specific feedback*** | | |
| *openEO API* | *The process of testing a workflow is often to download the entire DataCube first and work on it using different tooling (numpy and xarray). It would be nice to have something similar to a .getInfo() statement, to pull some metadata synchronously from the backend, without having to wait a few minutes for the full calculation. See for example https://developers.google.com/earth-engine/apidocs/ee-imagecollection-getinfo* | |
| *Storage of job results* | *Once the openEO job has been finished, the user has little control over how the result is stored. It is therefore sometimes hard to locate and identify previous jobs.* | |

## 2.2 Coastal hydrodynamic and water quality modelling

### 2.2.1 Functionality

A user can easily deploy a workflow that produces hydrodynamic and water quality hindcasts or forecasts for the coastal ocean for a geographic area of interest.

The use case has the following functionality

1. Download the necessary input data for the user's Delft3D Flexible Mesh model setup. Input data include Copernicus' Global Ocean Physics Reanalysis and Global ocean biogeochemistry hindcast, ERA5, ECMWF / GFS and FES2012.
2. Prepare the data for ingestion into the user's Delft3D Flexible Mesh hydrodynamic and water quality model (online coupled). This entails the preparation of forcings, initial conditions, and boundary conditions.
3. Produce hydrodynamic and water quality hindcasts or forecasts based on the user's Delft3D Flexible Mesh hydrodynamic and water quality model setups.
4. Post-process the model outputs by interpolating the unstructured grid output from Delft3D Flexible Mesh to a regular grid for user specified spatial resolution, timesteps, model vertical layers and variables.
5. Visualise the simulation outputs in an interactive Jupyter Notebook.

### 2.2.2 Dependencies

This use case is being developed at https://github.com/c-scale-community/use-case-hisea

| Dependency | Implemented by | Status |
|---|---|---|
| *EGI Check-in to access cloud IaaS* | *Provider* | *Users can access the providers' OpenStack interface via EGI Check-in and deploy VMs for their use case.* |
| *SRAM to access HPC IaaS* | *Provider* | *Users can access HPC resources through SRAM and ssh.* |
| *Delft3D Flexible Mesh Suite, specifically its hydrodynamic module and water quality module (online coupled)* | *Use case* | *Users need to bring their own models and schematisations. There are license dependencies for using the Delft3D Flexible Mesh Suite which are beyond the scope of the providers to facilitate.*<br>*Delft3D Flexible Mesh docker and singularity containers are available for easy deployment on cloud and HPC IaaS (to be obtained from Deltares).* |
| *Data:*<br>• *Copernicus Marine Service Global Ocean Physics Reanalysis data*<br>• *Copernicus Marine Service Global ocean biogeochemistry hindcast data*<br>• *Climate Data Store ERA 5 data*<br>• *ECMWF data*<br>• *GFS data*<br>• *FES2012* | *Use case* | *The data is not available in the C-SCALE federation and is downloaded to the provider from source.*<br>*A docker container to download the data including dependencies and instructions to build the docker image have been implemented by the use case (https://github.com/c-scale-community/use-case-hisea/tree/main/scripts/download).*<br>***Outstanding issues**:*<br>• *Users need to register accounts at the Copernicus Marine Service and Climate Data Store to access the data.*<br>• *Functionality needs to be developed to download ECMWF and GFS forecast data*<br>• *ECMWF forecast data is not free*<br>• *FES2012 data redistribution license states that "The Licensee makes a commitment to not distribute any AVISO+ Product in its original form via any media".*<br>• *Publish docker image to AppDB (C-SCALE's software distribution platform) for reuse by other users.* |
| *docker / singularity* | *Provider* | *Cloud container compute, including docker and Singularity, is available for users (to be obtained from Deltares).* |
| *bash, Python, CDS API, MOTU Client API* | *Use case* | *docker containers including dependencies and instructions to build the docker image are being implemented by the use case (https://github.com/c-scale-community/use-case-hisea)* |
| *Slurm workload manager* | *Provider* | *Slurm workload manager is provided by the HPC IaaS provider* |
| *PaaS Orchestrator* | *Provider* | *PaaS Orchestrator is provided by the provider to easily deploy e.g., Kubernetes clusters* |
| *Kubernetes cluster* | *Provider* | *Kubernetes clusters can be deployed using the PaaS Orchestrator.*<br>***Outstanding issues**:*<br>• *Testing the PaaS Orchestrator and its ability to deploy Kubernetes clusters has been hampered by the availability of additional cloud IaaS resources.*<br>• *Resources required to maintain a Kubernetes cluster are currently beyond the scope of providers to support.* |
| *Argo workflow* | *Use case* | *Argo workflows are planned for implementation by the use case.*<br>***Outstanding issue**: the Argo dependency on the Kubernetes cluster is a bottleneck, and Argo does not support backwards compatibility. Alternatives are being* |

| | | |
|---|---|---|
| | | *investigated ([https://github.com/c-scale-community/discussions/discussions/13](https://github.com/c-scale-community/discussions/discussions/13)).* |

