Business Report: Using the ESA Network of Resources with Research Platforms (example: JASMIN)

Version History

Version number	Authors	Date
0.1	Ag Stephens (AS), Matt Jones and Alex Manning (CEDA)	20/12/2023
0.2	AS	27/03/2024
1.0	AS	28/03/2024

1 Contents

2 Abbreviations and Acronyms

ADES:	Application, Deployment and Execution Service
CEDA:	Centre for Environmental Data Analysis
CWL:	Common Workflow Language
EOEPCA:	Earth Observation Exploitation Platform Common Architecture
ESA:	European Space Agency
ESACCI:	ESA Climate Change Initiative
JASMIN:	Super-data-cluster managed by CEDA (no acronym)
LDCs:	Least Developed Countries
LOTUS:	The JASMIN Batch compute cluster
NoR:	Network of Resources
OGC:	Open Geospatial Consortium
RP:	Research Platform
UKRI-STFC:	United Kingdom Research and Innovation (UKRI) – Science and Technology Facilities Council
USP:	Unique Selling Point
WES:	Workflow Execution Service
WPS:	Web Processing Service

3 Introduction and Context

This report is part of a project funded by ESA, through Telespazio Vega UK, as a vehicle to support uptake of operations of the Earth Observation Exploitation Platform Common Architecture (EOEPCA). The work was carried out by the Centre for Environmental Data Analysis (CEDA[1]), which is part of United Kingdom Research and Innovation / Science and Technology Facilities Council (UKRI-STFC). CEDA runs the JASMIN[2] data and compute cluster that primarily supports scientific research conducted under the Natural Environment Research Council (NERC). This report focusses on Task E4 of the contract ("Report on deployment of EOEPCA within a research platform: implications and findings") and Task E5 ("Investigate the integration of ADES with a batch processing environment to enable the scaling up of processing tasks").

CEDA manages a collection of data centres and services that are hosted on JASMIN. The primary communities of users are made up of the NERC centres: National Centre for Atmospheric Science (NCAS) and National Centre for Earth Observation (NCEO). JASMIN provides a comprehensive set of storage, data and processing services, including:

- Lotus: The batch computing cluster, allowing for high-throughput and parallel computing.
- Group Workspaces: Collaborative storage areas for projects and consortia to share datasets and work collaboratively.
- Cloud Computing: Virtual machine (VM) services enabling users to host their applications and services.
- CEDA Archive: Hosts a vast collection of environmental science datasets accessible for research.
- Data Transfer Services: Tools and services like GridFTP for efficient data movement.
- Cylc Suite Engine: For managing and automating workflow suites.
- Scientific Analysis Servers: Interactive computing resources for data analysis.
- JASMIN Accounts Portal: Management of user accounts, group memberships, and access to resources.
- Notebook Service: Provides Jupyter notebooks as a service for interactive data analysis.
- Cluster-as-a-Service (CaaS): Enables users to deploy their own computing clusters within JASMIN's cloud infrastructure.

JASMIN is mainly provided to support academic research, but extends to support the work of organisations, such as the Met Office, that have a strong collaborative link with the atmospheric research community. During its lifetime, JASMIN has supported a range of high-profile projects and datasets, including the 6th Climate Model Intercomparison Project (CMIP6), the EU Horizon Europe PRIMAVERA project, the ESA-CCI Open Data Portal and a multi-petabyte store of Sentinel satellite products.

As a Research Platform (RP), JASMIN is used in many modes, for example:

1. A scientist logs in via SSH and runs their own code against terabytes of climate/EO data.
2. A team of scientists develop and run a data-processing model and generate a new product that is published to the CEDA archive and is minted with a DOI.
3. An international project uses JASMIN to store data and builds tools to optimise access to that data.
4. A project provides its own web-tool as an interface to existing data (on JASMIN): using the JASMIN cloud, deploying on their own Kubernetes cluster (managing their own users and access rules).
5. A scientist logs into the JASMIN Notebook Service to develop a data-driven notebook that will accompany a scientific paper to explain the workflow.
6. A University is running a training event (such as a Hackathon) and uses JASMIN training accounts to provide 50 participants with temporary access to JASMIN resources.

