Call: HORIZON-WIDERA-2022-ERA-01
(European Research Area)

Topic: HORIZON-WIDERA-2022-ERA-01-41

Type of Action: HORIZON-RIA

Proposal number: 101094759

Proposal acronym: SOURCE

Type of Model Grant Agreement: HORIZON Action Grant Budget-Based

Table of contents

Supporting Open, Useful, and Reproducible Computational Environments

Acronym: SOURCE

Date of Preparation: Wednesday 20th April, 2022: 14:16

Coordinator: Simula Research Laboratory

Keywords: Open Science, reproducibility, reusability, education, accessibility, Jupyter, Binder, notebooks, cloud, HPC, EOSC, FAIR data, physics, chemistry, biology, materials, geosciences

#	Participant organisation name	Short name	Country
1	Simula Research Laboratory	Simula	NO
2	Max Planck Gesellschaft	MPG	DE
3	QuantStack	QuantStack	FR
4	Ifremer	Ifremer	FR
5	University of Oslo	UiO	NO

Contents

1 Excellence

1.1 Objectives and ambition

As the values of **open science** and **reproducible science** are adopted by governments, funding agencies, research institutes, and society, there are practical challenges to implementing such practices on a global scale.

Two of the key societal goals of open science are facilitation of

- **verification** of results, improving the reliability and trustworthiness of scientific output, and
- **reuse** of research products, enabling commercial exploitation and/or derivative research.

If results are merely *available*, however, much of the value of Open Science remains unrealised. **Practical reproducibility** is required for efficient, widespread fulfilment of Open Science goals. But what does it mean to be reproducible? Without the **practical** ability to verify results, results are not any more likely to be verified. Without the **practical** ability to reuse work, results are not any more likely to be exploited in new research or commercial work.

In SOURCE, we focus on "practical reproducibility", specifically focusing on computational results, such as computer simulation, data processing, data analysis, and creation of figures and tables in publications. We differentiate "technical reproducibility", where enough information is *technically present* to reproduce the work given sufficient effort, from "practical reproducibility" where effort to reproduce the work is not a burden [18].

1.1.1 Ambition

Computational reproducibility is a major challenge facing all scientific domains, from social sciences to life sciences, physical sciences, engineering, and digital humanities. Almost every area of academia must spend some time computing, ranging from large-scale computer simulations to basic data analysis to produce a figure in a publication.

All of these face a common reproducibility challenge: **the computational environment**, or the collection of software used to produce the output. To reproduce the work, a **sufficiently similar** computational environment must be produced to re-execute the code. This may be a specific version of an application, or a large collection of software dependencies, as is common in data science fields.

In order to meet the goals or policies of open and reproducible science, **researchers need tools** to address the challenge. Ideally, those tools should

- be freely available and open source (to maximise access and longevity); and
- meet researchers where they are, as much as possible (to maximise practical adoption).

SOURCE's main goal is to **improve the global reproducibility of scientific results** with a focus on those aspects of the research process that are supported by computation and software, such as computer simulation, data processing, data analysis and creation of figures and tables in publications.

We plan to achieve this goal through

- educating researchers about good reproducible practices, and
- making it easier to perform computational research in a reproducible way through improving and developing relevant software tools.

Rather than re-inventing the wheel, we will build upon **existing tools and standards**.

Increasingly, open science and reproducibility are declared as values or requirements at research institutions, governments, and funding agencies. In order to meet these goals or requirements, researchers need **tools** that help them accomplish the tasks required for reproducible work.

It is in this context that we set our objectives to **develop, experiment with, and mainstream Binder tools as concrete solutions and best-practices to increase the reproducibility of research**:

Objective	Description	Relation to work programme
Objective 1: **Evaluate and facilitate better computational reproducibility and FAIR data**	Tools for reproducibility must be evaluated by how successfully they produce the correct environment. We will provide evaluation tools for reproducibility, and use these tools to guide improvements to the Binder tools, solving known issues where we can improve the **reproducibility of computational environments** used for science, and facilitate **FAIR data practices**.	The core of this effort is to **develop, validate, pilot, and deploy practices and practical tools for funders, publishers, and scientists**. The robustness of these tools is key to their utility and value, and thereby adoption in research communities.
Objective 2: **Enable reproducibility using common tools in a wider variety of environments**	We develop **generic tools for reproducible software environments**, but have identified several areas where the Binder tools do not yet meet user needs, such as traditional **HPC environments**, or users with **large datasets**. We shall address those gaps so that tools and strategies can be shared by a wider variety of communities, aligning effort and reducing necessary duplication.	In order to **promote uptake, greater collaboration, and increased alignment of the activities of stakeholders**, it is most efficient when knowledge and tools can be shared. When tools do not work for significant fractions of the research community, they must develop their own, often similar tools, as has often been the case e.g. for the HPC community.
Objective 3: **Demonstrate reproducibility in specific scientific applications**	We will demonstrate the utility of the Binder tools for achieving reproducible research in a variety of scientific domains, which serves both to **inform and motivate** improvements to the system, and as **illustration and reference** for others to follow.	By collaborating across domains, we **promote uptake, greater collaboration, and increased alignment of the activities of stakeholders**. Further, by publishing working examples, we contribute to an **open knowledge base of results, methodologies and interventions on the drivers and consequences of reproducibility** in these specific domains, to be used as reference or followed by others in the same domain and other domains with similar challenges.
Objective 4: **Educate researchers about reproducible practices**	Develop **best practice guidelines** for reproducible science, and disseminate this by **educating the research communities** about reproducible practices and available tools for reproducible publications and policies. **Reach out** to scientists, and the wider research communities and reproducibility stakeholders to encourage engagement with this project.	By collecting expertise and guidelines, we contribute an **open knowledge base of results, methodologies and interventions** for computational reproducibility.

Table 1.1.1: Our objectives and their relationship to the work programme

1.1.2 Binder tools for reproducible research

Binder[1] is a project providing open source tools to solve this computational environment reproducibility challenge. It has already proven useful for at least tens of thousands of researchers, educators, developers, and students, and we believe it has the potential to **facilitate practical reproducibility for millions** more, including in institutions, funding agencies, and policy makers. We aim to realise that potential by extending Binder open source tools.

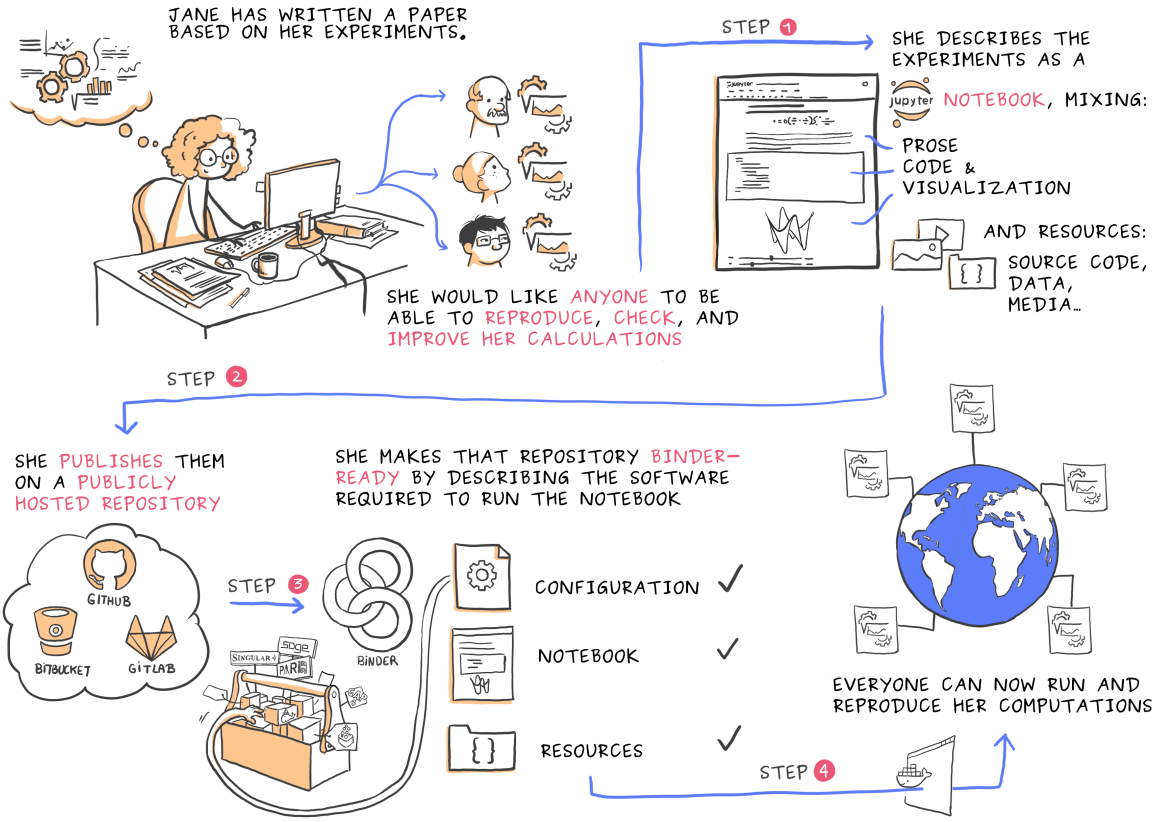


Figure 1.1.1: A typical use case for Binder-based reproducibility using Jupyter notebooks in research. Image by Juliette Belin for the OpenDreamKit project, used under CC-BY-SA.

We start with an example making use of existing **Binder tools for computational reproducibility**. Starting from this base-line, we can explain what progress and additional impact we will enable through this project.

Figure 1.1.1 depicts how a scientist can use a Jupyter notebook and the Binder software to make her research results easily reproducible and reusable. Jupyter notebooks are popular with millions of researchers and allow the creation of notebook documents, containing a mixture of text and interactively executable code, along with rich output from running that code. In short:

- Scientist Jane has created a **notebook that carries out computations** or data processing and creates a figure based on those results. For the purpose of the introduction of this workflow, we assume that Figure 1.1.2 shows this figure.
- She has made the notebook and raw data available in a **public repository** (for our example at `https://github.com/fangohr/reproducibility-repository-example`).
- As she has described **what software is needed** to execute her notebook (more details below), it is possible for all interested scientists (and anybody else, including the reviewers of this proposal) to **reproduce** her figure.
  To reproduce the figure, one needs to visit an online URL[2] (which contains a combination of the public repository and online Binder service adress) in a browser, and then wait a minute or so for a dedicated computational environment to be created and started. Then select “Run” -> “Run all cells” from the menu of the interactive Jupyter notebook that will appear in the browser.

[1] `https://jupyter.org/binder`
[2] `https://mybinder.org/v2/gh/fangohr/reproducibility-repository-example/HEAD?labpath=figure1.ipynb`

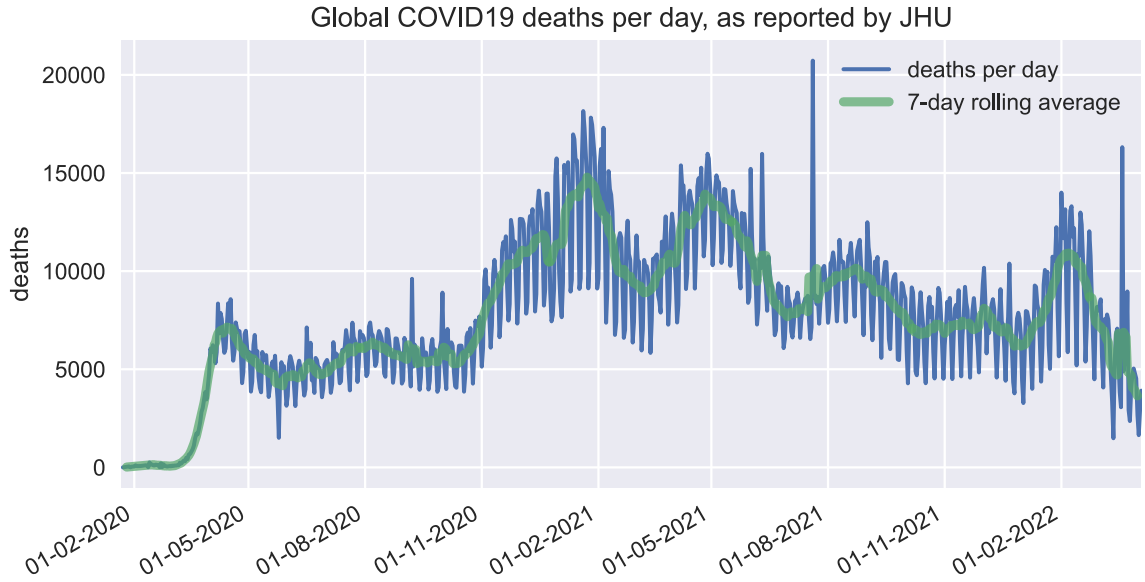


Figure 1.1.2: Figure (`figure1.pdf`) which can be reproduced in our explanatory repository example [36].

At that point, the **computational steps** that Jane has carried out **to create her results are repeated**, and the results are **reproduced**.
(We provide a more detailed description of the components and steps in this process in the Section 1.2.6.)

- Because all the computational steps are captured in the Jupyter notebook, they can be inspected and interrogated if desired. In particular, the steps can be modified and re-executed: this allows very efficient **re-use** of the results by other researchers.

The reproducibility workflow shown in Figure 1.1.1 is working today, and we provide estimates on the number of current users in section 1.2.6.4. The work proposed for this project will build on this existing technology and (i) make it **more robust and easier to use** (WP2), and (ii) extend the functionality (WP3) so that the existing tools can be **used in many more use cases** (WP4).

We note in particular that the Binder-enabled workflow for reproducibility has originally been developed to reproduce results that are created within Jupyter notebook (as shown in Figure 1.1.1). However, **the approach taken is generic, and not specific to notebooks**: it is based on `repo2docker` (Section 1.2.6.1), a tool to **build, run, and publish** Docker images from source code repositories. As part of this project, we will demonstrate that reproducible computational environments can be created using the same approach in studies that do not make use of Jupyter notebooks.

1.1.3 State of the art

To make computational research **reproducible**, we generally need to make available (i) required **data**, (ii) the required **software**, (iii) the **protocol** that explains how to process the data to obtain the result that is to be reproduced. Using services such as Zenodo, it is possible to deposit such archives with a DOI and make reference to them in publications.

In order to reproduce the results, it should be possible for anybody to take such an archive of a research output, and to carry out the two necessary steps:

- Step 1 to install the required software, and
- Step 2 to follow the protocol to reproduce the results from the archived data.

It is of particular value if Steps 1 *and* 2 can be *carried out automatically* by executing a program included in the archive. First, if the automatic execution is possible, we know that there is a complete description of the protocol included in the archive, and that no mistake is made in trying to follow the protocol. Second, the automatic execution saves time.

There are two common approaches to achieve this reproducibility: (a) to use a **workflow** tool or environment

that caters to a given use case (for example [2, 24, 35]) and encapsulates the full process for strong reproducibility guarantees, or (b) to use **standard** (software engineering) **computing tools** and conventions (git, make, python, perl, bash, …) to specify the compute environment and reproduce steps piecemeal.