### 2.2.3 Feedback

| *Category* | *Feedback* | *Improvement suggestions from users* |
|---|---|---|
| *Access to resources and data* | *1. Access to resources is generally slow, since it requires a relatively complex workflow to register users in EGI Check-in, set up a VO and enable it in Perun.*<br>*2. The different access mechanisms (EGI Check-in for cloud and SRAM for HTC/HPC) add complexity.*<br>*3. Access to data for this use case was a user action*<br>*4. Group access to HPC requires the use of a jump host (range of IP for an organisation cannot be used)* | *1. Make the VO creation and Perun enabling a provider action.*<br>*2. Integrate EGI Check-in and SRAM for harmonised access to heterogenous IaaS.*<br>*3. Data provisioning, including data not yet on the federation, should be a provider action (for requested datasets).*<br>*4. Access to resources should be automated and programmable.* |
| *Ease of use of the resources* | *1. OpenStack is a bit complicated to use for users not familiar with it. The process to deploy VMs in OpenStack is different across different providers.* | *1. Implement a simplified interface to deploy VMs in the cloud, e.g., SURF's Research Cloud. Or try to harmonise OpenStack VM deployment procedures or provide dedicated trainings.* |
| *Appropriateness of the technology used to implement the use case* | *Except for the Argo and Kubernetes dependencies, for which workarounds are being explored, the technologies used to deploy the use case are appropriate.*<br>*The tests we've run on the HPC using a containerised model (singularity) indicated good performance levels in terms of model run time.* | *Dedicated support for workflow orchestration would be of added value for the use case.*<br>*The use case relies on containerisation technology (docker and singularity). Not many scientists know how to use these. Dedicated training on how to use containers on C-SCALE with workflow examples would be beneficial for users.* |
| *Resultant usability of the application running on the federated infrastructure* | *The individual components of the workflow have been tested and work well on the infrastructure, both cloud and HPC.* | *Support is needed to make the workflow hybrid, i.e., allow the workflow to run the different components across cloud and HPC resources.* |
| *Missing functionality / resources* | *1. The PaaS Orchestrator was proposed as a solution to deploy a Kubernetes cluster, but we were unable to test it due to resource limitations on the cloud provider.*<br>*2. It is unclear how to set up a hybrid workflow leveraging both cloud and HPC in one workflow.* | *1. Providers are encouraged to proactively communicate resource limitations and manage user expectations.*<br>*2. Proactive support and guidance for hybrid (cloud & HPC) workflows is welcomed.* |
| *Effectiveness of support from the providers* | *Generally, support and collaboration from the providers has been satisfactory.*<br>*There are occasions where issues take too long to resolve. It is unclear if this is because of a lack of provider* | *Proactive communication from both the users and the providers is encouraged.*<br>*Additionally, providers are encouraged to proactively manage user expectations, particularly around roles* |

| | | |
|---|---|---|
| | *capacity or if roles and responsibilities between users and providers are not clarified.* | *and responsibilities when using provider resources.* |
| *Overall satisfaction of the service/resource* | *Overall, the resources and available tooling available to the use case is satisfactory and performant.* | *The providers are encouraged to improve their response time to issues and requests.* |

# 2.3 Seasonal river discharge ensemble forecasting

## 2.3.1 Functionality

A user can easily deploy a workflow that produces a monthly high resolution, seasonal, ensemble river discharge forecast for a river basin of interest.

The service has the following functionality, which is automatically started every month when new SEAS5 forecasts become available:

1. Download the necessary input data for the user's WFLOW hydrological model setup. Input data are the ERA5 reanalysis and the SEAS5 seasonal forecast.
2. Prepare the data for ingestion into the user's WFLOW hydrological model. This includes the preparation of forcing fields, initial conditions and boundary conditions.
3. Produce a 50 member ensemble forecast based on the user's WFLOW hydrological model setup.
4. Visualise the forecast in an interactive Jupyter Notebook displaying river discharge timeseries and interactive maps of soil moisture anomalies (in development).