EOEPCA, funded by the European Space Agency (ESA), follows the model of data exploitation that occurs in hosted environments with co-located computing and storage ("bring the user to the data"). This results in a platform-based ecosystem that delivers infrastructure, data, compute, and software as a service, enabling scientific and value-added activities that produce targeted outputs for end-users. EOEPCA is a Reference Architecture built of free and open-source

[1] CEDA: https://ceda.ac.uk
[2] JASMIN: https://jasmin.ac.uk

components that can be deployed in cloud-native environments. At the heart of EOEPCA is the Application and Deployment Execution Service (ADES) which allows users to create, publish, deploy, discover and execute bespoke code in a containerised environment.

The ESA **Network of Resources (NoR)** provides a catalogue of EO data, processing, training and other resources hosted in cloud environments by a range of providers The NoR initiative offers support to research, development, and pre-commercial users, encouraging them to innovate their working. By providing an ecosystem of EO resources that includes data, software, applications, and IT services in the cloud, the NoR seeks to foster a more efficient and effective use of EO data. The NoR includes a sponsorship program, which provides vouchers of up to €5,000 to help users employ cloud computing services for geospatial data and algorithms.

4 Overview of the JASMIN EOEPCA developments

4.1 Deployment of EOEPCA on the JASMIN cloud

A tenancy was created in the JASMIN cloud environment to allow the CEDA team to install, test and develop the JASMIN EOEPECA instance. This was built on top of a Kubernetes "Cluster-as-as-Service" offering provided by the JASMIN cloud infrastructure. A vanilla install of EOEPCA was installed using the recipes provided by the EOEPCA help guides, and with expert help from the EOEPCA team.

4.2 Integrating the Slurm scheduler and the ADES

A key part of the work was to investigate using the ADES with a scheduling tool rather than using Kubernetes to control and execute workflows. The potential advantages of this would be:

- Enabling the deployment of large workflows (i.e. those that might require multiple nodes and more complex compute environments).
- The potential to re-use pre-installed software environments rather than deploying the software for each execution.
- The possibility of connecting to pre-existing processing clusters, such as LOTUS on JASMIN, and executing workflows on them.

The work involved various components, as follows:

1. Preparation of a command-line tool (see next section) to provide remote subsetting of example datasets held on JASMIN (e.g. ESACCI data), to be deployed and executed through the ADES.
2. Integration of the ADES scheduling and deployment components with the Slurm scheduler, using a Workflow Execution Service (WES).
3. Deployment of a Slurm cluster within the cloud tenancy.
4. Integration with the TOIL tool for executing CWL workflows as Slurm jobs (see Appendix 1 for more details). This was managed using a plugin for the Zoo Project.
5. Integration with the Singularity tool to enable Docker files to be converted into containers that can run with user (rather than *root*) permissions.
6. Testing of the integration with the prototype "snuggs" application.
7. Integration and testing with the CEDA subsetting tool.

Details of the integration work are documented at:

https://github.com/cedadev/eoepca-zoo-wes-runner/tree/main/docs

This explains how TOIL is configured and deployed alongside Slurm and Singularity. This enables a full integration with the ADES so that workflows are deployed as follows:

1. The job is described in command using the `toil-cwl-runner`, e.g.:

```
toil-cwl-runner \
 --maxMemory 10Gib \
 --batchSystem slurm --singularity \
 --workDir ~/daops-work/work_dir \
 --jobStore ~/daops-work/job_store/$(uuidgen) \
 ~/daops-work/daops/app-package.cwl#daops \
 ~/daops-work/daops/cli-params.yml
```

2. The TOIL runner contacts the Slurm scheduler to schedule the job.
3. When Slurm is ready to execute the job, it invokes the instructions in the `app-package.cwl` workflow description which explains the command-line tool signature, the processing environment and the Dockerfile required for the application.
4. Singularity is then invoked to pull and build the container image (or use a locally cached version).

5. The input parameters are described in the `cli-params.yml` file, e.g.:

```
---
area: "30,-10,65,30"
time: "2000-01-01/2000-02-30"
time_components: ""
levels: "/"
file_namer: "simple"
output_dir: "."
collection: "https://data.ceda.ac.uk/neodc/esacci/cloud/metadata/kerchunk/version3/
             L3C/ATSR2-AATSR/v3.0/
             ESACCI-L3C_CLOUD-CLD_PRODUCTS-ATSR2_AATSR-199506-201204-fv3.0-kr1.1.json"
```

6. The Toil runner then invokes the command, with the required inputs, into the container, and the job is executed.
7. Finally, the outputs are staged out to the user workspace.