The workflow tool approach (a) is robust, but requires "all-in" adoption by authors and reproducers alike, and is **difficult in practice**. It can also have associated "vendor lock-in" effects — once a tool is adopted for one piece, it must be used for all associated work. A consequence of this more tailored approach is **reduced re-use and duplication of effort**. As a workflow tool may not meet the needs of a community, that community must then build its own tool, or be left **without appropriate tools** if they lack the resources or expertise to build their own. Workflow tools also often *dictate* a significant amount of how researchers perform their work, which can inhibit adoption of the workflow tool by researchers when it does not suit their existing patterns.

The **standard computing tools** approach (b) is generic, but **not accessible** to all researchers as it requires substantial training or experience to be effective, and it is prone to errors or incompleteness when executed, leading to the "Works for Me" problem: the author may have no trouble reproducing their own work, but they have not effectively communicated the requirements such that others can do the same. The **gap between researchers' expertise and practical need** when it comes to these tools has led to the creation of whole industries of remedial skills training. The loose coupling of tools and modular choices make it **more flexible** (covering more use cases), but **more difficult** to follow robustly. Identifying which tools to use and following appropriate installation is a challenge for both authors, who must communicate requirements clearly without knowing the reader's context, and readers, who must find, understand, and follow potentially complex directions, even assuming they were complete and correct.

Researchers who use the Jupyter notebook to orchestrate their computational research can achieve this automatic reproducibility with little additional effort [4]: they use the notebook document as the protocol of their analysis (Step 2), which can be executed automatically. They can make use of the **Binder tools** (Section 1.2.6) and/or the associated mybinder.org[3] service (free and public) that has been designed by the Jupyter team to **automatically create the appropriate software environment** (Step 1) in which the notebook can be executed. But different workflows and tools are not as well served, yet.

BinderHub instances such as mybinder.org are extremely convenient, but being hosted services they do not offer the autonomy of private execution on one's own computational resources, be they a local machine or cloud or on-premises clusters.

1.1.4 Beyond the state of the art

In this project, we will focus on the **reproduction of the software environment** (Step 1) which is a prerequisite for any attempt to reproduce the actual research outputs. In particular, we want to make the creation of this computational environment **automatic**, **generic** and **robust**, especially for long-term preservation.

We will go beyond the current state of the art by bringing the following selected improvements:

- Currently, producing valid computational environments from published scientific publications and/or existing online repositories is difficult and usually requires manual handling of the process which is cumbersome and time consuming. **Improving the robustness of Binder tools** and in particular `repo2docker`'s ability to produce computational environments long after the publication and/or release — through testing and development, as well as taking additional context information into account, such as repository publication date — will **enable seamless reproduction of computational environments**;
- The existing **Binder tools** are already widely used in the Jupyter user community, but the focus has been on the reproducibility of Jupyter notebooks that may not fit everyone's needs. Highlighting and extending **Binder tools' capabilities beyond notebooks** will undoubtedly **attract new communities** of users and can facilitate **transfer of knowledge** between academia and industry;
- The current Binder tools rely on Kubernetes and deploying a Binder service requires technical skills that are beyond many institutional or companies IT support staff. As a result, most researchers rely on existing deployments that are overloaded and cannot cope with the huge demand, and they may disregard Binder as a viable solution for creating reproducible computational environments. **Being able to use `repo2docker` anywhere** e.g. from the user's personal laptop (Binder@home, **T4.2**) to the most powerful supercomputers (Binder@HPC, **T4.4**) by removing technical restrictions such as the dependence on Kubernetes, and sup-

[3] `https://mybinder.org`

porting a wider variety of community practices for reproducible environments will open new ways of using Binder tools that are more in line with the current needs of end-users;

- Another important bottleneck is the need to **access and to reuse very large and complex datasets** (sometimes with restricted access permissions) that are published and deposited on (domain-) specific long-term archives. Specific use cases (to read and process such datasets) are provided by end-users (either as part of the dataset itself or separately) but the usage of the Binder software and/or existing public Binder deployments for this use case is not yet well supported (such amount of data cannot be easily and efficiently moved to public Binder instances). *Ad hoc* or domain-specific solutions (for instance the usage of cloud optimized data formats and associated catalogs such as intake[4] or STAC[5] by the Pangeo[6] Geoscience community) have been explored by diverse communities but are technically difficult and not generic enough to be adopted by everyone. The Binder software will be extended to **facilitate data publishing** (see **T4.3**) with the plugin of external long-term archive resources and enable the **publication and reuse of large and complex datasets**.

1.1.5 Motivation — why?

We focus on the **computational reproducibility** because it is a real obstacle for **practical reproducibility**.

First, it affects the majority of all researchers: there are estimates that **over 92% of all researchers** work with research software and over 50% develop their own [16]. Where experiments drive the research, this is often data processing, analysis, and plotting. Each of those computational research cases needs a **software environment** in which the actual processing can be carried out. The software environment may consist of somewhat standard packages (for example use of a Python, R, or Julia plotting library), or it may include tailored programs that have been developed specifically for a study.

Second, software packaging and management is a technically challenging topic, and **we cannot expect 92% of all researchers to master it** – so we believe there is a clear need to support them with appropriate tooling.

The complexity arises in parts from the increasing age of archived studies, and also in the often unusual combinations of research software and libraries that need to be combined for a particular study. Other difficulties include that a reproduction typically needs to be done on a different computer, perhaps even on a different operating system. If, say, a plotting library is used, then it may change its interface or behaviour over time, so it is important to install exactly the right version of the plotting library, before a reproduction of results is attempted using it.

Third, being able to recreate the appropriate **software environment is** a **prerequisite** before any actual reproduction of results can be attempted: it would be inefficient to educate researchers what data and programs to archive, if in the future nobody (or only very few highly trained people) will be able to execute those scripts.

Fourth, there are low-hanging fruits: the work proposed here will make it possible to create computational environments **automatically for existing data archives**: the Binder philosophy is to support existing standards for software specification, and to automatically build a software environment based on those standards. Where researchers have used the existing software specification already, Binder tools will work immediately on their archived files. This means that (i) a researcher putting together a well-organised archive does not need to know about Binder tools, yet the researcher who wants to reproduce the results later can use Binder tools to automate the recreation of the software environment. This also means that (ii) improvements we propose in this work, will make some existing archives (that have been created in the past) more easily reproducible.

Finally, by popularising the standard practices, a **benefit** is achieved fully **beyond the community we reach directly**.

[4] https://intake.readthedocs.io/en/latest/index.html
[5] https://stacspec.org
[6] https://pangeo.io

1.2 Methodology

Supporting Open, Useful, and Reproducible Computational Environments (SOURCE)

The ideas behind our project title:

- The work done in SOURCE will be **Supporting** scientists in their endeavours to make their work more reproducible and reusable.
- We believe in the value of **Open** science and **Open** source software. The best reproducibility and reusability of scientific results is given through complete transparency of the steps taken in the derivation of a result. For the computational aspects this means making all simulation and/or post-processing and analysis steps open source. While this may not always be possible, we advocate such openness as the best practice for reproducible science.

  All work done, including software, training, and documentation materials, will be open source and available through an open access license. (We note that the collective development of the grant proposal you are reading is also done as open source, and can be inspected at `http://github.com/minrk/horizon-widera-2022`.)
- Measures towards better reproducibility have to be **Useful** and practical: if a proposed approach or tool burdens the scientist with additional work, or requires significant additional skills, it becomes less likely to be widely accepted.

  The philosophy we support here is that the proposed (Binder) tools for reproducibility are based on existing standards which are already adopted by many and considered best practices.
- Within the wide field of reproducibility in science, we focus this project on the improvement of the automatic generation of **Reproducible Computational Environments**.

1.2.1 Outline of concept and approach

In the following sections, we explain our concept and the technology on which this proposed project builds in more detail.

- Sections 1.2.2 Reproducibility and 1.2.3 Challenges of reproducibility contextualise the proposed work within the wide field of reproducibility.
- Section 1.2.4 Approach presents our approach to improving reproducibility.
- We provide details on some of our science use cases in 1.2.5 Science applications. These studies will provide feedback to the development of the Binder tools, validate the tools, and serve to showcase the outcomes of SOURCE.
- We provide some technical background and context for our implementation in 1.2.6 Open source ecosystem — Jupyter and Binder.
- Section 1.2.8 Community Engagement Panel presents our strategy to engage with all stakeholders in reproducibility in science. We use the forum to share requirements, constraints, and experiences from different communities, and to learn from each other. This will help us to create training materials that target different audiences and contribute to a relevant presentation of best practices.

1.2.2 Reproducibility

Before describing the focus of the work that we propose here, we want to embed this into the much wider context of **reproducibility challenges**.

We will exclude the challenges of reproducing *experimental* data. Our study starts at the point where such experimental data is available in digital form.

We will focus on the challenge of **computational reproducibility**: can we carry out the same data analysis or creation of figures and tables as they are presented in a paper, at a later stage, and get to the same results?

Such tables and figures in a publication may be computed from the analysis of some type of raw data which could originate from an experiment, another publication, a data base, post-processing of another dataset or from executing computer simulations.

1.2.3 Challenges of reproducibility

We can classify the reproducibility challenges listed above into different categories:

1. **Workflow:** Are the processing commands (and their order) **correctly recorded**? Do we know which part of the dataset the analysis is meant to be applied to? Do we know the protocol, *i.e.* are the different processing steps for that data recorded? This could be the order in which analysis scripts need to be executed – for example to compute intermediate results – which will be turned into a figure in the last step?

   We will call this sequence of steps the **workflow**. This is to a significant degree a question of the organisation and documentation of the research process. This workflow could be archived – for example – through a `README` file, or scanned pages of a hand-written laboratory notebook as a pdf file, or as a machine-executable script (or a Jupyter notebook).

   This is particularly challenging where software is used which can only be controlled via a Graphical User Interface, as it may require manual recording and description of the different clicks and steps in a laboratory logbook.

2. **Software environment:** Can we **recreate the software environment** that is required to execute these commands?

   - Where software is involved, have we recorded which version of that software is needed (or was used)? If compilation is required, do we know which compilers (and which version) and which additional dependencies are required?
   - Are there instructions on how to obtain / compile the required dependencies, and the software itself (in particular where this is about simulation-based science or more complex analysis and interpretation software tools)?

3. **Importance:** Is the researcher convinced that investing effort into making their work more reproducible is a **worthwhile investment**? This is a wide topic, touching on expectations, existing cultures, lack of metrics that acknowledge reproducibility efforts, and policies.

4. **Other:** There are other related topics, for example the challenge of archival of (large) research datasets, of making the data FAIR, and strict (important for some domains) bit-wise reproducibility.

In this proposal, we start from practices that researchers increasingly adopt, and which we argue are **good reproducibility practices**. We propose to carry out additional work to **improve the tool set enabling this practice**.

To deal with the **Workflow** challenges, we recommend automating the workflow steps as much as possible. In particular, the use of Jupyter notebooks to orchestrate the execution of commands seems effective [4]. The use of the notebook is perceived by many as an improvement of their research effectiveness because it supports "Thinking with Code and Data" [13]. As such, the practice of using notebooks (which helps improving research effectiveness) has the very positive side effect of making the work more reproducible. (However, we note that an important task of this project is to support reproducible science that does not make use of Jupyter notebooks.)

To deal with the **Software environment** challenge, we recommend following standard practices to describe software requirements. The **focus of this project is to extend the capabilities of the `repo2docker` tool** to be able to **automatically create software environments** based on such software requirement descriptions.

We can only partially address the **Importance** challenge as this needs concerted efforts from many stakeholders (such as employers of researchers, research funders, publishers). However, we will offer training that advocates the value of open science and that **teaches existing best practices** in effective computational science. The step from following such best practice to making the work reproducible is – given the Binder tools we want to develop further here – relatively small, or even possible without additional effort. Increasingly, funding policy is making this case for us, as researchers are increasingly obligated to pursue reproducibility. Our role comes in making that practice **practical** so that researchers can continue effectively fulfil these obligations.

The **Other** challenges are mostly outside the focus of this work, although our proposal will also assist in reproducible and FAIR data publishing, see for example Task **T4.3**.

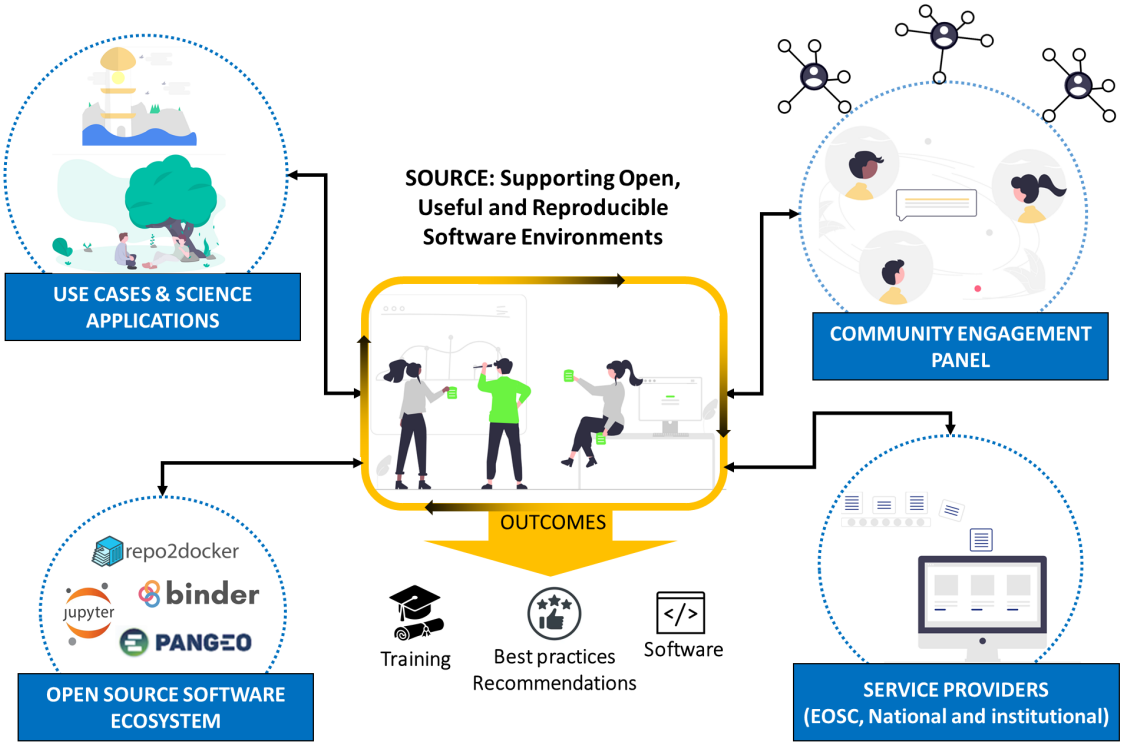


Figure 1.2.1: Overview of SOURCE approach

1.2.4 Approach

Our approach is summarised on Figure 1.2.1 and is centred around the following ideas:

1. We put the researcher at the centre of our work. In the end, researchers are responsible for making their work reproducible. It is therefore essential to find technical and social solutions that are **useful and practical**. This idea is reflected in researchers being involved in WP1 through the **Community Engagement Panel**, the requirements gathering and application of the work carried out in WP4 and the interaction with scientists through our outreach and engagement activities in WP5. All of these inputs drive the technical software work done in WP2 and WP3.
2. We also need to consider wishes and constraints from other reproducibility stakeholders. These include research councils and funders, publishers, research infrastructure providers, and educators. There are also opportunities to use the same software environment creation to support outreach and citizen science projects (see 1.2.6). A set of representatives of these domains will be gathered in the Community Engagement Panel, and connect us with the relevant communities. A list of confirmed panel members is available (see Section 1.2.8) and will be extended if the project is funded.
3. The design idea for the software components is to build on **existing standards and conventions**. This means that, in many cases, researchers have already created reproducible repositories (if they use conventions to specify software requirements), but we have not developed the tool yet to create the **software environment** for that repository automatically. This approach means our work immediately benefits researchers who already follow good reproducibility practices, and we can have impact well beyond the community members we are able to reach directly.
4. Providing **useful training** plays a key role in enabling and motivating researchers to work reproducibly. We will explain the benefits for reproducible work, and teach good practice for reproducible science such as keeping raw data, metadata, and relevant software using version control and automating analysis steps and workflows. We will also showcase tools to support the **creation (and use) of reproducible research artefacts**, including those developed through this project.
5. We will explain the possibility of using Jupyter notebooks to create **reproducible records of computational science**, while supporting non-notebook driven use cases.
6. The measures we advocate to improve reproducibility in science are designed to be embedded into the ongoing research activity to **minimise the additional burden** of reproducibility associated with the publication

of results. When executed successfully, maintaining reproducible practices should require **less effort** than non-reproducible practices.