## 2.3.2 Dependencies

This use case is being developed at https://github.com/c-scale-community/use-case-high-res-land-surface-drought-analysis

| ***Dependency*** | ***Implemented by*** | ***Status*** |
|---|---|---|
| *SRAM to access HTC IaaS* | *Provider* | *Users can access HTC resources through SRAM and ssh.* |
| *WFLOW hydrological model* | *Use case* | *Users need to bring their own models and schematisations.* |
| *Data:*<br>• *Climate Data Store ERA5 data*<br>• *Climate Data Store SEAS5 seasonal forecast data* | *Use case* | *The data is not available in the C-SCALE federation and is downloaded to the provider from source. Functionality has been developed to download the necessary data (https://github.com/c-scale-community/use-case-high-res-land-surface-drought-analysis).* |
| *crontab* | *Provider* | *crontab is provided by the provider.* |
| *bash, Python, Julia* | *Use case* | *bash is made available on the login node by the provider. Python and Julia and necessary libraries are installed on the login node by the use case with support from the provider.* |

### 2.3.3 Feedback

| *Category* | *Feedback* | *Improvement suggestions from users* |
|---|---|---|
| *Access to resources and data* | *Access to the resources was easy due to good documentation. Data was provided by the user and internal download scripts.* | *It would be interesting to have batch data loading services available when porting data from internal systems or from commercial cloud sources.* |
| *Ease of use of the resources* | *The HTC environment was easy to use as there was sufficient documentation available. The internal SLURM cluster was configured according to best practices. This helps researchers who already have experience with SLURM.* | *Exposing the results from the workflow is currently performed using an automated Apache http server. For future projects, we are looking forward to trying to expose some API via federation services.* |
| *Appropriateness of the technology used to implement the use case* | *The workflow used in the use case is constructed using crontab and chaining bash scripts together. This is appropriate for simple workflows.* | *Complex workflows may require a more advanced workflow system, such as snakemake (https://snakemake.readthedocs.io/en/stable/). This was suggested by the provider.* |
| *Resultant usability of the application running on the federated infrastructure* | *The software used in this use case is containerized using Singularity. Access to the data is, however, hardcoded to the HTC infrastructure. Packaging a standard generic data structure requires packaging the workflow itself into a container with mount paths. We are still in the process of figuring out a good way of doing this.* | *From a user perspective, it would be interesting to see how other projects have packaged and automated their workflows. Having some example projects available in GitHub or another repository type would help speed up this use case.* |
| *Missing functionality / resources* | *The crontab file defining the scheduling is currently linked (and stored) to a specific user account, meaning that only that user can view and/or edit this file* | *It would be nice if multiple users in a project group would be able to change the schedule. Alternative software to crontab could be considered.* |
| *Effectiveness of support from the providers* | *The provider has been great in answering any questions and thinking about solution with the use case.* | *It would be welcomed that the provider offered a "getting started" guide to its service portfolio.* |
| *Overall satisfaction of the service/resource* | *Implementation of this use case was quicker than expected. Communication with and support the provider was swift and effective.* | |

## 2.4 Monitoring tropical forest recovery capacity

### 2.4.1 Functionality

This use case utilizes time series of Sentinel-1 (S-1) backscatter intensity (Sigma0) images to monitor tropical forest recovery capacity over the Amazon basin. Sentinel-1 is a C-band, synthetic aperture RADAR imaging mission consisting of two satellites (S-1A and S-1B) operating together since 2016 and acquiring images globally, with two polarizations channels (VV and VH), a spatial resolution of 20 × 5 $m^2$ (in the interferometric wide-swath, IW, mode), and a revisiting time of maximum 6 days. A C-SCALE data provider, Earth Observation Data Centre (EODC) GmbH, has provided the S-1 backscatter intensity datacube over the Amazon basin for this use case (Wagner et al. 2021). For each location (pixel) of the datacube, a 5-year-long S-1 backscatter intensity time series (2017–2021) has been analysed to detect the magnitude and recovery time of the signal disturbance. The analysis was based on an algorithm that combines the original and smoothed time series and the statistical properties of the first- and last-year or the S-1 data to detect the signal disturbances.

This use case aims to (a) derive Amazon-wide disturbance magnitude and recovery time maps from the Sentinel-1 image time series and (b) analyse the functional relation between derived magnitude and disturbance time.

### 2.4.2 Dependencies

This use case is being developed at https://github.com/c-scale-community/use-case-return

| *Dependency* | *Implemented by* | *Status* |
|---|---|---|
| *Sentinel-1 ARD data* | *Provider* | *The data provider made the data available for user.* |
| *Python, conda* | *Use Case* | *Python anaconda environment has been installed on the login node by the use case with support from the provider.* |
| *yeoda module* | *Use Case* | *The python yeoda module has been installed in a separate conda environment by the use case with support from the provider.* |
| *Jupyter Lab* | *Use Case* | *The Jupyter lab has been installed in the yeoda environment by the use case with support from the provider.* |
| *User-defined Python code* | *Use Case* | *Python code has been prepared to process and analyse the Sentinel-1 time series by the use case.* |