A template repository is also provided as a starting point for the creation of new WES using TOIL:

https://github.com/cedadev/eoepca-proc-service-template-wes

4.3 Development of the "daops" subsetter and deployment to the ADES

The "daops" library[3] is part of the "roocs" framework[4] developed by CEDA and DKRZ to provide subsetting and processing capabilities for climate simulation data for the Copernicus Climate Change Service Climate Data Store.

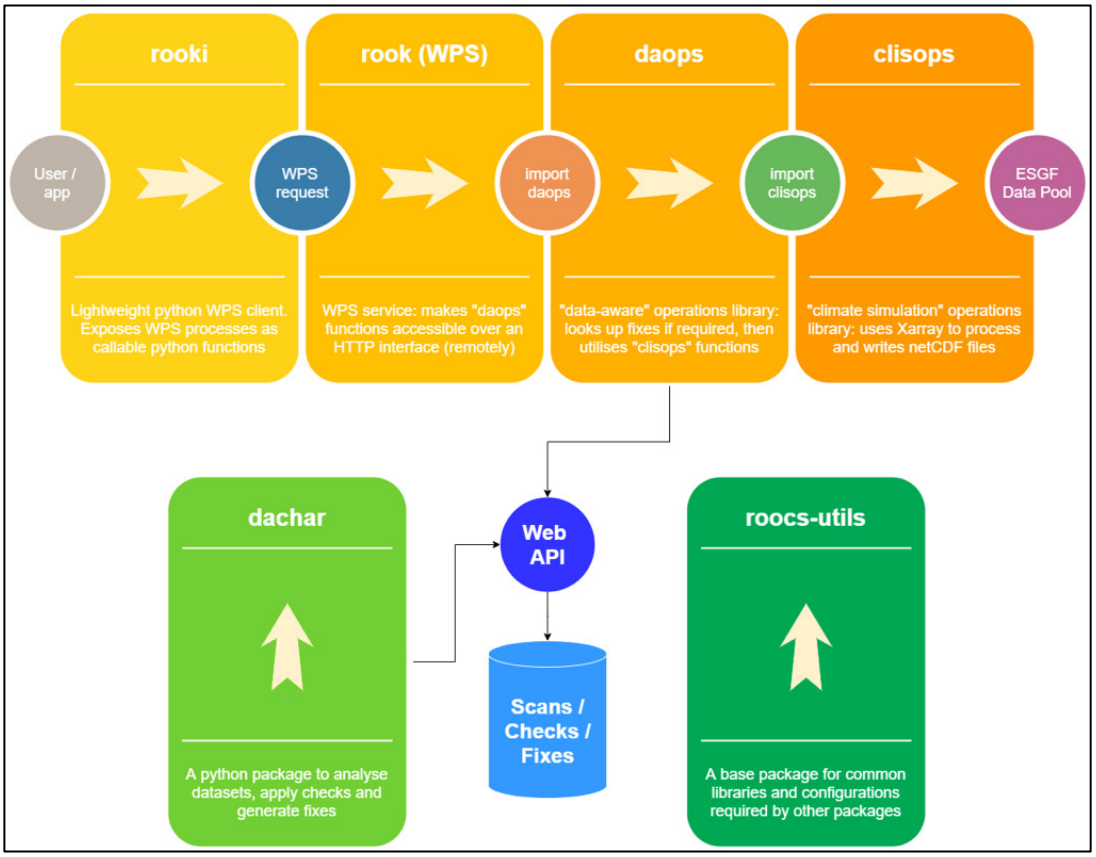

Figure 1. The "roocs" framework for climate data processing.

[3] daops: https://github.com/roocs/daops
[4] roocs: https://roocs.github.io/

“daops” stands for *data-aware* operations on climate simulations. It serves 3 main functions:

- Providing an **aggregation layer** that allows the client/user to refer to *datasets* (via meaningful identifiers) rather than individual files, thereby simplifying data usage.
- Enabling **“hot-fixes” to datasets** where errors exist in the source data/metadata, that can be fixed on-the-fly (at processing time).
- Providing a **command-line tool** to simplify access to **basic operations such as subsetting and averaging**.

In order to integrate the “daops” tool with the ADES, the following updates where made:

1. Changes were made to the command-line tool:

   https://github.com/roocs/daops/blob/enable-kerchunk/daops/cli.py

2. The Docker file was updated for integration with the CWL runner:

   https://github.com/roocs/daops/blob/enable-kerchunk/Dockerfile

3. The “app-package.cwl” workflow description file was created to describe the “doaps” tool to the ADES and the WES:

   https://github.com/roocs/daops/blob/enable-kerchunk/app-package.cwl

   This includes a pre-built Docker image stored on Docker Hub for quick deployment.