7. We believe in an agile approach to effective software engineering to get the most **useful and fast feedback** from the use of those features in a real-world context.
8. Prototypes and testing of new features and workshop services shall be deployed on commodity cloud services for cost-effective development.
9. All our outputs will be open source and published with permissive open source licenses.

We implement our approach through 5 Work packages:

WP1 deals with the administrative (**T1.1**), technical (**T1.2**) and communication (**T1.3**) management of the project, and the organisation of the Community Engagement Panel (**T1.4**).

WP2 will improve the **robustness of reproducible environments** through technical work on `repo2docker` and `BinderHub`. In particular, we will first create a metric to be able to measure our progress (**T2.1**), improve the ability to create software environments for older repositories (**T2.2**) and improve the performance (**T2.3**). We explicitly allocate some time in **T2.4** for maintaining the tools we extend and want to build on. Maintenance is crucial to creating reliable, sustainable software, but its cost is often swept under the rug in funding applications because of the perceived pressure to focus on novelty.

WP3 will extend the feature set of **Binder tools** to broaden the impact of the project. This includes understanding more data sources and **software specification** patterns (**T3.1**), to refactor the tools to not depend on a Kubernetes installation support other container runtimes beyond Docker (**T3.2**), to export identified software dependencies, to support new use cases, and to better support use outside the Jupyter ecosystem (**T3.3**).

WP4 will test, evaluate and apply the improvements from work packages WP2 and WP3 in **real-world reproducibility use cases**, such as best practice reproducibility show cases (**T4.1**), use of `repo2docker` on decentralised hardware (**T4.2**), publishing of large, complex, or restricted access data sets (**T4.3**), and reproducibility in High Performance Computing (HPC) (**T4.4**).

WP5 disseminates outcomes, delivers training materials and activities, and **supports community engagement**. We will develop **best practice guidelines** for reproducible science (**T5.1**), organise and **deliver training** events (**T5.2**), **work closely with scientists** wishing to contribute to the project (**T5.3**).

1.2.5 Science applications

Throughout this project, we will apply the reproducibility tools to **ongoing research projects that need reproducible processes**. This is to inform the project, evaluate the tools, and provide demonstrators. We expect the demonstrators we develop to be **exploited as production services** already during the grant, and following its completion.

We will actively work with representatives of key user communities, including biophysics, climate change and biodiversity, molecular dynamics, and *ab-initio* physics calculations. Our science applications (Figure 1.2.2) follow a workflow that is similar to the one outlined in Figure 1.1.1. Below is a non-exhaustive list of science applications we will consider. These are not just exemplars, but real cases that will **drive technological developments and future exploitation** of SOURCE results and **onboard adopters** to fully demonstrate that SOURCE solves practical issues for reproducible science. In particular, we select cases that are **not yet served** by Binder tools, and require improvements, to motivate and validate our work.

- Introducing reproducibility to a **fish track analysis model** [37] based on marine physics data using the Pangeo ecosystem: Pangeo enables interactive data analysis on HPC infrastructures [26] and significant oceanographic data sets [23]. A particular challenge here is the size and complexity of marine physics data to be ingested to obtain high spatial and temporal resolution of resulting sea bass tracks required by scientists researching fish habitats. Access to parallel computing resources through Dask[7] on HPC or Cloud architectures together with access to big data stored next to the parallel computing architecture is required to make reproducibility realistic. SOURCE will add real value by **enabling access to HPC from Binder tools** (**T4.4**).

- Reproducibility of analysis pipelines and their software requirements at the example of a **biophysics application**[8]: While the compute requirements are moderate and mostly satisfied by Binder in the Cloud, the

[7] `https://dask.org`

[8] `https://gitlab.mpcdf.mpg.de/MPIBP-Hummer/glycoshield-md`

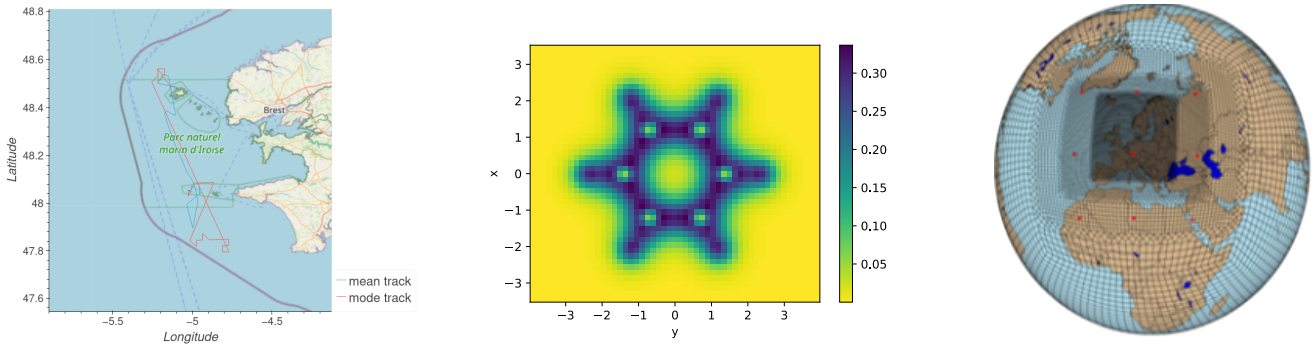


Figure 1.2.2: Selected domains from our use cases. From left to right: interactive analysis of sea bass fish track reconstructions on biologging tag data (sea bass) with low resolution marine physics data; electron density of benzene molecule computed with Octopus; NorESM arctic grid for running Earth System Modelling experiments. (Background map: ©OpenStreetMap contributors)

underlying **software stack is highly complex** – it requires the Gromacs simulation code[9] among others – and poses a particular challenge for long-term reproducibility. SOURCE will increase the re-usability of such complex pipelines, highlighting the ability of the newly developed Binder tools to re-create valid computational environments, and re-run the same pipelines, long after their publications (**T4.1**).

- Reproduciblity of results computed with the Octopus software[10] which provides time-dependent *ab-initio* computation capabilities. Octopus is a highly parallelised code. Significant compute resources (**T4.2**) or (Slurm) **job submissions to HPC computing resources** (**T4.4**) are required for reproduction of simulation results.

- FAIR@UiO[11] is the flagship project for FAIR[12] data at the University of Oslo and will leverage SOURCE work for data publishing (**T4.3**) to increase the **reuse of enormous amounts of data** that will be made available to everyone through FAIR@UiO platform. FAIR@UiO is led by the University's Center for Information Technology of the University of Oslo (USIT[13]).

- **Reproducibility of Earth System Models** with the Norwegian Earth System Modelling Consortium (NCC), represented in SOURCE by the University of Oslo. The Norwegian Earth System Model[14] [34] (NorESM) is a coupled Earth System Model and has been an important tool for Norwegian climate researchers in the study of the past, present and future climate. NorESM has also contributed to climate simulation that has been used for the sixth phase of the Coupled Model Intercomparison Project (CMIP6)[15] [34] and in research assessed in the IPCC's reports. NorESM can be run on a laptop (small configuration such as single point or single component) to the most powerful HPCs (including EuroHPC JU such as LUMI) using containers (Docker and Singularity). The current bottleneck is the need to create a container for each specific simulation (which is difficult or impossible for most researchers) and make sure a docker/singularity container can be re-created long after the simulation has been archived. Thanks to SOURCE, **researchers will be able to use `repo2docker` to automatically re-generate a docker/singularity container** leveraging the full performance of the target host hardware (CPU, communication fabric and file system) and software. This science application leverages Binder@home (**T4.2**) (model development, education, single column or very simple model configuration), Binder@HPC (**T4.4**) (operational runs at scale including on EuroHPC), and data publishing (**T4.3**) (publication of simulation results from blue-sky research). This science application is led by the University of Oslo (both the Department of Geosciences[16] and USIT) and supported by Sigma2,[17] the Norwegian National provider of e-infrastructure for the provision of computing, storage and services.

[9] https://www.gromacs.org
[10] https://octopus-code.org
[11] https://www.usit.uio.no/prosjekter/fair-uio
[12] Findable, Accessible, Interoperable, Reusable
[13] https://www.usit.uio.no
[14] https://noresm-docs.readthedocs.io/en/latest
[15] https://www.wcrp-climate.org/wgcm-cmip/wgcm-cmip6
[16] https://www.mn.uio.no/geo/english/index.html
[17] https://www.sigma2.no

1.2.6 Open source ecosystem — Jupyter and Binder

The Binder Project [18] (https://jupyter.org/binder) is a sub project of the Jupyter project, although it is not confined to be useful only for notebooks, which are the focus of much of Jupyter. **Project Jupyter** [32], which has grown increasingly popular in the scientific computing community, has become the *lingua franca* of interactive computing in both academia and industry [30]. The main goal of Project Jupyter is to provide a consistent set of tools to improve researchers' workflows from the exploratory phase of the analysis to the communication of the results [19, 13].

The key components of the Binder tools are `repo2docker` (Section 1.2.6.1) and `BinderHub` (Section 1.2.6.2). The `repo2docker` tool creates a **software environment** inside a Docker container from a **software specification** in a **repository**. `BinderHub` starts a Jupyter notebook server within this container from which the user can **execute** the notebooks or other scripts from the repository.

The *mybinder.org* service (see 1.2.6.4) is provided by the **BinderHub Federation** that collectively host a service running the Binder software under the URL https://mybinder.org. (This is the service we made use of in the example of Section 1.1.2.)

The BinderHub Federation is currently composed of four deployments of the BinderHub software

- https://gke.mybinder.org, operated by the Binder team, and hosted on Google Cloud,
- https://ovh.mybinder.org, operated and hosted by OVHcloud,
- https://gesis.mybinder.org, operated and hosted by GESIS (Leibniz Institute for the Social Sciences),
- https://turing.mybinder.org, operated and hosted by The Alan Turing Institute,

with one additional member in the process of joining at KAUST (King Abdullah University of Science and Technology).

The focus for this proposal is to improve `repo2docker`. In particular, `repo2docker` solves the software environment challenge (see Section 1.2.3) in a generic way and is independent from Jupyter notebooks.

1.2.6.1 `repo2docker` The `repo2docker` tool aims to **automate existing practices** for reproducible environment specification. It looks in **repositories**, finds *standard* **environment specifications** and produces *typical* installation commands, constructing a **software environment** in a Docker container image. `repo2docker` can fetch a remote repository and build a software environment for this repository. Currently, `repo2docker` can retrieve repositories from at least the following services and formats: GitHub, Gist, GitLab, any git repository, Zenodo, Hydroshare, Figshare, and Dataverse.

For the automatic building of reproducible computational environments, `repo2docker` understands commonly used conventions for environment specifications and community standard tools such as Docker, conda, mamba, and pip. The documentation[18] contains a full list.

`repo2docker` is successful enough to be widely adopted, but many shortcomings have been identified, especially when a repository contains an *incomplete* specification. A common issue is to produce an environment that works when published, but which may not produce a working environment at a later point in time, due to version drift and insufficiently strict specifications. Additionally, community practices not yet supported by `repo2docker` have been proposed by their respective community members, but the `repo2docker` team has not had the funding support to incorporate and maintain them.

1.2.6.2 BinderHub BinderHub is software for hosting a **web service** built on `repo2docker` and JupyterHub where individuals can share **reproducible environments for immediate and free interaction** by readers in their browser.

The BinderHub software exposes `repo2docker` and Jupyter as a service, allowing **one-click reproduction of published environments for interactive exploration**. BinderHub is shown to work well, but has limited application due to technical limitations, such as its reliance on the Kubernetes deployment platform, or challenges with combining the convenience of BinderHub's anonymous-by-default model with authenticated and/or performant access to large data or compute resources.

While the `repo2docker` terminal utility *can* technically be used anywhere, the technical expertise required is dramatically greater than a single click on a website such as mybinder.org.

[18] https://repo2docker.readthedocs.io/en/latest/config_files.html

1.2.6.3 Binder example The most common reproducibility use case – with the current state of the **Binder tools** – is the one we introduced in Section 1.1.2 using the mybinder.org service. We will use this to describe the role of the individual components of the Binder tools:

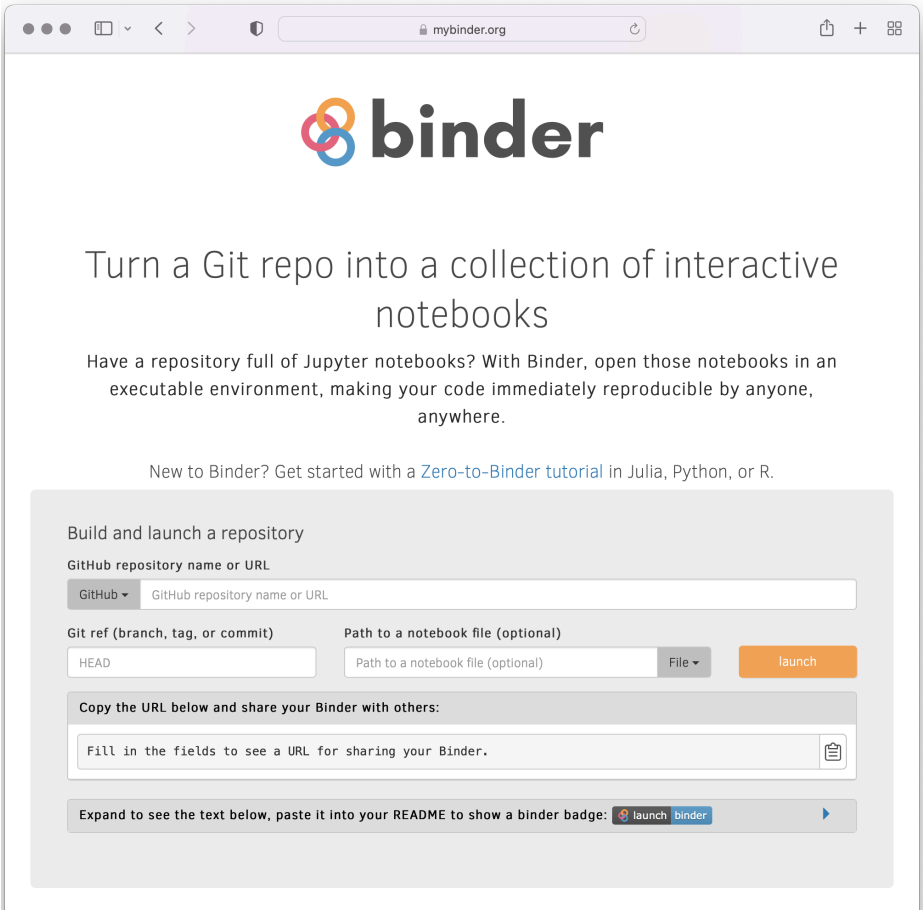


Figure 1.2.3: Home page of the mybinder.org service.