### 2.4.3 Feedback

| *Category* | *Feedback* | *Improvement suggestions from users* |
|---|---|---|
| *Access to resources and data* | *1. The user had to make a licence agreement with the data provider. Such a procedure may take longer as the agreement had to go through the approval at the user side, which is odd considering the user is working in an open science cloud.*<br>*2. The data had to be transferred from the data provider to the compute-resource provider.*<br>*3. Yeoda module installation was complicated because of its own dependences.* | *1. Currently no suggestion on how this can be improved.*<br>*2. Data/resource providers may automate the data transfer between one another.*<br>*3. As yeoda module works with the data- provider data (eodc data cubes), it can be considered that this module, i.e., the corresponding conda environment, is already available at all C-SCALE providers.* |
| *Ease of use of the resources* | *Jupyter Lab has been used to further develop the code and access the data / computing resources. This requires several ssh connections to run Jupyter Lab and accesses it via local browser.* | *Provider may consider offering Jupyter Lab as a service directly to the user.* |
| *Appropriateness of the technology used to implement the use case* | *The technology is appropriate to implement the use case. A clear advantage is the flexibility with onboarding the user-defined tools, code, and analysis, which may be problematic on some Earth Observation data platforms such as GEE.* | |

| | | |
|---|---|---|
| *Resultant usability of the application running on the federated infrastructure* | *The use case is possible to deploy thanks to the availability of Sentinel-1 data and a flexible way of applying user code in the infrastructure. Although the deployment required some small adaptations of the code, the support from the providers was good and helped to develop appropriate code wrappers.* | *The providers might consider automating/abstracting for the user the code-wrapping process, e.g., with an openEO backend, or other tools.* |
| *Missing functionality/resources* | *As mentioned above, providers may consider offering Jupyter Lab as a service with preinstalled EO python packages such as (geo)pandas, xarray, yeoda etc.* | |
| *Effectiveness of support from the providers* | *The support from the providers was quick and effective. This is another advantage of the C-SCALE approach because it allows to develop customized solutions with the support from providers.* | |
| *Overall satisfaction of the service/resource* | *The use case has not yet been fully deployed, but so far I was very satisfied with the support in the implementation of the use case and the flexibility the C-SCALE approach offers in terms of user-defined code and processing.* | |

## 2.5 Real-time reservoir surface water area monitoring

### 2.5.1 Functionality

A user can easily deploy a workflow that produces real-time satellite derived surface water area estimates for a user's geographical area of interest.

The service has the following functionality:

1. Access Sentinel-2 L1C data for a user-defined geographical area of interest
2. Based on a database of known reservoirs, the algorithm finds reservoirs in the user-defined geographical area of interest and estimates aggregated surface water areas from all known reservoirs contained within the user-defined geographical area of interest.
3. Return a .csv file containing surface water area data and associated statistics.

### 2.5.2 Dependencies

This use case is being developed at https://github.com/c-scale-community/use-case-waterwatch

| ***Dependency*** | ***Implemented by*** | ***Status*** |
|---|---|---|
| *EGI Check-in to access cloud IaaS* | *Provider* | *Users can access the providers' OpenStack interface via EGI Check-in and deploy VMs for their use case.* |
| *Sentinel-2 L1C data* | *Provider* | *Provider has the necessary data available at their site.* ***Outstanding issue****: Register metadata in a STAC catalogue to use the openEO backend to process the satellite imagery.* |
| *JRC water occurrence data* | *Provider* | *Provider will make the necessary data available at their site.* |

| | | |
|---|---|---|
| *openEO backend to process satellite data* | *Provider* | *openEO backend will be installed at the provider.* |
| *STAC catalogue to register metadata (openEO dependency)* | *Provider* | *A prototype STAC catalogue as a Service has been deployed for providers of the C-SCALE federation to register their EO metadata.*<br>***Outstanding issue**: Functionality to register metadata in the STAC catalogue is in development: https://github.com/c-scale-community/stac-ingestion* |
| *Python libraries and JupyterHub* | *Use case* | *Docker container including dependencies and instructions to build the docker image are being implemented by the use case (https://github.com/c-scale-community/use-case-waterwatch).*<br>***Outstanding issue**: Publish docker image to AppDB (C-SCALE's software distribution platform) for reuse by other users.* |