4. A new capability was added to support remote data access using the Kerchunk file format which describes, and enables read access, to chunks in remote files. This unit test demonstrates the capability:

   https://github.com/roocs/daops/blob/enable-kerchunk/tests/test_operations/test_subset.py#L97-L110

5 Discussion: the NoR and usage by Research Platforms

The NoR provides a potential single point of discovery for a range of processing and data offerings from numerous providers. For offerings that are managed on public cloud resources, these are typically measurable, and the resources can be ring-fenced in a way that it is relatively straightforward to map usage to a "per-unit" cost (e.g. CPU usage or GBs of data processed/retrieved). RPs tend to manage their resources in a more *pooled* manner, where a large group of users may share access and are expected to adhere to fair usage policies. Whilst there is usually monitoring and some rate-limiting, it is rare that resources are monitored and captured at the "per-unit" level.

5.1 JASMIN-specific use of the NoR

5.1.1 Exposing the "daops" subsetter via the NoR

When considering JASMIN as an example RP, we start by looking at the "daops" subsetter tool that has been connected to the ADES and is deployed on a Slurm cluster. In this specific case, the use of EOEPCA would allow CEDA to develop and expose new processing capabilities via a Web Processing Service (WPS) interface. If deployed into production, this would enable new use cases where external users could invoke functionality on JASMIN as part of workflows run elsewhere. The real value would come if the processing was run on the main JASMIN Slurm cluster (known as LOTUS), offering significant compute and storage capabilities.

However, at present there are significant hurdles to overcome in order to achieve such a user-facing service. The main obstacle is the lack of "per-unit" request management and monitoring. If LOTUS was opened up to external usage, then it would be important to pre-estimate the *cost* of a request to inform the use/system of the impact, and we would need measurement tools in place to capture actual usage cost. Whilst there are some plugins for Slurm to attempt to measure usage, they are quite coarse and may not give an accurate picture of usage per user or per job.

In fact, providing processing services through an EOEPCA instance deployed in Kubernetes would actually enable greater ring-fencing and resource management. Therefore, it is most likely that CEDA would favour an isolated EOEPCA deployment on JASMIN in order to ensure that user and resource management was being controlled and monitored appropriately.

As for publishing services (such as the "daops" subsetter) to the NoR, the above issues of estimating the cost of processing a request and measuring the resource usage of the job would remain as pre-requisites before it would make sense to expose JASMIN capabilities via the NoR.

A further issue is that, in most cases, JASMIN resources are free at the point of use. The NoR typically expects resources to be costed and provided per-unit cost. Unlike a public cloud provider (such as AWS or Google Cloud), JASMIN does not have a system for translating usage into financial cost, nor does it have a mechanism for managing billing and transactions. Such as system would also be required. (Note that the UK Earth Observation DataHub project, with Telespazio UK as a lead contractor, aims to develop a solution that can meet this need in a cloud context which might be re-usable for RPs).

5.1.2 Exposing other JASMIN services via the NoR

We now consider other use cases in which the NoR might be an advantageous location for JASMIN-hosted services to be exposed. The following cases are considered:

1. Pre-commercial services built on top of JASMIN capabilities
2. "Mini-JASMIN-as-a-Service"
3. Services exposed by JASMIN users/projects to allow use by external web-based workflows

5.1.2.1 Use case 1: Pre-commercial services built on top of JASMIN capabilities

In use case 1, an instance of EOEPCA runs in the JASMIN cloud and a user generates new scientific outputs as an ADES WPS. The service is costed and is published on the NoR for new (potentially commercial) users to test out. Assuming

significant user uptake, the natural evolution would be to migrate the service to a fully commercial offering running on a public cloud provider with a guaranteed Quality of Service (QoS).

In this use case, the role of the RP is to enable development and testing of new services that are built by the research community and then transitioned into commercial offerings.

5.1.2.2 Use case 2: "Mini-JASMIN-as-a-Service"

In use case 2, a collection of JASMIN tools are deployed on a large EOEPCA Kubernetes cluster running on JASMIN. Each tool is very specific and operationally robust, hence the Kubernetes cluster should be running inside a ring-fenced part of the JASMIN network to protect it from cross-user impacts and downtime. Each tool would be exposed via the NoR in a basic form so that the cost (in terms of resource usage *and* financial) of a request is easy to calculate.