- The `BinderHub` service is given a URL that encodes the location of the data repository[19] (see Figure 1.2.3)
- `BinderHub` will use `repo2docker` to create a **software environment** (Docker image) in which the notebooks or scripts from the repository can be executed.
- `repo2docker` searches the repository for **software specifications**.
- `repo2docker` constructs the **software environment** from the **software specification(s)** (a Docker image).
- `BinderHub` asks JupyterHub to start a **notebook session** in this **software environment**.
- `BinderHub` forwards the user who requested this environment to the URL at which the repository (or a particular notebook) can be **explored interactively** from within the Jupyter application.

1.2.6.4 The mybinder.org service The mybinder.org service is run by the BinderHub Federation[20] of organisations. Collectively, they host a service running the BinderHub software which can be reached from `https://mybinder.org`. The service is **actively used** with approximately 200,000 sessions being requested and delivered by the mybinder.org service every week in 2021. The number of sessions is growing from approximately 10,000 per day in November 2018 (beginning of the available records) to about **30,000 sessions per day** in 2022. We have identified **60,000 unique repositories published** in the last few years which have used the mybinder.org service. The data is available [25].

Examples of reproducible repositories that make use of the mybinder.org service include reproducible research repositories [3, 4], interactive textbooks [7, 38] and citizen science and outreach activities [14, 9].

[19] For example the `https://mybinder.org/v2/gh/fangohr/reproducibility-repository-example/HEAD?labpath=figure1.ipynb` refers to the GitHub repository "reproducibility-repository-example" of the github user "fangohr", asking to open the "figure1.ipynb" file.

[20] `https://mybinder.readthedocs.io/en/latest/about/federation.html`

This SOURCE project will not provide or operate a public BinderHub service such as the global mybinder.org instance. The resulting improvements to Binder tools, however, will be immediately available for **exploitation by all operators of BinderHubs**, including mybinder.org.

1.2.7 Technical Readiness Level (TRL)

Binder tools can be used in many ways, and their TRL depends on the context. The Binder software and service demonstration at mybinder.org is TRL 6, where scope *excludes* long-term robustness. The Binder software for deploying services with authenticated access to data is TRL 3. We will bring it to TRL 6. The Binder tool repo2docker when targeting robust reproducible environments is TRL 4. We will bring it to TRL 6.

1.2.8 Community Engagement Panel

The Community Engagement Panel (CEP) is a forum we created for this project to bring together representatives of **current and potential user communities** of Binder tools. Through the community engagement panel, we aim to maximise the interaction between existing and new users. This will help shape the software features so that we can achieve the highest possible impact for practical reproducibility in science. Stakeholders for the topic of reproducibility that should be represented in the community engagement panel include **researchers, research infrastructure providers, publishers, research councils, librarians, and educators**.

We have already secured agreement from the following to be part of the panel, and will extend this if funded:

- Suzanne Dumouchel, Head of European Cooperation at TGIR Huma-Num CNRS unit, a large infrastructure for digital humanities and member of the EOSC Association Board of Directors. She is the partnerships coordinator of OPERAS Research Infrastructure, devoted to scholarly communication in Social Sciences and Humanities and is a member of the DARIAH ERIC Coordination Office, dedicated to Digital Arts and Humanities. She is also the scientific coordinator of TRIPLE, H2020 project (INFRAEOSC2). Strongly committed to the open science movement and to the promotion of research in Social Sciences and Humanities (SSH), she is particularly active in the field of research infrastructures.

- Andy Götz, Software Group Leader at European Radiation Synchrotron Facility (ESRF), coordinator of the EOSC project PaNOSC for making data from photon and neutron facilities FAIR, and chairman of the IT working Group of the "League of European Accelerator-based Photon Sources" (LEAPS). The LEAPS facilities wish to enable their users to create reproducible publications based on large datasets captured at the light sources.

- Paula Andrea Martinez, Project Coordinator — Software Program, Australian Research Data Commons (ARDC). She is also the co-chair of the FAIR4RS RDA Working Group and Community Manager at Research Software Alliance (ReSA), and actively contributing to increase the visibility of research software.

- Aleksandra Nenadic, Training Lead of the Software Sustainability Institute, based at the University of Manchester (UK). She is also an active member and promoter of the Carpentries community and involved as an instructor, instructor trainer, mentor, workshop organiser and regional coordinator for the UK, driving and supporting new material creation using the Carpentries collaborative and pedagogical lesson development principles.

- Gergely Sipos, head of services, solutions and support department at the EGI Foundation. He is representing EGI, "Advanced Computing for EOSC" (EGI-ACE) and the EOSC Compute Platform, which are working on a large-scale deployment of BinderHub as part of their services for researchers in Europe and beyond.

- Violaine Louvet, head of GRICAD (Grenoble Alpe Research - Scientific Computing and Data Infrastructure), supported by CNRS, Grenoble Alpes University and INRIA. GRICAD is a Tier 2 infrastructure and provides data and computing resources to all the science communities in Grenoble. In particular, GRICAD provides HPC, HTC, cloud and storage resources for all the disciplinary fields, from computer sciences to human sciences and health. GRICAD also offers a JupyterHub and a BinderHub platform. She is also very involved in helping the scientific communities and in training activities.

- Rollin Thomas, Big Data Architect and HPC expert at the National Energy Research Scientific Computing Center at Lawrence Berkeley National Laboratory (US). He represents the HPC community, and focuses on interactivity, real-time, and reproducibility in supercomputing for science.
- Andreas Zeller, Professor of Software Engineering at Saarland University. He uses Notebooks to provide open-source text books to his students and the world-wide community of readers. He will represent Binder users in academia, who use it to deliver zero-install computational environments for educational purposes.

1.2.9 Gender aspects

SOURCE is committed to implement communication and outreach activities for promoting the role of women and underrepresented groups in science and STEM: i) present showcases to demonstrate the results of the project through the eyes of underrepresented Research Software Engineers and researchers; ii) systematically offer hybrid or online training opportunities to encompass the lack of mobility of some potential attendees; iii) monitor gender participation to our training, workshops, and hackathons and track progress. Members of the consortium are involved in a number of programmes and activities who aim at upskilling women and diversity and inclusion, for instance The Carpentries and CodeRefinery training programmes and mentoring programs such as Outreachy or Open Life Science.

Beyond outreach and communication, the Jupyter Project is committed to fostering an inclusive and welcoming environment for everyone, and we will benefit from that experience for example through a code of conduct for workshops and online events.

1.2.10 National and international research or innovation activities

The SOURCE project, as part of the overall Jupyter environment, will intrinsically build on the achievements of many national and European and international research and innovative developments. The Jupyter Project itself and its software ecosystem provides a substantial international research and innovation activity which will both feed into this project and benefit from it directly. Initial points of contact, beyond common ambitions to foster open science, are provided through the participants who have been long-time members of Project Jupyter (**Simula**, **QuantStack**).

The science applications build on the outcomes of national, Nordic, European and/or international research and innovation projects and initiatives. For example, both Ifremer and the University of Oslo are active members of the international Pangeo[21] Big data Geosciences Community. Ifremer, through its co-ownership of Mercator Ocean International,[22] participates in the operation of the Copernicus Marine Environment Monitoring Service (CMEMS), a part of Copernicus, the European Union's Earth Observation Programme and is also actively involved in the development of a Digital Twin of the Ocean that will be fully compatible with the Destination Earth (DestinE[23]) architecture and supports the European's Digital Strategy and Green Deal Package. See also Section 1.2.5 and Section 3.2 for additional research activities.

1.2.11 Interdisciplinarity

SOURCE brings together research software engineers with researchers from multiple science domains to develop tools and methods that work across scientific domains. An interdisciplinary approach is key to our success (see Section 3.2.2). In particular, we bring together core software engineers with expertise in developing and maintaining research software, with researchers delivering cutting-edge research, educators, and research infrastructure providers. Each represents different stakeholders and expertise in the full lifecycle of reproducible research — from tool development, to user needs, to service provider, ensuring we have the best chance to solve real-world problems with a lasting impact for better reproducibility in science.

[21] https://pangeo.io
[22] https://www.mercator-ocean.eu
[23] https://digital-strategy.ec.europa.eu/en/policies/destination-earth

1.2.12 Open science practices and implementation

SOURCE will make use of open source licensed software, packages and libraries, following the OSI recommendations.[24] All codes in this project will be open source and collaboratively developed using GitHub, following best software engineering practice such as version control, tests, and continuous integration. Training materials will be collaboratively developed through the Carpentries Incubator[25] using the Carpentries Curriculum Development Handbook:[26] all Carpentries lessons are licensed under the Creative Commons Attribution version 4.0 (CC-BY)[27] and any related software under the MIT license.[28] All code will be developed fully in the open, actively soliciting community engagement and transparency.

Use cases and showcases will only make use of data that are openly available.

All publications and/or any research data produced within SOURCE will be published Open Access.

1.2.13 Research data management and management of other outputs

The Data Management Plan (DMP) will be prepared and regularly updated within WP1.

Except for the software and usage data described below, SOURCE activities will not generate or collect data. While we have many demonstrators that interact with data, they do not generate or collect that data themselves, but rather provide analytical mechanisms or access to data governed by existing data management plans and data policies of project partners at each site, as well as publicly accessible open data.

All data generated, collected, processed and stored will be made available following the relevant standards and regulations. Processing of personal data in relation to training, hackathons and/or workshop events will comply with GDPR regulations e.g. data anonymisation and minimisation before sharing. Procedures to monitor the real-time effectiveness of our dissemination and communication strategies will also be GDPR compliant. We have no plans to collect or produce personally identifiable information (PII) during the project.

Software Most research data from SOURCE will take the form of code. All code produced by SOURCE will be developed in the public using open source licenses (see Section 1.2.12). Milestone achievements of the software may be archived in separate repositories such as Zenodo, e.g. to coincide with publications.

Service usage data Any data collected through the operation of public services (e.g. popularity data for public open science repositories) will be fully anonymised to the satisfaction of relevant best privacy practices and regulations, such as GDPR, and made publicly available in the standard JSON Lines format, as is done already for mybinder.org [25]. This is very small data and easily archived on free hosting services such as GitHub/Zenodo, and will be made available under the Creative-Commons Universal Public Domain Dedication (CC0). There is no cost to the project associated with archiving this data long-term.

[24] https://opensource.org/licenses
[25] https://carpentries-incubator.org
[26] https://cdh.carpentries.org
[27] https://creativecommons.org/licenses/by/4.0/legalcode
[28] https://opensource.org/licenses/MIT

2 Impact

2.1 Pathway toward impact

Almost all researchers perform some kind of computation in the course of their work, from production of simple charts and figures to large-scale simulations. Researchers today are faced with a great variety of useful and increasingly open tools to perform their research tasks. However, this variety presents its own challenge. Because we are building and extending **generic tools for computational reproducibility**, we help researchers of all kinds perform their work reproducibly without dictating how they do the research part of their work. By focusing on (**reproducible computational environments**), and offering a generic and interoperable solution, we *are* solving a big piece of the reproducibility problem and act as a bridge between any solutions to other parts of the reproducibility problem, for instance those classically addressed by workflow management systems. "Do one thing and do it well" is key to success and will accelerate progress by allowing others to concentrate on different challenges in the reproducibility problem.

Helping researchers navigate this diverse landscape of research software with both education and practical tools will enable a **large fraction of all researchers to perform research in a more efficient, more productive, more reproducible, and more *useful* way**.

2.1.1 Contributions towards outcomes and impact

The expected outputs and impact of SOURCE with respect to the work program is detailed in Tables 2.1.1 and 2.1.2, respectively.

Expected outputs from call	Expected outputs from the SOURCE project
Structured understanding of the underlying drivers, of concrete and effective interventions – funding, community-based, technical and policy – to increase reproducibility of the results of R&I; and of their benefits;	Developing a structured understanding of the technological challenges of practical reproducibility of software environments is a prerequisite for our work on the community-effort-based Binder tools (WP2, WP3, WP4). We know that the technologies such as the Jupyter notebook and their reproducibility with Binder find some acceptance in the scientist community (see Section 1.2.6). One of the outcomes of this project is an improved structured understanding of the underlying drivers for the wider research community to adopt or not adopt these technologies as a concrete and effective action to improve the reproducibility of their results (WP5). We expect an interplay of technical, social, community-specific and policy drivers affecting this.
Effective solutions, policy-, technical- and practice-based, to increase the reproducibility of R&I results in funding programmes, in communities and in the dissemination of scientific results;	The Binder tools are effective and practice-based technical solutions to increase the reproducibility of R&I results in funding programmes and in scientific research more widely [4]. The combination of Jupyter notebooks and Binder are also effective solutions for the dissemination through workshops [5] and education activities [38] involving computation as participants can use complex software environments through their browser, and do not need to install any additional software locally.
Greater collaboration, alignment of practices and joint action by stakeholders to increase reproducibility, including but not limited to training, specialised careers and guidelines for best practice.	SOURCE is developing tools that enable **automatic computational reproducibility** across many domains by aligning to the existing software specification practices. Having one tool that respects existing practices means it can be adopted across many domains, thus **facilitating greater collaboration**. We will develop **best practice guidelines** for reproducibility and open science, and disseminate them through training activities and materials. The staff hired for the project will be highly specialised research software engineers. We expect that voluntary contributors attracted by the project will also receive additional training for such specialised careers.