### 2.5.3 Feedback

| ***Category*** | ***Feedback*** | ***Improvement suggestions from users*** |
|---|---|---|
| *Access to resources and data* | *1. Even for the providers, the process of installing the openEO backend on the provider side remains non-trivial and requires significant expert support and effective communication. Challenges faced by the providers around installing the openEO backend have caused significant delays for the use cases dependent on openEO.*<br>*2. For our project, we need to expose data via a STAC catalogue. Setting this up and ingesting data is not yet completed.* | *1. Leading cloud providers offer infrastructure as a REST service, which is standardized and programmable. If a user can program against the C-SCALE infrastructure, the reproducibility and start up time for a project will be greatly improved.*<br>*2. Once available, the Metadata Query Service will help finding and standardizing data storage and retrieval. Additionally, offering data via well-known geospatial interfaces such as a STAC catalogue will also help use cases find and access data* |
| *Ease of use of the resources* | *1. Once you have an account registered using EGI Check-in, getting access to compute resources using openEO is fully automated.*<br>*2. Once a STAC catalogue is available, finding and using data within the OpenEO framework requires only a few lines of code* | *1. For the openEO batch system, instead of relying on the generic logic built into the backend, it would be nice to be able to specify the resources to reserve for the job.*<br>*2. Standards and instructions on how to request and integrate new data sources in openEO are missing.* |
| *Appropriateness of the technology used to implement the use case* | *1. Kubernetes is perhaps the best container orchestrator and can be installed and used on any infrastructure as a reusable manner.*<br>*2. OpenEO is the logical choice for replicating Google Earth Engine* | *1. In combination with the mentioned programmable services as described above, having Kubernetes as a Service would allow use cases to easily host websites and workflows.* |

| | | |
|---|---|---|
| | *(GEE) functionality for Petabyte-scale EO data processing. openEO supports parallelization and abstracts infrastructure management away from the user. It is less mature than GEE and during the development of the use case, requests for new features or bug fixes were needed.* | *2. A roadmap for development and support for openEO for the coming years is needed. Investing in openEO in large projects without knowledge on its continuity poses a risk.* |
| *Resultant usability of the application running on the federated infrastructure* | *The use case still needs some bugfixes on the openEO side. Once fully implemented, it will be able to run at any provider.* | *Porting use cases between providers can be as simple as switching a URL, given that the metadata of the data products is equal. Having an integrated view on metadata of common datasets will help avoid portability issues.* |
| *Missing functionality/resources* | *Missing functionality within openEO is continuously being communicated at https://discuss.eodc.eu/. Most of the functionality requested are improvements on openEO processes. Many workarounds exist using user defined functions in python.* | *The documentation on how to work with user defined functions is still sparse. For example, how the DataCube will be represented within your function is unclear.* |
| *Effectiveness of support from the providers* | *1. The main dependency of the project is the STAC catalogue, which is still in development.<br>2. Optical satellite data needed for the use case is only available for the country that the provider is located in, i.e., typically only national data is available.<br>3. For deployment of the OpenEO backend, support from the providers is required.* | *Support for the use cases would be more effective if all the use case relevant data is available directly (or second best, after staging), including data beyond the providers national geographic boundaries.<br>Automation of requesting resources and data is a logical next step.* |
| *Overall satisfaction of the service/resource* | *Using regular meetings and staying in contact, the use case made steady progress. Some of the software was clearly new to the providers, but they were open for experimentation. More co-development is encouraged, so that both the users and providers are aware what software and workflows are being used in practice.* | |
| ***Specific feedback*** | | |
| *Discussion forum* | *The discussion forum: https://discuss.eodc.eu/ is a great way to stay close to the developers and get issues resolved quickly. It assisted in getting bugs and bottlenecks out of the way.* | |

# 2.6 Wetland Water Stress Analysis

## 2.6.1 Functionality

Goal of this use case is the identification of wetland worth for protection. NOAA revealed that wetland behaviour has a major effect on climate change, acting as potential sink or source of methane – depending on its health. While $CO_2$ emissions are currently very dominant in the public perception, emitted $CH_4$ is around 25 times as powerful in trapping heat in the atmosphere. Because it does not stay in the atmosphere as long, it more has a short-term influence on the rate of climate change. According to NOAA, "reducing methane emissions is an important tool we can use right now to lessen the impacts of climate change in the near term, and rapidly reduce the rate of warming" [https://www.noaa.gov/news-release/increase-in-atmospheric-methane-set-another-record-during-2021]. Wetlands have been one of the major drivers of methane in the atmosphere, acting as source while not being stable. This includes water stress as well as renaturation. The only state of wetland acting as a methane sink is a healthy wetland patch.

Since 1975, the RAMSAR convention identified many regions globally as protected wetlands. Many other regions also fulfil the required conditions but are currently not defined as protected area. At the same time, governmental programmes exist throughout Europe to identify new regions worth for a renaturation of wetlands – following the European Green Deal. Since a renaturation does lead on short-term to a methane source, it is more effective to identify existing, but still not protected wetland areas for future management. This use case aims for identifying such potential areas, making use of Copernicus' Sentinel-1 and –2 data and machine learning approaches.