The QoS guarantees would remain relatively low due to existing limitations on JASMIN (e.g. ~90% uptime) and would be stated clearly in the service offering. Users would only be charged for successfully completed requests, so any loss of service during the processing of a job would not be charged. Charging would have to happen through a (third-party) costing service. For example, each user/project would be granted *n* x JASMIN Tokens which are used up by accessing service offerings. JASMIN Tokens could be gained through sponsorship (from ESA) or purchase.

The most likely audiences for use case 2 would be "mini-JASMIN" instances to support specific communities and funded by a single entity, such as:

- A specific (commercial) customer: Met Office
- A Government Department: Defra, DSIT, FCDO
- Research programme/body: World Climate Research Programme (WCRP), NCEO, NCAS

The unique selling point (USP) of such a service would be the ring-fencing of resources to reduce the impact of service degradation due to the activities of other users (i.e. those outside the specific project/organisation for which the mini-JASMIN service has been deployed).

5.1.2.3 Use case 3: Services exposed by JASMIN users/projects to allow use by external web-based workflows

In use case 3, the service offerings would be those developed by users/projects who are already working on JASMIN. They would develop functionality that they wish to make visible to clients running *outside* of JASMIN, and so the EOEPCA and ADES model provides a useful mechanism for doing so. This approach would require an instance of EOEPCA running *inside* JASMIN so that existing resources could be made available through it.

As with approaches already mentioned, this would require a method of charging and capturing resource usage that does not currently exist on JASMIN. If the resource estimation, management and charging aspects were dealt with, this use case could be a generic and inexpensive solution for users to build, test and expose their own services and to advertise, and potentially monetise them, through the NoR.

However, this use case would raise significant IT security concerns because the JASMIN managers would need to ensure that a user is not exposing protected services or data to the external world. This would require human review as part of the process of publishing each service offering. Users would also need support and tooling to correctly package their applications (into Docker and CWL) which would impose an additional cost on the JASMIN team. Overall, this use case would require greater resource separation and management within JASMIN to uphold existing access restrictions.

5.2 Use of the NoR by Research Platforms in general

Other RPs may have different business models and capabilities to JASMIN which would allow them to engage with the NoR more seamlessly. As a way to consider the possible value of the NoR in the context of a RP, the following scenarios are considered, and discussed:

- Scenario 1: Everybody uses the NoR as a point of entry
- Scenario 2: NoR access enables additional services on the RP
- Scenario 3: ESA wishes to promote the use of certain RP services

For each scenario, we consider the pros, cons, likelihood, constraints and barriers.

Table 1. Scenario 1: Everybody uses the NoR as a point of entry.

Scenario	Scenario 1: Everybody uses the NoR as a point of entry
Description	In this scenario, the NoR provides the *shop window* to data and processing capabilities for all end-users. It would essentially act as the global resource catalogue for the RP, and all other relevant services.
Pros	- All resources would be discoverable and accessible from a single location. - New users could find out about the RP and access its resources.
Cons	- Bringing new communities to the RP would increase contention over resources, and would have an impact on performance and user experience. - This would require a funding/resourcing model that could accommodate a "pay-per-use" model.
Likelihood	It seems unlikely that this scenario would happen, as most providers offer their own service and data catalogues. In the case of RPs, they will typically already have their own user base.
Constraints and barriers	Building and managing a viable "pay-per-use" model is a major hurdle in the case of JASMIN, as many of the services are provided as an open and shared resource that cannot be capped or measured at a "per-unit" level. For RPs that already operate such a model, this would not be a barrier.

Table 2. Scenario 2: NoR access enables additional services on the RP.

Scenario	Scenario 2: NoR access enables additional services on the RP
Description	In this scenario, certain "special" services are made available ONLY when accessed via the NoR. This would mean that the NoR was adding value to existing services in a way that gave it a unique purpose.
Pros	- Access to services through the NoR would augment existing capabilities. - Users would be compelled to interact with the NoR. - The NoR would bring new users to a RP.
Cons	- The service offerings of an RP would be disjointed, in that certain services would only be discoverable and accessible via a non-standard route for users (the NoR).
Likelihood	It is unlikely that an RP manager would choose to separate out its service offerings in this way.
Constraints and barriers	See: Likelihood.

Table 3. Scenario 3: ESA wishes to promote the use of certain RP services.