Table 2.1.1: Relating SOURCE outputs to the outputs expected by the call

Expected impacts from call	Expected impacts from the SOURCE project
Increased proportion of reproducible results from publicly funded R&I	The SOURCE project will improve the ease of use tools for reproducibility, thereby **increasing the number of reproducible research outputs**. The project will also **educate researchers** to provide the motivation, required technical skills, and an understanding of best practice to achieve reproducible science outcomes.
Increased re-use of scientific results by research and innovation	**Reproducibility enables reusability**, in particular where the reproduction of the results can be carried out automatically (for example using the tools developed and improved in the SOURCE project). From anecdotal evidence, it is known that a new PhD student may need many months (sometimes exceeding 12 months) to reproduce a published study which is to form the basis of their new research task. An immediately reproducible research output made available together with the publication can reduce this time immensely: a binder-enabled reproducible repository may be able to recompute the results within an hour or a much shorter time because the process is automatic. The availability of an immediately reproducible research output – as fostered by SOURCE – plays a very important role in modern research, where generally advances are built based on prior results.
Greater quality of the scientific production.	The availability of practical reproducibility as advanced by SOURCE has the potential to improve the quality of scientific research through new opportunities for collaboration and interdisciplinary research: We have already seen a good example of the Jupyter ecosystem facilitating an interdisciplinary collaboration: the Nobel Prize-winning LIGO scientific collaboration shared notebooks detailing the data processing steps which led to the discovery of gravitational waves, using the Binder service to allow anyone to re-compute the published plots [14]. Scientists with no background in gravitational waves studied these notebooks and improved the signal processing. In this proposal, we want to provide this ability to a wider audience through improved tools and documentation, including for disciplines which rely on processing much larger volumes of data.

Table 2.1.2: Relating SOURCE impacts to the impacts expected by the call

2.1.2 Measuring impact

As we are building tools for open and reproducible science, the best measure of our impact is in the adoption and use of these tools and services based on them. This can be observed qualitatively (anecdotal feedback and case studies) and quantitatively (counting workshop attendees, for example). We measure progress toward our objectives (Table 1.1.1) via the following Key Performance Indicators (KPIs):

KPI 1: Fraction of published repositories for which Binder tools can build the computational environment – measuring both improvements in Binder tools and impact from our training efforts. We will implement the required methodology in **T2.1**. (Objective 1)

KPI 2: Number of researchers who benefit from using SOURCE results in their work. Due to the open source nature of our outcomes, it is difficult to track this number accurately. To solicit this information, we can request feedback from users on forums, mailing lists and social media, and count the users we know directly working with them through WP4 and WP5 for example. We expect to under count beneficiaries for this metric, but believe the KPI is still useful to measure our direct impact for better reproducibility in science. (Objective 2)

KPI 3: Number of demonstrators enabled that make results from scientific research reproducible. (Objective 3)

KPI 4: Metrics to capture the number of people that have made use of the training and education activities of SOURCE, such as workshop attendees, viewers of training videos, access to online documentation of best practice. (Objective 4)

2.1.3 Target groups and scale of impact

We have already discussed in Section 1.1.5 that **computational reproducibility affects the majority of researchers**, including those who carry out work that is not predominantly computational.

The impact of this project will be realised through (i) the training we develop and disseminate, and (ii) the improved Binder tools. The improved tools will directly impact the community of researchers using Jupyter notebooks. However, the **improved functionality and applicability of the Binder tools outside Project Jupyter** will benefit all researchers who need computational reproducibility, and may be – and in the long run – more important.

We can give some indication of the size and activity of the Jupyter notebook community and use of the existing Binder tools: Jupyter notebooks are used to support research in numerous communities, including Journalists and practitioners of data-driven journalism at the LA Times, BuzzFeed News, Columbia Journalism School [21] [15] [17]; Data scientists in academia, industry and services [30]; Research institutions such as CERN, EuXFEL, JRC, and many more, operating institution-wide Jupyter deployment; Universities using Jupyter as a teaching platform; Large cloud providers building commercial products on the top of Jupyter (Google DataLab and Colaboratory, AWS Sagemaker, OVH AI Notebook); within the European Open Science Cloud, Jupyter is used in many EOSC projects, and a JupyterHub service is provided by the EGI foundation. This impact was recognised by the *ACM Software System Award* that was awarded to the Jupyter team to honour *"developing a software system that had a lasting influence"* in 2017. (Prior recipients include *Unix*, *TCP/IP*, and the *World Wide Web* [1].)

There are **at least 8 million notebooks** deposited on GitHub [28], and the size of the **notebook user base was estimated** to be of the order of **millions in 2015** [29]. We know that the public Binder service **mybinder.org was used to create over 10,000,000 computational environments** from at least 60,000 unique repositories in 2021 (Section 1.2.6.4).

2.1.4 Potential barriers

A central barrier towards impact would be if the research community would not accept or embrace the tools and practical guidance developed here. To minimise this risk, we'll stay in close touch with all our stakeholders (WP1, WP4, WP5), and use the experience of the team members who are active researchers themselves.

A key strategy in mitigating barriers to adoption is the approach of **automating existing practices**. All of the **practices promoted** and developed by SOURCE **are beneficial** to producers and consumers of reproducible research alike, **independent of the specific Binder tools** in which we choose to implement the automation. This avoids lock-in and ensures positive impact even among researchers who choose not to use any of the specific tools we develop.

2.2 Measures to maximise impact – dissemination, exploitation and communication

SOURCE is contributing to tools for open and reproducible science. It is essential that we disseminate our work in order to **reach and support user communities** (such as researchers and research infrastructure providers), enable them to best exploit our software and services, and achieve impact. This section outlines how the project will establish and organise the dissemination, communication, and exploitation actions to promote the project and the adoption of its outcomes beyond the project's lifetime.

2.2.1 Dissemination of results

SOURCE endeavours to make open and reproducible science practices both more **understandable and actionable** to practitioners through the development and use of open and freely available tools. Most of SOURCE software will be in the form of contributions to existing projects, which will be governed by the licenses of those projects. All Jupyter and Binder software is released under the permissive BSD license, which specifically allows commercial exploitation, as has proven successful in enabling collaborations with industrial partners such as Google, Microsoft, IBM, OVHCloud, and more. Any other developments will be made publicly and freely available under open source licenses, and hosted on public code hosting sites such as GitHub. This means that all **SOURCE software will be available and accessible to all** who find it, at no cost to SOURCE, enabling long-term access beyond the funding of SOURCE. Similarly, non-code products such as dissemination works (workshop materials, etc.) will be made freely available under open Creative Commons licenses.

All the partners will be involved in the dissemination of SOURCE results (see draft plan of the dissemination, communication and exploitation plan in the text box in Section 2.2.4) and WP4 and WP5 will play a central role.

As a result, the primary dissemination effort is to:

1. make sure that **prospective users are aware of the work** through show cases, science demonstrators, demos, best practice documentation, project communication, and

2. **enable them to use and exploit the tools** through learning resources, training, and services.

Our focus for dissemination will be on **T5.2** operating workshops, training various communities in the availability, purpose, development, and use of SOURCE software and services. We will make a particular effort to use these workshops as an opportunity to **support diversity and inclusion in the open science community**, by running (online, hybrid, or in-person) workshops for under-served and under-represented groups in the academic and open source communities. **Free and online training** will be streamed (e.g. Twitch), made available to the wider community for instance on YouTube, and archived on Zenodo for long-term availability and findability to the wider community who may not be able to attend workshops.

These resources will be hosted on free, public hosting services, such as GitHub Pages, ReadTheDocs, or YouTube channels, and as much as possible co-developed and co-hosted with existing and well-established organisations (The Carpentries,[29] CodeRefinery,[30] Galaxy Training Network,[31] Pangeo,[32] etc.).

We will also disseminate our results through **publications and conferences**. All publications funded by SOURCE will be **open access**, and sites expecting publications have budgeted funds for paying open access fees. We will identify and attend appropriate conferences for disseminating our work, including running tutorials at conferences in historically interested communities such as PyData and SciPy. Also, we will identify and attend conferences from complementary communities such as ROpenSci, Mozilla Science, and Julia, as well as domain specific conferences to maximise the impact of SOURCE and to broaden its audience outside the traditionally included communities.

The **operation of prototype services** in WP4 is also a dissemination activity, as services like Binder not only enable open and reproducible science by facilitating interactive publications, they also enable **interactive demonstration of tools and functionality** developed in SOURCE.

SOURCE will also actively seek **collaboration** (through the Community Engagement Panel and SOURCE partners) with existing **EOSC projects** (many already use/deploy Jupyter notebook services) to inform and support them in exploiting SOURCE developments and adapting their service offerings.

2.2.2 Exploitation

Our primary outputs are in the form of software tools, prototype services, and information resources, all of which will be freely available to all under appropriate permissive open licenses (such as BSD/MIT). This means that exploitation generally has the form of:

1. Use of Binder tools for **producing reproducible environments** in research, as enabled by WP2

2. Use of Binder tools for **evaluating reproducible environments** in research or publication **policy**, as enabled by WP2

3. Use of Binder tools in the **construction and operation of services**, as enabled by WP3

4. Deploying **new services** derived from our science demonstrators in WP4

5. **Application of effective practices for reproducibility**, developed and disseminated in WP5

6. **Commercial exploitation** via product development and consulting.

[29] https://carpentries.org
[30] https://coderefinery.org
[31] https://training.galaxyproject.org
[32] http://gallery.pangeo.io

A key aspect of our exploitation strategy is to focus our work on existing, active projects in Binder and the wider Jupyter ecosystem. This means that our contributions are of **immediate practical benefit to between thousands and millions of users** worldwide (Section 2.1.3).

SOURCE contributions to Binder tools will be immediately (*i.e. already during the duration of the project*) **exploited via the public mybinder.org service** and its thousands of daily users (Section 1.2.6.4), as well as **many BinderHub instances worldwide**.

The science demonstrators in WP4 are themselves exploitations of the outputs in WP2 and WP3, building services not possible before the project. **Continued operation and adoption of** these **reproducibility services** by project partners and the researchers they serve beyond the duration of the project is anticipated for a majority of the science applications, and will indicate successful exploitation of the project.

The **QuantStack** SME provides **commercial support and development services around the open-source** scientific computing ecosystem. The team has specialised in the Jupyter ecosystem and package management solutions, which are both at the core of this proposal. The consolidation of Binder's technical foundations will help QuantStack provide robust solutions to its clients.

Industrial use cases met by QuantStack also show that the reproducibility issues addressed in this proposal go beyond the scientific use case:

- For example, **financial institutions** using numerical libraries to price and hedge derivatives must be able to reproduce results obtained with past versions of their software, especially in cases of mispricing or mishedging. Rolling back to a state of the software environment as of several years earlier may prove extremely difficult. Adopting the approach of Binder to favour reproducibility has proven to be a viable way to address this issue.
- Similarly, in the field of **industrial robotics**, QuantStack has developed a conda-based distribution of the ROS open-source ecosystem called RoboStack, and a tool akin to repo2docker to produce container images providing all required packages for a ROS node, that will directly benefit from this work.

This shows that techniques and software tools that arise from the computational research communities to address reproducibility will help address the same issue for a much broader audience, including commercial applications.

2.2.3 Communication activities

The main goals of the communication activities is making sure that researchers and potential users are aware of the outputs, tools and services developed by SOURCE on an ongoing basis, and to demonstrate to the public the clear benefits of the work they have funded.

In order to maximise this impact, it is vital to address the audience as one project and ensure the immediate recognition of information stemming from it. Together with all partners involved, SOURCE will therefore build a **strong project identity** (see draft of the communication, dissemination, and exploitation plan in Section 2.2.4) to strengthen the project identity and to deliver clear messages to our audiences.

We will operate a **website** (**T1.3**) for collecting and sharing information about SOURCE and its progress. It will provide a centralised way to access the various publicly available deliverables, publications and articles related to the project. The site will be regularly updated over the lifetime of the project with the project publications and public materials, such as flyers, posters and public deliverables, organised workshops, available services, news, etc. Site analytics will be associated with the project website, in order to provide useful insight on how to improve its impact. In addition, the project intends to develop its presence on **the social and content networks**. The channels will be used for interaction with the professional community as well as the general public (differentiation on the content per channel based on the target group wishing to address). As part of the project's communication plan, SOURCE will develop a social media strategy in order to increase outreach and social impact, which can be summarised as follows: (a) identifying target audience and key stakeholders, (b) updating social media content and sparking discussion in social media/tweeting, (c) measuring social impact and reassessing social media strategy as required.

2.2.4 Communication, dissemination, and exploitation plan

We will continually refine our Communication, Dissemination, and Exploitation plans, starting with the draft below:

SOURCE DRAFT COMMUNICATION, DISSEMINATION, and EXPLOITATION

- Logo creation, standard document templates for deliverables, reports, letters, presentations, project posters/leaflets, etc. and creation and/or use of relevant social media accounts such as Linkedin, Twitter, YouTube channel (80k Twitter followers for @ProjectJupyter, 34k for @IPythonDev, and 132k for the PyData Youtube channel);
- Annual plan for publication in scientific and technical peer-reviewed journals and conference proceedings;
- Creation of SOURCE website:
  - links to SOURCE repository of results (videos, training material, use cases, demonstrators, software and associated documentation such as repo2docker, JupyterJub and Binder);
  - links to social media accounts, e.g. on Facebook, Instagram, LinkedIn and Twitter and public communication channel (Zulip);
  - live "Infoboard" to highlight news and outcomes, event calendar, public resources, specific social media posts; This infoBoard will be relayed by the News (and Newsletter) of partnered projects (Jupyter, Pangeo, Ifremer, NeIC, Sigma2, etc.).
  - links to the EOSC catalogues of services and national/institutional services using and supporting SOURCE tools and outcomes;
- Use the Community Engagement Panel to review the target audience for dissemination (workshops, training, hackathon, etc.), create a live list of communication partners and associated channels and tools;
- Coordinate (annually) with other initiatives (The Carpentries, CodeRefinery, Jupyter, Pangeo) for the collaborative development of training material and the delivery of demos, training/workshops/hackathons;
- Refine measurable goals (based on inputs/feedback from the Community Engagement Panel), quantitative KPIs for dissemination and monitoring procedures;
- Develop and maintain a live collaboration plan for links and interactions with other projects (EC-funded, Nordic and/or national projects), research institutions and industries to find opportunities to show-case SOURCE results;
- Develop a consolidated exploitation plan to ensure the long term and sustainable exploitation of the project results beyond its lifetime, in particular the deployment of EOSC services.

2.3 Summary

SPECIFIC NEEDS

Computational reproducibility is a widely recognised challenge. **Most researchers need computational reproducibility** to achieve reproducible science outcomes, such as publications. There are many tools that solve *part* of the problem, or aim to solve the whole problem while requiring wholesale adoption of a specific tool. This may not be practical or desirable across a variety of domains or communities.
Additionally, **solving software environments is hard**, and researchers need easy-to-use **tools**.

EXPECTED RESULTS

Improved software tools for reproducing computational environments: by improving the Binder tools for reproducible environments, it will be easy for researchers to produce and share results openly and reproducibly.
Improved understanding of good practices for reproducibility: study results, documentation, and workshops will aid researchers and policy makers in understanding the most appropriate practices to adopt in their pursuit of reproducible research.

D & E & C MEASURES

Exploitation: We contribute to the **Binder tools**, already widely used by SOURCE members and hundreds of thousands world-wide. All existing applications, from the mybinder.org service to individual users, immediately and directly benefit from our improvements to the tools. Our science applications specifically exploit new features developed by the project.
The QuantStack SME will **commercially exploit** the improved Binder tools in areas ranging from industrial robotics to quantitative finance.

D & E & C MEASURES

Dissemination: Results, insights, tools, and guidelines will be disseminated to potential users and stakeholders through conventional academic channels, in-person and remote workshops, websites, video tutorials and social media.