Therefore, Sentinel-1 and -2 time-lines of RAMSAR areas throughout Europe are analysed and compared to their surrounding (following the assumption that the direct surrounding are affected by similar environmental influences and consist of similar land cover). Afterwards the identified characteristics are compared with other Natura 2000 areas to identify similar, but currently not protected areas. CORINE Land Cover (2018) information are used as additional information on the diverse land cover types of wetlands.

The machine learning workflow runs on premises of CloudFerro and of EODC, currently using provided CPUs. For optimising the workflow, a shift to GPUs is planned, depending on availability at the providers. It will be analysed, which input data are relevant for identifying / characterising wetlands / wetland water stress. Output is a segmentation / classification of intact wetland areas similar to the requirements of the RAMSAR convention. A stretch goal is the conduction of an investigation on statistical markers of intact wetland surfaces, by using machine learning tools. The usability of the approach will be validated for detected regions in Austria with the help of Austrian governmental users that intend to identify and to monitor new national wetland regions.

### 2.6.2 Dependencies

This use case is being developed at https://github.com/c-scale-community/use-case-wetland-water-stress.

| *Dependency* | *Implemented by* | *Status* |
|---|---|---|
| *CREODIAS login to access cloud IaaS* | *Provider* | *Users can access the providers' OpenStack interface via CREODIAS and deploy VMs for their use case.* ***Outstanding issue:*** *integrate CREODIAS with EGI Check-in* |
| *Shared volumes* | *Provider* | *Data can be stored, organized shared between VMs using dedicated SSD space* |
| *Exploring provided Sentinel-1 data and RAMSAR area to make* | *Use case* | *A Jupyter Notebook has been implemented to explore the Sentinel-1 data provided by CREODIAS with respect to RAMSAR regions of interest.* |

| | | |
|---|---|---|
| *informed decisions on which models to select* | | *Reusable code modules have been extracted, to load and display data. These can be reused by the EODC version of the Notebook.*<br>***Outstanding issue**: Doing the same for EODC data, where we encountered access problems during implementation* |
| *Pre-process data suitable for applying selected models* | *Use case* | *Common transformation modules have been implemented to extract and pre-process data provided. These can be reused across different providers.*<br>*A Jupyter Notebook for pre-processing Sentinel-1 data has been provided by CREODIAS*<br>***Outstanding issue**: Notebook for EODC data, as we encountered access problems during implementation.* |
| *Testing different models* | *Use case* | *Simple models have been tested; the pytorch framework has been investigated to use for machine learning approaches.*<br>***Outstanding issue**: training more complex models using pytorch, using also GPU resources* |
| *Access to 3rd Party Services* | *Provider* | *Floating IPs are provided to give outside services such as Github actions access to run our CI/CD pipeline, as well as publishing dedicated docker images for testing/deploying.* |
| *CI/CD and automated tests* | *Use case* | *Automated tests for common software modules have been implemented as well as docker image files containing the dependencies.*<br>***Outstanding issue**: hooking it up to Github Actions to run tests and deploy automatically.* |

### 2.6.3 Feedback

| ***Category*** | ***Feedback*** | ***Improvement suggestions from users*** |
|---|---|---|
| *Access to resources and data* | *1. On CloudFerro it is not immediately clear what the actual best practise to access data is. Several existing methods for reading and inspecting buckets use S3 data as default; also, examples of accessing S-1 and S-2 data would be helpful.*<br>*2. The EODC storage experienced some mayor problems during our implementation, however staff at EODC was quick to response.* | *1. Extend the EODATA FAQ with some very simple examples that shows best practise on how to read a satellite image – maybe even cut out a ROI using geo-referenced coordinates, similar to the robo3 examples. This would make it immediately clear how to get access to data.*<br>*2. Hard to improve as unforeseen things like this happen, so it was good we had access to a 2nd provider to continue our work* |
| *Ease of use of the resources* | *Both providers use OpenStack as its base making it easy to transfer knowledge between setups.* | *No real complaints, the EODC web page is a bit unresponsive at times, however, since in our use case we do not spend much time orchestrating VMs this is negligible.* |
| *Appropriateness of the technology used to implement the use case* | *OpenStack is every flexible and we can tailor the infrastructure to our needs. However, since we mostly process data "offline" and are not providing a live service with fluctuating demands,* | *Provide more VM flavours so we are even more flexible in resource assignment, allowing us to tailor individual nodes to our needs.* |