Scenario	Scenario 3: ESA wishes to promote the use of a certain RP service.
Description	In this scenario, ESA wishes to promote specific services provided by certain RPs. This might be done to encourage greater use of specific data sets or to promote new services that are perceived to have strategic, economic or environmental importance. Another example might be services that are aimed at supporting user communities that have limited access to computing resources, such as scientists in Least Developed Countries (LDCs).
Pros	- The promoted services will get additional visibility and extra traffic should be created towards them. - This would be a good model for supporting ESA programmes aimed at supporting knowledge transfer to LDCs.
Cons	- This approach might be considered as unfair in terms of commercial competition.

Likelihood	There may be rules that exclude ESA from promoting specific services within what is considered as a "free marketplace". For the case of supporting "public good" and international development programmes, this would be a viable approach. Many RPs will have commitments or aspirations to work in this area so sponsored access to resources could demonstrate positive outcomes for all parties.
Constraints and barriers	See: Likelihood

5.3 Current limitations and potential impacts on uptake

The discussion above outlines that the relationship between existing RPs and the NoR is complicated. Whilst there is potential to expose offerings via the NoR, the following issues are identified as significant barriers:

1. **RP Funding and Resourcing Model:** many RPs are funded at a high level and resources are provided in *pools* where a group of users share access and are expected to adhere to a policy of fair usage. If providing, or selling, services through the NoR, a "per-unit" approach to resourcing is required. At present, this would require significant additional work to enable the estimation and capture of computing and storage usage for each request submitted to the RP.
2. **Research *vs* Commercial concerns:** since RPs are funded to support research, there is a tension introduced by offering access to resources that will be used for commercial work. Some RPs will have a proportion of the overall resource that is available for commercial usage, others may be instructed to avoid any commercial usage. In cases where services are free at the point of use, access by commercial entities could degrade the service for the primary (academic) users. This concern would lead RPs to prefer ring-fenced offerings via the NoR, rather than general access to the platform.
3. **QoS and uptime:** RPs tend to provide a collection of interconnected services and typically allow users significant control in order to perform their scientific explorations. As such, RPs will incur more unexpected incidents of downtime and service disruption (often to parts of the infrastructure that have knock-on effects on other aspects). Guaranteeing a commercial level of QoS is not going to possible for many RPs. This would have to be made very explicit in the NoR offerings and appropriate disclaimers would be required for "paid-for" services.
4. **Existing RP users already have what they need:** in general, all existing users of an RP will already have access to the service offerings that they need. For that group, the provision of access through the NoR would be irrelevant. For *new* users, there is an opportunity for the NoR to expand the user base and to highlight service offerings that could benefit a wider community (see: next section).
5. **Availability of service offerings elsewhere:** for a user to sign-up for resources via the NoR, it needs to be the most efficient way to achieve their research outcomes. In many cases, processing offerings are available from other portals and platforms. Service offerings would need to have a unique component in order for the NoR route to be chosen.

An additional feature of the NoR, as provided now, might also limit uptake:

6. **Application process to NoR sponsorship:** users want to click through to a solution/product with the minimum delay. The NoR Sponsoring Wizard is a large form and presents a potential barrier to uptake.

5.4 Opportunities

The previous section lists obstacles to significant uptake of the NoR by RPs. This section focusses on some of the potential opportunities that have been identified.

5.4.1 Supporting specific offerings – such as LDC access

Given the global inequity in access to scientific and computing resources, there are many international, governmental, and institutional initiatives to support scientists and organisations in LDCs by providing remote computing and data capabilities. Joint ventures between ESA and RPs would be a natural fit in this area, where sponsorship could be

provided and the RP could make specific offerings available via the NoR within a resource limit. This could be done by ring-fencing a set of services that run on the RP without a commercial-level QoS agreement.

In the case of JASMIN, it is likely that such offerings could be connected to existing schemes run by NERC, STFC, NCAS and NCEO to support knowledge transfer and assistance. These could include:

- Processing capabilities
- Data products
- Training

All transactions would be managed via sponsorship so that the RP would not need to develop a complex charging mechanism.

5.4.2 Enabling new functionality and products to be brought to market

There are instances when research, or RP activities, shows the potential to transfer into the commercial marketplace. Whilst it is not the main focus of RPs, such activities are often supported by funders (such as government departments) because they demonstrate impact from the research community.

We have already outlined that there are policy, security and operational barriers to commercialisation, but a specific role in *exposing* new services could make use of the NoR to test market-readiness. This could be done with a limited QoS agreement and appropriate caveats on pre-production services.