D & E & C MEASURES

Communication: The media and general public will be informed about the project through news releases, videos, a website, social media, further media engagement, flyers and interviews, with a consistent visual identity.

TARGET GROUPS

Computational science practitioners: researchers with an interest in reproducibility of their own work.
Research infrastructure providers: organisations supporting researchers through experimental and computational services.
Research institutions: institutions facilitating or enforcing the reproducibility of their researchers.
Policy makers and publishers: Funders and institutions requiring their subjects to follow reproducible practices.

OUTCOMES

Adoption of Binder tools for reproducibility: practitioners will have access to Binder tools for reproducibility, having improved their usability and robustness.
Resources for reproducible practices: practitioners will have access to resources to learn about effective reproducibility practices.
Facilitating practical policies for reproducible publications: policies will have access to tools for validating reproducible practices that they require.

IMPACTS

Improved reproducibility of scientific results: Improving the *ease of use* tools for reproducibility lowers the barrier for adoption, and thereby increases the number, quality, and access of reproducible research outputs, as well as increasing public trust in science.
Increased re-use of scientific results: Practical reproducibility enables rapid re-use of published results leading to more effective research.
Democratised science by lowering the barrier to participation in the research process and reproduction of scientific results.

3 Quality and efficiency of the implementation

3.1 Work plan and resources

3.1.1 Quality and efficiency of the implementation

As shown in Figure 3.1.1, the work plan is broken down into five work packages: WP2 focuses on robustness improvements of the Binder tools; WP3 is advancing the Binder tools' feature set to broaden their applicability and increase the impact of the tools and project; WP4 applies and evaluates the reproducibility tools in real-world research contexts; WP5 is focused on engaging with and educating researchers and the wider public in best practices for reproducible science. This is supported by the usual management work package (WP1).

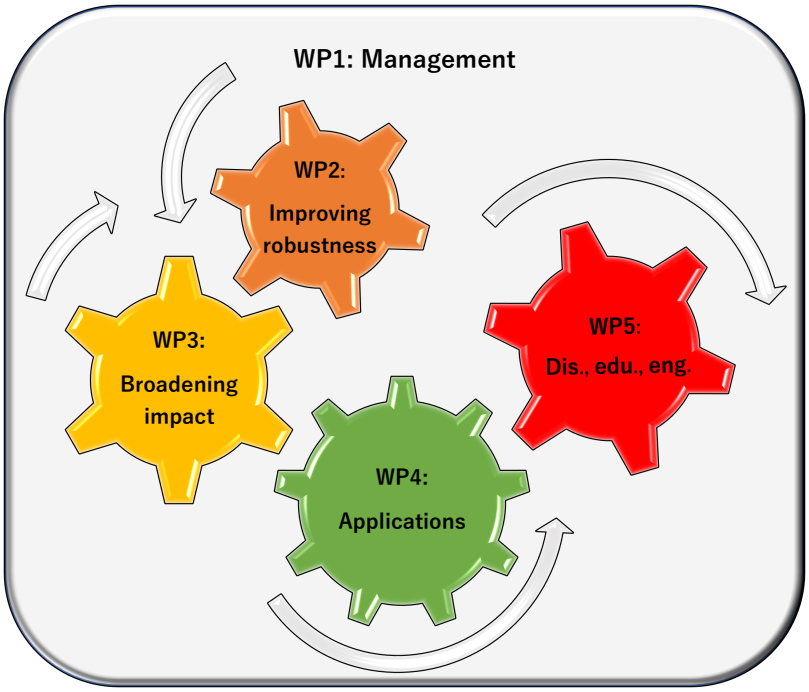


Figure 3.1.1: The relationships and interactions of the work packages, broken up into four categories: management (WP1), development of new functionality surrounding Jupyter's Binder tools to improving robustness and broadening impact (WP2, WP3), applications of tools developed in real-world research context (WP4), and dissemination, education, and engagement (WP5: Dis., edu., eng.).

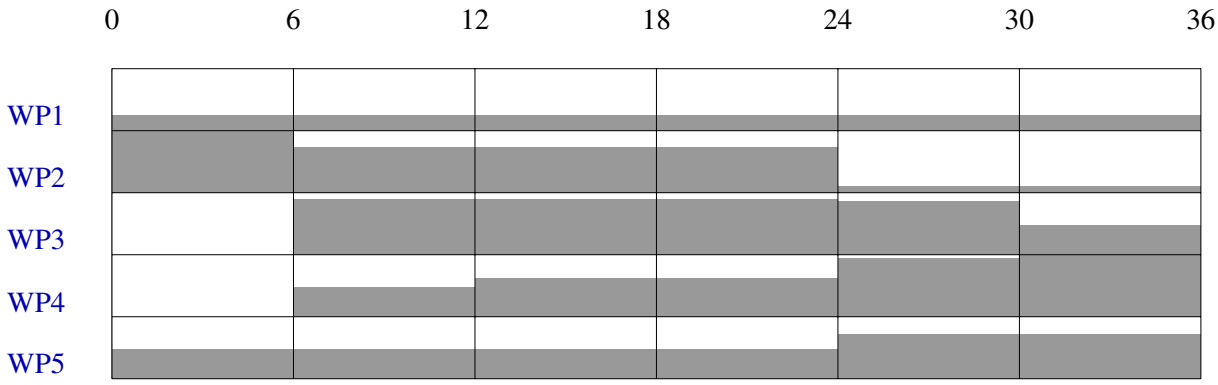


Figure 3.1.2: Gantt chart illustrates the distribution of effort across the work packages over time. The height of grey bars corresponds to total person month effort per month for each work packages. We ramp up overall activity during year 1, then remain at a constant level. In the beginning of the project, the work on core parts of the technology dominates (WP2 and WP3). The activity in application of the new features (WP4) and dissemination (WP5) grows over time.

3.1.2 Work package descriptions

Work Package 1: Project management							**Start month**	0
Lead beneficiary	**Simula**						**End month**	36
Participant	***Simula***	**MPG**	**QuantStack**	**Ifremer**	**UiO**	**all**		
Person months per participant	*12*	1	1	1	1	**16**		

Objectives

The main objective of WP1 is to establish and maintain an effective contract, project, and operational management approach ensuring:

- Timely and successful implementation of the project; including administrative and legal coordination
- Technical management and quality assurance
- Risk and innovation management of the project as a whole; including data and IPR management
- Smooth communication and interaction with the EC and other interested parties

Description

The project will be managed by Simula, which has extensive experience in administering and leading EU funded and national projects. The coordinator together with the WP leaders, will be responsible for monitoring WP status, coordination of work plan updates and annual internal progress reports.

T1.1 Administrative Management; Sites: **Simula** (lead), **MPG**, **QuantStack**, **UiO**, **Ifremer**
The task includes the following activities:

- Preparation, distribution and maintenance of all contractual documents (Consortium Agreement, Grant Agreement and all other legal frameworks).
- Establishment of appropriate communication and collaborative environment for the consortium, as well as the EC and other relevant academic and industry stakeholders (the project website, intranet and communication procedures) to organise transfer of knowledge, present and promote project results (D1.1).
- Organisation of project review and progress meetings.
- Performing qualitative and quantitative risk analysis, planning risk mitigation and control.
- Progress and Financial Reporting to the EC.
- Data and IPR Management will be managed in accordance with agreed rules stated in the Consortium Agreement and in accordance with the Data Management Plan (D1.3).

T1.2 Technical Project Management; Sites: **Simula** (lead), **MPG**, **QuantStack**, **UiO**, **Ifremer**
The task includes the following activities:

- The scientific and technical management to ensure coherent quality and soundness of the work and results.
- Applying quality assurance measures across all partners for all tasks and deliverables.
- Reporting of outcomes and quality assurance activities in technical reports and reviews.
- The project coordinator, with the help of the work package leads, will regularly review technological risks and recommend mitigation plans to minimise or remove them. This will be reported on at each reporting period in the project's technical report.
- Set up and maintenance of technical management infrastructure required for a software project of this type, such as a web site, open source hosting of code and documentation, mailing lists, task trackers, automatic tests and continuous integration. We will feed back into existing open source repositories projects where they exist already, and make use of commonly used tools and services such as GitHub. All outputs will be published under an open source license.

T1.3 Management of dissemination and communication activities; Sites: **Simula** (lead)
This task comprises the management and administrative aspects of all forms of direct dissemination and

public communication activities such as press releases, scientific and technical publications, seminars, talks, promotion through social media, creation of advertisement materials such as flyers, posters, and electronic feeds as well as their distribution. We will use standard community building technology such as mailing lists, wikis and forums, to ensure dissemination to and engagement with the user community.

T1.4 Community Engagement Panel; Sites: **Simula** (lead), **MPG**, **QuantStack**, **UiO**, **Ifremer**

The task includes the following activities:

- Form the community engagement panel (Section 1.2.8) by inviting representative of relevant communities. Ensure that representatives from stakeholder communities include current and potential future Binder users.
- Organise regular (online) community engagement panel meetings, soon after the beginning of the project, and subsequently at the end of years 1, 2, and 3.
- Ensure that input and feedback from community engagement panel members are considered to direct the project to improve the usefulness of Binder tools and broaden the range of their applicability to maximise overall impact.
- Encourage and foster voluntary collaboration and direct contributions to the project from the communities represented in the community engagement panel that go beyond the advisory role of the panel itself.

Deliverables:

D1.1 (Due: 2, Type: DEC, Dissem.: PU, Lead: Simula) *Basic project infrastructure (web site, mailing lists, issue trackers, mailing lists, repositories)*

D1.2 (Due: 3, Type: R, Dissem.: PU, Lead: Simula) *Detailed dissemination, communication, and exploitation plan.*

D1.3 (Due: 6, Type: DMP, Dissem.: PU, Lead: Simula) *Data Management Plan*

D1.4 (Due: 32, Type: DMP, Dissem.: PU, Lead: Simula) *Revised Data Management Plan*

Work Package 2: Improving robustness of reproducibility tools					**Start month**	0
Lead beneficiary	**QuantStack**				**End month**	36
Participant	**Simula**	**MPG**	***QuantStack***	**all**		
Person months per participant	23	2	*12*	**37**		

Objectives

- to better understand and evaluate successful reproduction of computational environments
- to improve the practical reproducibility of environments constructed with SOURCE tools
- to support and maintain core Binder software infrastructure in order to keep it healthy and useful for open science and reproducibility

Description

This work package is focused on making `repo2docker` do the things it does already *better*, *more robustly* and *more sustainably*. (Orthogonal to those improvements, we plan to significantly extend the `repo2docker` features and use cases in WP3.)

To be able to asses the impact of our planned improvements, we need to have a metric. We will create this in Task **T2.1**. In addition to the evaluation of the improvements in this proposal, this can be used more generally as an indicator for reproducibility of software environments.

One major improvement to the existing capabilities of `repo2docker` is the *time-machine* functionality, implemented in **T2.2**.

In task **T2.3**, we will speed up the execution time of `repo2docker` to improve the user experience when reproducing or re-using existing software and data.

Open source software needs ongoing maintenance to adapt to changing requirements and dependencies. We schedule a certain amount of time for this in task **T2.4**.

All changes to the software will be made available online as open source already during development (i.e. throughout the whole project), and new features will be made available through software releases of the Binder tools. A final release will be made and reported through the deliverable D2.3.

T2.1 Towards quantifiable progress for reproducible software environments; Sites: **Simula** (lead), **MPG**

The `repo2docker` tool is a key component of the Binder software for reproducibility (see 1.2.6.3). It can be used to create a software environment based on software dependency specification standards (see 1.2.6.1) that are widely used.

If the required software is specified – for example through a `requirements.txt` file for Python dependencies – then `repo2docker` can create the software environment (currently limited to such environments in Docker images), within in which the main computation or data analysis can be reproduced.

In this task, we will develop a tool – with working name `repo2docker-checker` – that allows us to *automatically* assess the reproducibility of software environments for software that is publicly available on GitHub, Bitbucket or GitLab repositories.

For every repository, the `repo2docker-checker` tool will report if an appropriate software environment could be produced, or if a problem occurred. Software environments in repositories may be reproducible because the authors already use Binder to offer their repository in an interactive Binder environment. Or the software environment may be reproducible because the authors have followed standard conventions understood by `repo2docker`.

The task includes the following activities:

- Through manual inspection of selected repositories, identify common failure modes of building of the software environment (such as for example not specifying the Python version to use).
- Design and develop the `repo2docker-checker`. A prototype exists.[33]
- Where possible, identify for what reason the software build has failed.
- Develop a strategy and heuristic to evaluate success of the build process.
- Identify suitable software repositories for the study.
- Automate the software reproduction process for the available repositories.
- Automate the analysis of the results, so the study can be repeated later.
- Carry out the study to estimate the fraction of reproducible repositories. (This is one of our KPIs, see Section 2.1.2.)
- Repeat the study after the robustness of `repo2docker` has been improved (**T2**) to evaluate progress.

The tool will be made available as open source (D2.1).

T2.2 repo2docker development; Sites: **Simula** (lead), **QuantStack**

This tasks improves the robustness of `repo2docker`. We illustrate this with one specific example: Often, a repository of scientific results may specify which software library is required (such as the Python library `pandas`), but not which version.

A software environment creation tool – such as `repo2docker` – can then attempt to install the most recent version of `pandas`. This is usually the intention of the authors, and was correct *at the time the repository was created*. However, as time moves on, the interface, behaviour and dependence on other packages of `pandas` will change, and at some point an automatic build of the software for the whole repository may fail because of conflicting dependencies.

We have found through prior study[27] that these problems can be overcome if a `pandas` version can be chosen that was the most recent at the time when the repository was created. A related issue is that the Python version itself (such as 3.8, 3.9 or 3.10) may not be specified at all.

We will teach `repo2docker` to establish the date of publication (or last modification) of the repository, to determine the appropriate version of software libraries from that time, and to select libraries with those versions if no specific version is specified.

[33] https://github.com/minrk/repo2docker-checker

In the context of Python packages, we can use the `pypi-timemachine` package [34], and we will implement a similar fetaure in the context of conda packages.

T2.3 Performance optimisation; Sites: **QuantStack** (lead), **Simula**

The creation of reproducible software environments can take – depending on the complexity and overall size of the destination environment – quite some time. Often, an image can be built within few minutes, but there are environments that can take much longer.

In this task, we will profile and optimise `repo2docker` performance, both in terms of build time and image size (can be several gigabytes), which contributes to user-experienced performance as launching a large image can take longer than a small one when the image must be transferred across a network.

T2.4 Maintenance of open source reproducibility software; Sites: **Simula** (lead), **QuantStack**

Developing software that people will use requires maintenance of that software, not just new development. Through this proposal we will contribute general support to open source reproducibility software where this is helpful for SOURCE. Such contributions are expected to the Jupyter and Binder code bases. These projects support not just all participants in SOURCE, but also millions of people relying on Jupyter software.