| | | |
|---|---|---|
| | *the kubernetes and cluster features are overkill for us. We'd benefit more from few but relatively powerful nodes, than from orchestrating many small nodes.* | |
| *Resultant usability of the application running on the federated infrastructure* | *Being able to quickly host a Jupyter Notebook and making it accessible via a web interface, makes it easy to demonstrate our research findings* | *Providing some more ready-made security groups for Jupyter Notebooks, tensorboard access or HTTP/S, similar to the "allow_ping_ssh_icmp_rdp" group would probably make it easier for less dev-ops savvy users.* |
| *Missing functionality/resources* | *For more complex models and especially for applying machine learning or machine vision to our raster data we'd need GPU resources.* | *Provide the option to connect, an NVIDIA GPU. The CREODIAS documentation suggest that they already provide VMs with GPU support, so this might just be a matter of requesting these resources.* |
| *Effectiveness of support from the providers* | *1. Cloud-Ferro has excellent support staff, who reacted within the hour to our urgent requests. Also, when making requests for additional HW-Flavours I got a response within minutes.*<br>*2. We encountered problems accessing the EODC storage, and support stuff replied to us immediately, working on the issue. They even provided us with a status page where we could check the progress of the issue* | *---* |
| *Overall satisfaction of the service/resource* | *OpenStack seems to become the de-facto standard for IaaS platforms, making it easy to get something running quickly. When there was still a problem support staff was quick to respond on CloudFerro as well as EODC side* | *Documentations could do with more examples and how-to guides* |
| ***Specific feedback*** | | |
| *CREODIAS / CloudFerro VM Images* | *The standard images report the correct user for the ssh login when the wrong one is used. This error handling saved us some time to figure it out on our own.* | |
| *OpenStack error messages* | *Some errors reported by the OpenStack platform are quite uninformative. More informative error messages would probably allow us to fix minor issues ourselves without involving support staff.* | *Investigate if the error reporting can be improved including more information about the cause, without compromising the security of the platform* |

# 3 Summary of improvement suggestions from users

The improvement suggestions proposed by the use cases are summarised in the below table and organised along three categories:

- Technology / feature requests
- Access and resource provisioning
- Support, documentation and training

The improvement suggestions in Table 2 are not organised in any particular order.

*Table 2. Summary of improvement suggestions*

| ***Technology and feature requests*** | ***Access and resource provisioning*** | ***Support, documentation and training*** |
|---|---|---|
| *Interface (wizard) to simplify and automate compute, data and storage provisioning (Figure 1).* | *Simplify access: Harmonise Cloud and HPC/HTC AAI. Make VO registration and Perun enablement a provider action* | *Documentation in general needs to be improved and extended* |
| *Training for programmable access to resources should be provided – i.e., offer users training to build Infrastructure as Code* | *Make data provisioning an (automated) federation action and extend the available datasets in the federation based on user requests.* | *Provide documentation, examples and training for workflow orchestration using different tooling available on C-SCALE* |
| *Kubernetes as a Service to support web applications and workflow development* | *Find a federation-wide solution for accessing data from source and redistributing these in the federation. E.g., CMEMS and CDS require user registration for access.* | *Provide documentation, examples and training for hybrid (cross cloud-HPC/HTC) workflow deployment* |
| *Batch data loading system for porting data from internal systems or from commercial cloud* | | *Provide documentation, examples and training for containerisation of applications* |
| *Expose application results via API instead of Apache http server.* | | *Provide documentation, examples and training on automation and packaging of applications* |
| *EO / Copernicus specific Notebooks as a Service* | | *Improve response time of requests for support* |

Judging from the number of times "access and resource provisioning" was mentioned by the use cases as an improvement suggestion for the federation, it could be considered as an important feature to focus on in future iterations of the C-SCALE federation (perhaps in a C-SCALE follow up project).

Figure 1 describes a proposed workflow for users requesting data, compute and storage resources from the federation. It is proposed that the workflow is managed via a simple (web)interface, where users follow a wizard to provide the data needed to specify user requirements and the federation in the backgrounds sets up the compute and storage resource on an appropriate federation provider. Ideally the federation would set up the compute and storage resources automatically for the users including making the requested data and tooling available.

Having data and tooling already available on the compute resource when a user logs in is a major benefit for users and will enable them to focus on science.

It is understandable that developing such a capability poses significant challenges for the federation, especially around fetching data that is not yet in the federation and deploying specific tooling, both of which require collaboration between providers and users with the relevant expert knowledge. If the workflow in Figure 1 allows for users to specify missing data and tooling, then over time, as more users make use of the federation, the federation's data catalogue will grow.

The below workflow could be piloted using online forms without the need for it to be fully automated.