If services demonstrate market-readiness, then there are two possible outcomes:

1. Migration to external service provider (i.e. away from the RP).
2. Operationalisation of the service within a ring-fenced environment on the RP. This option would depend on whether the RP could provide a mechanism for resource management and charging.

A potential USP for this approach would be that users of an RP would have a clear path for developing and testing the commercialisation of new products. This could potentially be provided with or without EOEPCA, but the latter would simplify the process of packaging up an application into a WPS.

5.5 Reproducibility

It is worth noting that reproducibility is a highly sought after feature within scientific data workflows. It is also incredibly hard to achieve[5]. The OGC Open Science Persistent Demonstrator[6] is a community initiative, backed by ESA, NASA and others, to create momentum to improving reproducible scientific workflows.

From the perspective of RPs, there are several components that change over time that make it very challenging to achieve long-term reproducibility:

- The location of resources may change – meaning that URIs become *dead links.*
- Operating systems deployed on scientistic computing systems change.
- Hardware systems change, meaning that code used to optimise certain patterns may become obsolete (e.g. when optimising Python code for GPUs).
- Software environments change and cannot always be reproduced. Tools such as “conda” have greatly improved how scientists can capture and regenerate software environments, but there are still situations where this fails to give exactly the same result.
- Datasets may change over time. Whilst it is not desirable, the size of some big scientific datasets makes it prohibitive to keep all versions. Therefore, updates can happen that mean the user will get a different result from running the same code.

[5] See: https://www.nature.com/articles/533452a for a discussion on reproducibility in science.
[6] OGC Open Science Persistent Demonstrator: https://www.ogc.org/initiatives/open-science/

- Whilst containerisation (e.g. using Docker) is a very good way of capturing a full software environment, it is not always enough. There may be security flaws found in curated Docker containers that mean they cannot be re-run safely on an RP.
- Re-running costs a lot. Typically, it is not possible to check all the above and re-run an old processing job without significant human intervention. Hence, it is not a viable option in many real-world situations.

The reproducibility challenge is an ongoing concern for the scientific community because it underpins the scientific method. The work going on through OGC and others in this field is of utmost importance to overcome this problem.

6 Conclusions and recommendations

Many of the points addressed in this report highlight the difficulties that would make RPs reluctant to publish their offerings through the NoR. Some of the main barriers reflect the way in which RPs are typically funded and how they provide access to resources. Funding from large grants and capital payments means that usage is often provided as a pool of resource rather than fine-grained *units* that are measured and accounted for at the level of the user or job. There is also an issue of policy in that RPs are typically supporting known communities and bringing in new users, particularly commercially, might have an impact on existing research users. RPs also tend to have lower QoS levels due to the large and complex nature of their offerings, providing services to non-research users might require a different QoS which would involve a costly partitioning of the platform. Existing RP users typically have access to the services they need and would not use the NoR route. In many cases, service offerings will be accessible from other points of entry, so that also limits the attraction of using the NoR.

Some potential opportunities for engaging RPs with the NoR have been identified. The first option would be to provide sponsored services through the NoR that allow to scientists from LDCs. This could integrate with existing schemes to support knowledge transfer and provide equitable access to less privileged communities. The RP could provide a ring-fenced set of service offerings without having to employ charging and monitoring tools which would be costly to develop. The second option relates to enabling RP managers and users to explore possible commercialisation of new products. An EOEPCA-driven approach could lower the bar for preparing and exposing new code as an application to be provided through the ADES, and publicised through the NoR. It is expected that full commercialisation would involve migration to a public cloud provider.

Following the investigations described in this report, we make these recommendations:

1. Engage with some RPs to investigate provision of “public good” services, such as access for scientists from LDCs:
    - This would be a good introduction and collaboration point for future work.
2. Engage with more RPs to get more understanding of the obstacles and opportunities:
    - Some RPs will have different funding models that would align with the NoR approach.
3. Support the extension of EOEPCA capabilities to manage and capture resource usage:
    - For use with Kubernetes deployment initially.
    - Explore potential tools for financial charging.
4. Simplify the NoR Sponsorship Wizard is likely to improve uptake:
    - Reduce the amount of content required.
    - Enable re-use of content when subsequent requests are made (if this is not already possible).

7 Appendix 1: overview of TOIL integration

We tested the StreamFlow, TOIL and Arvados CWL runners which all run on the head-node of the SLURM cluster. The TOIL runner met our requirements as follows:

- Able to automatically schedule SLURM jobs for a CWL workflow.
- Can automatically convert docker containers to Singularity containers, which is required to run containers without privilege in Slurm.
- Can run in server mode and provide a workflow execution service (WES) API.
- Uses a scratch space in the HPC cluster to move outputs between CWL steps.