Deliverables:

D2.1 (Due: 12, Type: OTHER, Dissem.: PU, Lead: Simula) *Release software tool for checking of reproducibility of software environments (`repo2docker-checker`)*

D2.2 (Due: 24, Type: R, Dissem.: PU, Lead: Simula) *Summary of reproducibility improvements achieved.*

D2.3 (Due: 24, Type: R, Dissem.: PU, Lead: Simula) *Release of `repo2docker` with improve robustness features.*

Work Package 3: Broadening impact					**Start month**	6
Lead beneficiary	**Simula**				**End month**	36
Participant	***Simula***	**MPG**	**QuantStack**	**Ifremer**	**all**	
Person months per participant	*24*	12	2	4	**42**	

Objectives

This work package extends the functionality of the SOURCE tools for reproducibility to broaden their applicability and increase impact. WP4 exploits these technological advances in real-world use cases.

The objectives of this work package are to

- remove the dependency of `BinderHub` on Kubernetes
- support container technologies other than Docker
- extend the range of software specification standards that are recognised and supported by `repo2docker`, such as Poetry for Python, renv for R, or Maven for Java.
- enable access to data from data sources outside the container
- enable `repo2docker` to extract and save the software installations instructions (independent from container generation)

Description

One of the design decision that have led to the current Binder software stack is to constrain the supported infrastructure to a few key components. These include:

- Software environments can only be created inside Docker container

[34] https://github.com/astrofrog/pypi-timemachine

- To run and orchestrate (multiple) Docker containers, a Kubernetes system must be available
- The user must interact with Binder environments through a BinderHub installation
- Access to data is not considered, and often challenging or expensive, depending on where the data resides.

The benefits of such a restrictive approach are that the software development and maintenance effort is kept small: the wider the range of supported infrastructure that the Binder tools can be deployed on, the higher the complexity.

At the same time, these restrictions prevent the following scenarios:

- To run the Binder software on systems where Kubernetes is not available (such as the Desktop of a scientist). A related use case is "Binder@home" (**T4.2**).
- To access and use significant amounts of data from inside the software environment is not wellsupported. An important use case that cannot be supported due to this restriction is that of "data publishing": the idea is that a (potentially large and/or complex) dataset is published *together* with software that encodes the necessary knowledge to extract meaningful data from that dataset. A Binder environment would make this dataset accessible in an interactive environment. This is needed for the data publishing use cases (**T4.3**).
- To use Binder on systems where Docker cannot be used. A use case for this is to create reproducible environments on HPC installations. The HPC administrators generally avoid use of Docker for security reasons, but much prefer to support container technologies that can be executed without administrative privileges). This prepares the "Binder@HPC" use case (**T4.4**).

T3.1 Support more software specification standards; Sites: **Simula** (lead), **MPG**

There are two different aspects of software specification that `repo2docker` needs to understand:

- the specification of the software environment, for example through `requirements.txt` files, etc (see Section 1.2.6.1 for more details on the currently supported standards),
- from where to retrieve the software itself: this will be different on a GitHub repository or a Zenodo archive (see also 1.2.6.1 for supported repositories).

For both aspects, there are requests from potential Binder users to extend the capabilities of `repo2docker`. In this task, we will prioritise such requests and extend the `repo2docker` functionality to support as many new standards as possible to best meet community needs.

T3.2 Reducing technical constraints to enable broader usage; Sites: **Simula** (lead), **MPG**, **QuantStack**

In this task, we refactor and extend the existing code to be more flexible. In particular, we want to address and remove constraints that currently exist:

- Dependency on existing Kubernetes system: currently, BinderHub can only start container environments within a Kubernetes installation. In this task, we will make it possible to start a container directly on the host machine. Setting up Kubernetes system is a complex task: while justified to exploit large computational resources through swarms of containers effectively, it is not necessary for single-machine execution of Binder-reproduced environments (such as anticipated for Binder@home in **T4.2**).
- Support of only Docker containers to host created software environments. Docker is widespread and popular, and in particular has an attractive user interface for Windows and macOS. However, many HPC centres refuse to allow use of Docker containers on their systems for security reasons. In this task, we will make it possible to user container technologies, which are more acceptable to use on large HPC installations, such as Singularity.

These steps are essential to enable the Binder@home (**T4.2**) and the Binder@HPC (**T4.4**) use cases.

T3.3 Support more use patterns; Sites: **Simula** (lead), **MPG**, **Ifremer**

In this task, we provide the technical possibilities to use the Binder tools in better and new ways. These are used and evaluated with real world applications in WP4.

- The `repo2docker` tool searches a given repository for specifications of software dependencies, and carries on to compose instructions to install all of the software dependencies within a Docker container image. In this task, we modularise this functionality, and make it possible to extract the required instructions on their own (for example into a stand-alone shell script).

Such functionality could be used to:
  - Extract the list of installation commands to carry out a local install of the required software, or an installation within any other environment.
  - In particular on HPC systems, it may be necessary to install software directly on the host, and this functionality would simplify that.
  - Having the installation instructions neatly summarised will also help the interested scientist to understand what software specifications are (explicitly or implicitly) given within the repository.
- Improve and document the options to access external data resources from within the computational environments generated by BinderHub deployments. This is a prerequisite for **T4.3**.
- Deploy BinderHub with authenticated data access - useful for many institutes who want to offer reproducibility services and data hosting but need to limit or control access to those systems (to ensure, for example, that the computational resources are not abused).
- Support using Binder features without a full BinderHub service. A use case for this is to create a Binder-like experience on a local Desktop (see **T4.2**).
- Support deploying Binder for non-interactive use cases.
  Binder's original design goal was to allow interactive execution of notebooks within a software requirement that provides all the software required by the notebook. This may include compiled and highly customised software, which might produce output files, which are then processed and visualised in the notebook.
  Here, we will provide the foundations for reproducibility work that is non-interactive and may not use Jupyter notebooks. A use case for this is reproducibility at High Performance Computing facilities, where often the computational tasks cannot be carried out interactively (see **T4.4**), but the software environment creation problem solved by repo2docker is the same.

Deliverables:
D3.1 (Due: 12, Type: OTHER, Dissem.: PU, Lead: Simula) *Release new `repo2docker` feature that exposes the command to install identified software environments in stand-alone script*
D3.2 (Due: 36, Type: OTHER, Dissem.: PU, Lead: Simula) *Final open source release of SOURCE tools, completed with automatic testing and documentation.*

Work Package 4: Applications and use cases					**Start month**	6
Lead beneficiary	**MPG**				**End month**	36
Participant	**Simula**	***MPG***	**Ifremer**	**UiO**	**all**	
Person months per participant	3	*15*	14	8	**40**	

Objectives

The objectives of this work package are
- to guide the development of core tools in WP2 and WP3 by simultaneously applying them to real-world use cases from various scientific fields
- to do this together with active scientists from these fields to ensure we develop tools which can cater for a broad European and global research community
- to demonstrate how the tools we develop can support more reproducible and reusable science

Description

Whilst the components issued from work packages WP2 and WP3 will be made available as generic tools for reproducible open science, this work package is focused on using (and if required tailoring) them to make them suitable for specific real-world cases, described in 1.2.5.

The use-cases we anticipate are

- **T4.1** Applications and best practice examples that demonstrate use of Binder for more reproducible and reusable science.
- **T4.2** Ability to recreate software environments and re-produce results using local compute resources (such as the Desktop of a researcher)
- **T4.3** Use of Binder to facilitate access to large datasets where the published resource does not only include the data itself, but also software to access and read the data.
- **T4.4** Use of Binder at High Performance Computing facilities to re-produce computationally more demanding results

Working collaboratively with core Binder developers and research active scientists, we merge state-of-the-art knowledge on what is technically possible with the understanding of the scientists what reproducibility features would significantly improve their workflow. We expect that – in addition to iteratively refining the features of Binder – we will also inspire each other to find out-of-the box solutions that each group on their own may not come to think about.

The tasks in this work package serve both as requirements capture (and so to inform and guide the work in WP2 and WP3) and evaluation of the technological advances of the Binder tools. The completed demonstrators will also feature in our best practice guidelines and dissemination activities in WP5.

We will use the regular technical reports to update on progress. An interim report (D4.1) and final report (D4.2) will summarise the results. Documentation of best practice guidelines will be developed as an open document throughout the project, used in WP5, and be submitted as deliverable at the end of the project (D5.1).

T4.1 Science demonstrators; Sites: **MPG** (lead), **Ifremer**, **UiO**

In this task, we want to demonstrate the value and usefulness of WP2 and WP3 with real scientific use cases from the research communities involved in SOURCE (see Section 1.2.5 on page 10 for the initial set of science applications).

The demonstrators are designed to exploit the solutions developed within SOURCE (such as Binder@home, Binder@HPC, data publishing) and leverage existing institutional and/or national e-infrastructures as well as core EOSC services. Synergies between the different science applications and communities will be ensured through the technical tasks (**T4.2** Binder@home, **T4.3** data publishing, and **T4.4** Binder@HPC).

T4.2 Binder@home; Sites: **Simula** (lead), **MPG**, **UiO**, **Ifremer**

In this task we target the compute power of the desktops of individual researchers and users to recreate software environments in which computational results can be reproduced and re-used: We envision to provide an experience identical or similar to that of using BinderHub, but only using the local computer available to the user as the compute resource.

Context: Reproducibility services using Binder currently rely on hosted Binder instances. The BinderHub Federation provides such a global service at mybinder.org which is free at the point of use. It delegates the building of the software environment and re-execution of the code to a small number of computer centres that have volunteered to contribute compute resources. It may hence be possible to overload the system, see Section 2.1.3 for current usage. To allow the up-scaling of good reproducibility practice, it is desirable not to depend exclusively on such a single hosted service (or even multiple similar hosted services).

Moreover, the compute resources typically offered through mybinder are modest, namely at most 2 GB of RAM and a single CPU core. In contrast, most laptops and desktops have similar or better hardware capabilities than the free mybinder.org cloud-computing resources currently offer, making it highly desirable to leverage these local resources. Finally, having the sovereignty to host reproducibility computation on their own machine, makes researchers independent from national or international service providers.

Task activity: Based on the preparations in WP2 and WP3, we intend to extend Binder such that users of the service can carry out the building of the environment, and – if desired – the launching of a notebook server *on their own hardware*, e.g. their laptop or on-premise or cloud infrastructure. The working title for such functionality is "Binder@home" (as a reference to the crowd-based SETI@home search for extraterrestrial intelligence at home.[35])

[35] https://setiathome.berkeley.edu

We will design, implement, and test the "Binder@home" functionality, and make it available as part of the Binder software. That new 'Binder@home' software component will essentially complement BinderHub with an easy-to-use local single-user case, it will trigger the local build of the software environment, the start of the Jupyter notebook server, and the opening of the relevant local URL and port in a browser. This effort will bridge the gap from mybinder.org to "Home"; by giving the freedom and digital sovereignty to the researchers to chose where they execute their computational experiments in a simple manner, without sacrificing the convenience of the mybinder.org service.

T4.3 Data publishing; Sites: **MPG** (lead), **Ifremer**, **UiO**

The task focuses on the use of reproducible software environments within Binder to provide working and interactive code that provides access to large or complex datasets.

Context: Scientists would like to publish their data. Such a data publication must include the dataset itself, and metadata that explains how to interpret the data. In addition to such information, it can improve the quality of the dataset if *computer executable libraries or commands* are provided, which simplify the reading of the actual data files. Such routines encapsulate meta information about the data file (format and structure) in a machine-readable format.
A binder-enabled repository can provide the access to such datasets by containing the specification of a software environment and hosting of the file-reading routines together. Such an approach significantly simplifies the re-use of the data (or reproduction of existing study) because the data reading routines, which can be very complex, do not need to be re-implemented.

Task activity: We will design and implement functionality that allows such data publishing based on Binder tools.
A major challenge is the link to the data: ideally, datasets are hosted on a separate infrastructure (such as archives, or files published together with a publication — for example on Zenodo). It will thus be necessary to reference the data on this data-holding archive and the data location within that resource. This data location will need to be used from the notebooks to access the data.
Some authentication for data access may be required: either because the data is not meant to be fully public, or because access to the data creates significant cost for the hosting party. Such authentication information/credentials from (the BinderHub) login must be passed to the point where the data-holding medium is mounted in a container.
We will work very closely with the Max Planck Compute and Data Facility (MPCDF) to prototype such functionality. We will allow access to datasets that the MPCDF hosts themselves.
An important outcome of this task is an evaluation of the chosen design and implementation, to propose a more generic model for the next feature extension of the Binder tools.

T4.4 Binder at HPC facilities; Sites: **MPG** (lead), **Simula**, **Ifremer**, **UiO**

In this task, we want to broaden the applicability of the Binder tools to become more useful in High Performance Computing (HPC) environments. In particular, this requires parallel execution of software based on the reproduced computational environments.

Context: Reproducibility of data created on HPC resources is difficult. In addition to a specification and access to the actual (simulation) software and dependencies, one may also need specialised HPC hardware to be able to execute the software.
The notebook interface – which works well for many examples in computational and data science – may not be appropriate for HPC applications, where jobs of substantial run-time are typically submitted to a queuing system, which will trigger execution of the (parallel) job, once the required resources have become available.
It is outside the scope of this proposal to find and implement a generic reproducibility approach for HPC use cases. Nevertheless, we propose to explore some aspects of HPC-reproducibility to influence the development of Binder, and start the – probably iterative – process of finding a generic solution.

Strategy: We focus on reproducible execution of an HPC application to compute data as the step of novelty here. This may well be without using notebooks, but will consist of the building of the software environment and the submission of a batch job making use of this environment to the HPC system's queue.

We assume for this to work that the user (who wants to reproduce some results) needs to login to the HPC resource of their choice, and uses repo2docker to create a suitable software container, and starts execution of those containers "manually", for example through submission of a compute job to the Slurm queuing system. We also assume that the hardware required for this reproduction is available on the HPC system.

Activity: We will use repo2docker to automate the creation of reproducible software environments on an HPC system. If no parallelism is required, this is similar to the Binder@home scenario (**T4.2**). Shared memory parallelisation, using for example OpenMP, will work well with the approach chosen.
A main task here will be to explore the feasibility of using distributed parallelisation (for example through MPI) where for the execution of an MPI-program the processes on the nodes run in containers but communicate via MPI as usual. We evaluate the situation with HPC software that allows MPI-parallelisation (such as Octopus). Subsequently, we will share our experience with reproducible software environments in HPC contexts (D4.1, D4.2).

Challenges: We expect that we need to use a container technology that is widely accepted at HPC sites (for example Singularity [20] or CharlieCloud [31]), as Docker on HPC systems is typically avoided due its strict root-user requirements [12].
Another challenge is that of using accelerators such as GPU cards which are installed on the host but need to be accessed and instructed efficiently from the software running inside the container. Similarly, for HPC systems, there are specialised drivers for high-performance network cards: can these be used from the container environment and what is the performance impact when doing so? [22]
These investigations will guide us in answering the following important questions: Should hardware requirements be archived in the reproducible repository? If so, what specification should we use?
The existing buildpacks that repo2docker supports 1.2.6.1 may need to be extended for HPC specific software provisioning tools such as Spack or Easybuild.