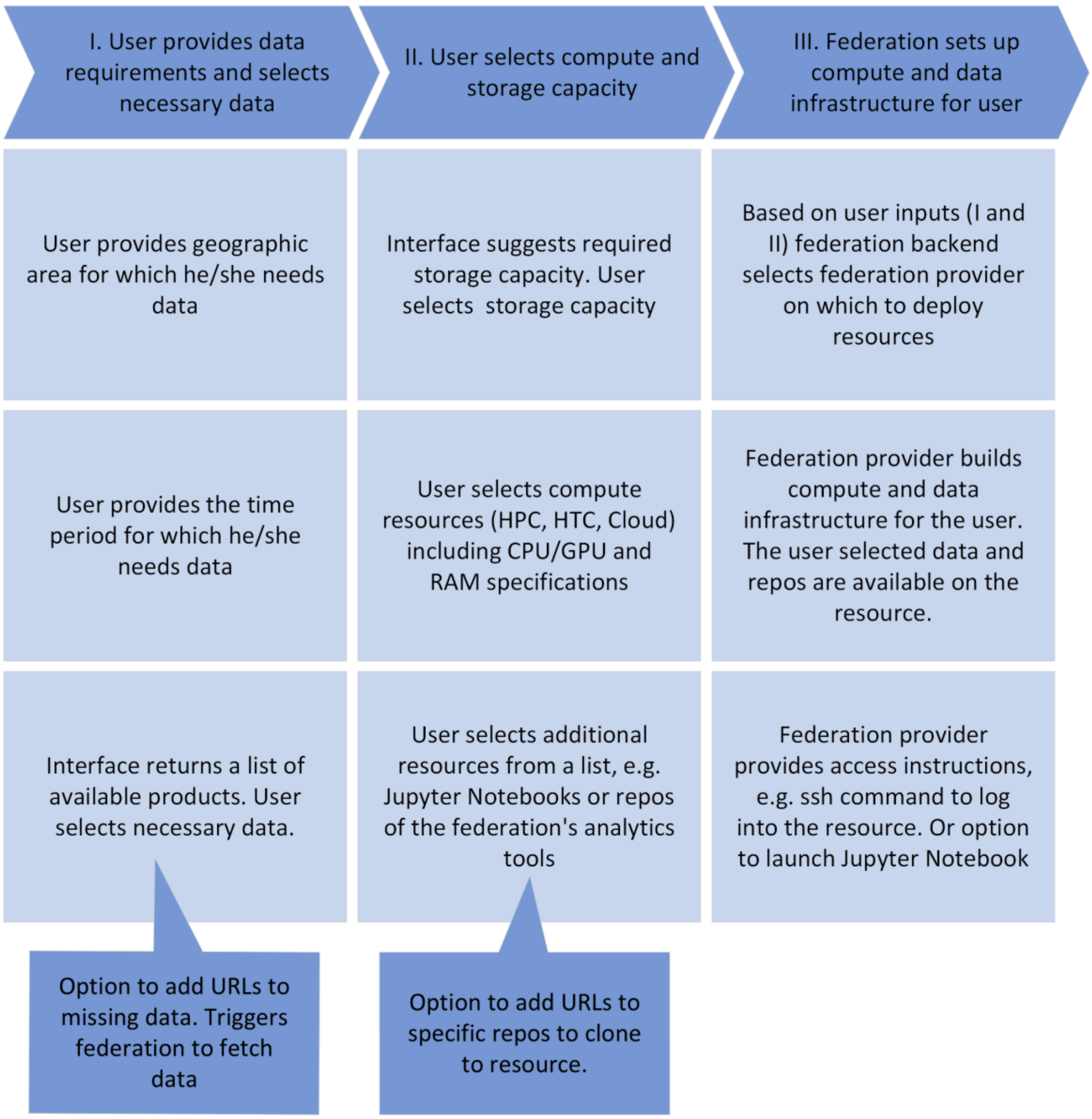


*Figure 1. Proposed workflow towards automated provisioning of compute and data resources for users*

# 4 Conclusion

In general, the feedback from the use cases is positive. Users recognise the value of developing customised solutions in collaboration with the federation providers and appreciate the flexibility of the resources being provided.

Support from the providers is generally quick and effective, but in some cases could be improved. Here proactive communication is key, but communication goes both ways, it is also up to the users to proactively ask for help.

It was found that regular meetings and setting feasible objectives (working in a scrum-like way) to achieve between meetings were an effective way to progress collaboratively on the use case deployments. Continued co-development is encouraged, so that both the users and providers are aware what technologies and workflows are being used in practice.

From the use cases, the main improvements proposed for the federation centre around:

- Simplifying and harmonising resources provisioning and access
- Improving documentation and providing examples and training
- Finding mechanisms to ensure that data and tooling are readily available for users so that they can focus on science

Finally, it should be noted that some of the technologies needed for the use cases were clearly new to the providers. The providers should be commended for their willingness to explore new technologies and find solutions to support the use case deployment.