A particular advantage of TOIL is the built-in WES server, which meant:

- Implementing our own API for this was not required.
- Implementation of communication between ADES and SLURM cluster was significantly simplified.

Figure 2 demonstrates how the ADES Kubernetes client can be replaced with a different (e.g. TOIL) WES client.

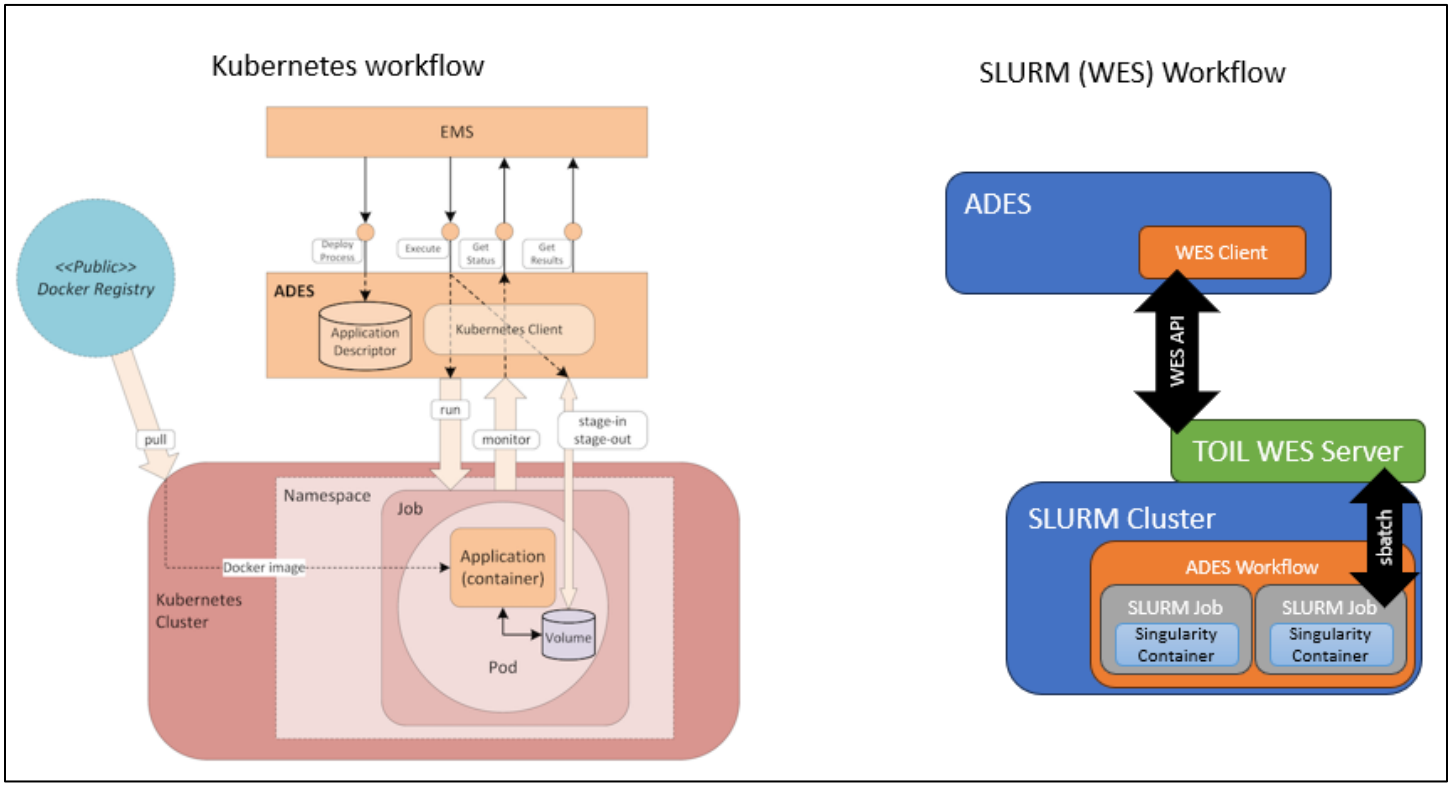

Figure 2. Diagram demonstrating Kubernetes and Slurm (WES) workflows. Left-hand panel reproduced from: https://github.com/EOEPCA/proc-ades-dev