Deliverables:
D4.1 (Due: 18, Type: DEC, Dissem.: PU, Lead: MPG) *Interim report on real world use cases of Binder for reproducible and reusable science*
D4.2 (Due: 34, Type: DEC, Dissem.: PU, Lead: MPG) *Final report on real world use cases of Binder for reproducible and reusable science*

Work Package 5: Dissemination, education and engagement							**Start month**	0
Lead beneficiary	**Ifremer**						**End month**	36
Participant	**Simula**	**MPG**	**QuantStack**	***Ifremer***	**UiO**	**all**		
Person months per participant	7	6	3	*10*	9	**35**		

Objectives

A key focus of this work package is to disseminate the results of this project, including the technical advances and guidance for best practice for reproducible science. This includes educating researchers about the value of open science, reproducibility and re-usability as well as the possibilities of integrating Binder tools in their workflows.

Beyond this activity, which is directed from the project members to the wider community of scientists, we use this work package to engage with researchers and stakeholders and seek input from them to the project. Desired input includes requirements for practical reproducibility in the different domain as well as technical contributions – for example through merge requests for Binder tools, or open source documentation of best practice for reproducible software environments. Strong collaboration with the Community Engagement Panel

is expected to take place.

Our dissemination, education and engagement objectives includes:

- Ensure awareness of the results of the project in the user community, and in particular in those groups that act as educators and multipliers of knowledge (such as the Carpentries and research infrastructure organisations).
- Educate the community on the value of open science, and in particular train researchers in best practices for open and reproducible science.
- Produce training and education material to disseminate the ability to publish reproducible computational science outputs using the tools we improve and develop.
- Address the shortage of researchers and research support staff trained in practical reproducibility.
- Provide documentation and tutorials which can serve as the technical components of reproducibility policies.
- Throughout these activities engage with users and stakeholders, to listen and understand their barriers or incentives towards more reproducible science, and the usability of the SOURCE outputs.

Description

Open science and reproducible science is entirely dependent on researchers adopting open practices.

We address this challenge in multiple ways:

1. The philosophy of the Binder tools is to respect existing standards and best practice (and not to invent additional syntax or requirements). It is thus possible to use the Binder tools (to recreate a software environment) even if the repository authors did not anticipate the use of Binder, or knew about their existence. In the best possible scenario, a scientific research output becomes automatically reproducible with Binder **without the author having to know** about Binder or **having to invest additional effort** (beyond following best practice).

2. In this work package, we produce education materials and carry out education activities to spread the knowledge about **good practice for reproducibility and re-usability in science**, such as for example automation of all analysis steps, and complete documentation of the required software stack. Only one aspect of this training is to show how Binder can help with reproducibility.

3. Throughout the activities of this project and the engagement with the wider science community through the Community Engagement Panel, we aim to **understand the underlying drivers** for acceptance, rejection or lack-of-interest **in adopting practices that lead to reproducible science results**.

Science applications (WP4) will also support the creation of tutorials and **best practice guides for reproducibility** (**T5.1**), and offer interactive (online, hybrid and/or in-person) workshops (**T5.2**) to help disseminate the content more effectively.

We will participate in the well established academic dissemination activities, and events of the European e-infrastructure projects and other relevant structures. EGI is a member of our the community engagement panel (**T1.4**) and the interaction with them will be useful to prioritise our resources in this very active field.

T5.1 Best practice guidelines for reproducible science; Sites: **Ifremer** (lead), **Simula**, **MPG**, **UiO**

The aims of this task are to (i) provide online resources for researchers on more reproducible and reusable science, and (ii) support delivery of our workshops (**T5.2**).

This task includes the following activities:

- Collect and compose **best practice guidelines** for reproducible and reusable science (D5.1).
- Split the content into multiple topic areas and target audiences so learners with different prior knowledge and needs can be directed to the most relevant content.
- Develop lesson materials on **open science** best practices (version control, testing, automation of all steps, collaboration and peer review, documentation, software licensing and open source, use of Jupyter notebooks).
- Develop lesson materials on **reproducible computational science**, which focuses on combining the open science tools for reproducible science.

- Develop materials on **using Binder tools to make science more reproducible and reusable**. This includes addressing and describing the use cases from WP4.
- Collaboration with the CodeRefinery project and The Carpentries (Carpentries incubator and Carpentries Lab) for the development and maintenance of the online lesson materials and delivery of workshops.

All material will be licensed under an open license such as CC 4.0 (D5.1).

T5.2 Training Workshops for more reproducible science; Sites: **Ifremer** (lead), **Simula**, **MPG**, **UiO**

This task is focused on taking the content from the **T5.1** (Best practice for reproducible science guidelines) and disseminating it through various channels and to different target audiences.

To achieve these goals, the following actions/activities will take place:

- Delivery of workshops on (i) open science, (ii) reproducible computational science, and (iii) the use of Binder tools to support this.
  The content is focused on key insights and tools need for more reproducible science, but will be contextualised and delivered in the wider field of Findable, Accessible, Interoperable and Reusable (FAIR) software and data.
- SOURCE Admin trainings: we will offer training events for learning on how to deploy SOURCE services such as BinderHub. This will be relevant for a very small (but important) group of users, i.e. those that want to host their own BinderHub instance. We know from multiple research organisations that this desire exists.
- Where possible (for instance after consultation of the Community Engagement Panel), we will schedule dissemination events to take place during conferences and community events, such as PyData, EuroSciPy, Supercomputing meetings.
- We will archive recordings of the training events to support the increasing desire of learners to make use of online streaming services (such as YouTube) to work through a learning programme at their own time and pace.
- We will offer in-person, remote and hybrid training.
- The work will be carried out in collaboration with the CodeRefinery project (the University of Oslo is a partner of the CodeRefinery project) and will make available its network of instructors and helpers to co-organise, advertise and run online workshops on open science best practices.
- We will detail our executed activities through the reporting at the end of each reporting period.

T5.3 Community support and engagement; Sites: **Simula** (lead), **MPG**, **QuantStack**, **UiO**, **Ifremer**

A project such as SOURCE has the ambition to develop a small set of tools that will **impact many researchers** and have the potential to be useful **across all scientific domains that need electronic data processing** as part of their scientific research and publication process.

As such, we expect that the demand through support queries, documentation clarification questions, and helpful feedback will be substantial. With this task, we explicitly reserve some time for such activities.

This task complements the Community Engagement Panel and address multiple aims simultaneously:

- to engage with community members (and potentially their computing support staff) to **help them best use and exploit** the Binder tools. This can range from helping to configure a BinderHub installation, to address usage questions of tools such as `repo2docker` in domain-specific contexts;
- to engage with community members to **better understand diverse requirements**, and use this information to make the Binder tools and reproducibility guidelines more useful for a wider diversity of scientific domains;
- to engage with community members to **train researchers and research software engineers in reproducibility practices** and tools (to address a shortage of staff with such skills)
- to engage with community members to **invite them to contribute** to the binder tools, the reproducibility guidelines and policy development, and other open source tools.

We will achieve those aims through listening to feedback, queries and requests for help from the community, and reserve time to respond. Depending on the complexity of an issue, guidance by email, chat, video meeting or even an in-person visit may be appropriate. (When demand exceeds the time budget, we will prioritise which issues we can deal with first.)

We know from our experience with running and contributing to open source projects that **such engagement activities are effective in training** interested and often highly skilled scientists and research software engineers to become contributors to open source projects. While they may have a primary interest in improving

an open source tool to suit their needs, this will likely benefit others as well. Once somebody has contributed to a particular open source software tool, they are more likely to make follow-up contributions — for example to improve documentation.

Deliverables:
D5.1 (Due: 24, Type: R, Dissem.: PU, Lead: Ifremer) *Best practice guide for reproducible science with Binder.*
D5.2 (Due: 36, Type: R, Dissem.: PU, Lead: Ifremer) *All training sessions material completed, reviewed, and published online.*

3.1.3 Deliverables

#	Deliverable name	WP	Lead	Type	Level	Due
D1.1	Basic project infrastructure (web site, mailing lists, issue trackers, mailing lists, repositories)	WP1	**Simula**	DEC	PU	2
D1.2	Detailed dissemination, communication, and exploitation plan.	WP1	**Simula**	R	PU	3
D1.3	Data Management Plan	WP1	**Simula**	DMP	PU	6
D2.1	Release software tool for checking of reproducibility of software environments (`repo2docker-checker`)	WP2	**Simula**	OTHER	PU	12
D3.1	Release new `repo2docker` feature that exposes the command to install identified software environments in stand-alone script	WP3	**Simula**	OTHER	PU	12
D4.1	Interim report on real world use cases of Binder for reproducible and reusable science	WP4	**MPG**	DEC	PU	18
D2.2	Summary of reproducibility improvements achieved.	WP2	**Simula**	R	PU	24
D2.3	Release of `repo2docker` with improve robustness features.	WP2	**Simula**	R	PU	24
D5.1	Best practice guide for reproducible science with Binder.	WP5	**Ifremer**	R	PU	24
D1.4	Revised Data Management Plan	WP1	**Simula**	DMP	PU	32
D4.2	Final report on real world use cases of Binder for reproducible and reusable science	WP4	**MPG**	DEC	PU	34
D3.2	Final open source release of SOURCE tools, completed with automatic testing and documentation.	WP3	**Simula**	OTHER	PU	36
D5.2	All training sessions material completed, reviewed, and published online.	WP5	**Ifremer**	R	PU	36

3.1.4 Milestones

1. **Milestone M1 (Month6) Select conda packages by date** The conda/mamba package manager shall be able to select packages for installation based on a given date. Necessary for repo2docker to best take time into account when creating a software environment.

2. **Milestone M2 (Month12) Reproducibility study and evaluation tool** We will have preliminary study results and an associated tool, regarding the reproducibility of repositories with repo2docker. These results will inform future development of the tools in WP2, as well as best practices resources and education in WP5.

3. **Milestone M3 (Month12) Prototype demonstrator services** By this point, prototype demonstrator services will be useful and accessible to a broad range of users, and we will have begun to experiment with early-adopter users and local demonstrators to guide further development in WP3, ensuring that development serves the reproducibility needs of the global science community.

4. **Milestone M4 (Month12) Draft best practices documentation** Draft version of documentation for best practices is online. Required starting point for education tasks in WP5.

5. **Milestone M5 (Month15) Support for alternative container technologies in repo2docker for suitability in HPC** The Docker container runtime is not suitable in all cases. In order to proceed with some demonstrators in WP4, we must ensure compatibility with container runtimes supported by our HPC providers, such as Singularity.

6. **Milestone M6 (Month18) repo2docker takes publication time into account** By taking publication time into account, repo2docker will reproduce environments with higher fidelity, especially when environments are not fully or strictly specified.

7. **Milestone M7 (Month18) Practical support for authenticated data publishing** It shall be practical to deploy BinderHub with performant, authenticated access to large datasets, required for some advanced science demonstrators in WP4.

8. **Milestone M8 (Month24) repo2docker produces robust computational environments** Taking input from earlier study and tests, repo2docker has been improved to produce environments more reliably and robustly, as verified by a comparison study with the baseline at the beginning of the project.

9. **Milestone M9 (Month36) Science demonstrators fully operational** Science demonstrator applications and services are fully operational, enabled by all the developments of the project, and in active use. The most successful demonstrators are prepared for operation beyond the life of the project.

#	Name	Related work package(s)	Due	Means of Verification
M1	*Select conda packages by date*	WP2	6	Feature available in conda/mamba software
M2	*Reproducibility study and evaluation tool*	WP2, WP5	12	Report produced
M3	*Prototype demonstrator services*	WP4, WP3	12	Deployed first functional prototypes of science demonstrators. Early users are able to access and test prototype services
M4	*Draft best practices documentation*	WP5	12	Resources available from project website
M5	*Support for alternative container technologies in repo2docker for suitability in HPC*	WP3, WP4	15	Feature available in repo2docker software
M6	*repo2docker takes publication time into account*	WP2	18	Feature available in repo2docker software
M7	*Practical support for authenticated data publishing*	WP3, WP4	18	Demonstrated example deployment
M8	*repo2docker produces robust computational environments*	WP2	24	Delivered in repo2docker software; Repeat study, comparing baseline results form start of project
M9	*Science demonstrators fully operational*	WP3, WP4	36	Services are running and can be accessed and used by users.

Table 3.1.2: List of Milestones

3.1.5 Risks and risk management strategy

Risk (**likelihood / severity**)	**WP**	**Proposed risk-mitigation measures**
		General technical / scientific risks
Implementing infrastructure that does not match the needs of end users (**Low/High**)	all	Many of the members of the consortium are themselves end-users with a diverse range of needs and points of view; hence the design of the proposal and the governance of the project is naturally steered by demand; besides, because we are building tools, users have the flexibility to adapt the infrastructure to their needs. In addition, the open source nature of the project facilitates and promotes the involvement of the wider community in terms of providing feedback and requesting additional features via platforms such as GitHub and Bitbucket on a regular basis.
Lack of predictability for tasks that are pursued jointly with the community (**Low/Medium**)	2, 3	The PIs have a strong experience managing community-developed projects where the execution of tasks depends on the availability of partners. Some tasks may end up requiring more effort from SOURCE to be completed on time, while others may be entirely taken care of by the community. Reallocating tasks and redefining work plans is common practice needed to cater to a fast evolving context. Such random factors will be averaged out over the large number of independent tasks.
Reliance on external software components (**Low/Medium**)	2, 3, 4	The non-trivial software components SOURCE relies on are open source. Most are very mature and supported by an active community, which offers strong long run guarantees. The other components could be replaced by alternatives, or even taken over by the participants if necessary.
		Management risks
Recruitment of highly qualified staff (**Medium/High**)	all	The majority of positions funded by SOURCE are already hired. Only two positions are to be filled, both full-time research software engineers, and partners have much experience hiring excellent staff at attractive sites. In addition, we have a critical mass of qualified staff in the project enabling us to train and mentor new recruits.
Different groups not forming effective team (**Low/Medium**)	all	The participants have a long track record of working collaboratively across multiple sites. Thorough planning of project meetings, workshops and one-to-one partner visits will facilitate effective teamwork, combining in-person and remote collaboration.
Partner leaves the consortium (**Low/High**)	all	If the Grant Agreement requires a replacement in order to achieve the project's objectives, the consortium will invite a new relevant partner in. If a replacement is not necessary, the resources and tasks of the departing partner will be reallocated to the alternative ones within the consortium.
		Dissemination risks
Impact of dissemination activities is lower than planned. (**Low/Medium**)	5	Partners in the consortium have a proven track record at community building, training, dissemination, social media communication, and outreach, which reduces the risk. The Project Coordinator will monitor impact of all dissemination activities. If a deficiency is identified, the consortium will propose relevant corrective actions.

Table 3.1.3: Initial Risk Assessment

WP	Title	Simula	MPG	QuantStack	Ifremer	UiO	total
WP1	Management	*12*	1	1	1	1	**16**
WP2	Improving robustness	23	2	*12*		0	**37**
WP3	Broadening impact	*24*	12	2	4		**42**
WP4	Applications	3	*15*		14	8	**40**
WP5	Dissemination, education, and engagement	7	6	3	*10*	9	**35**
totals		**69**	**36**	**18**	**29**	**18**	**170**

Efforts in PM; WP lead efforts light gray italicised

Table 3.1.4: Summary of staff effort

3.1.6 Resources to be committed

Table 3.1.4 shows the distribution of person months across participants and work packages.

Purchase costs All participants request less than 15% of personnel costs in purchase costs. These costs enable our work plan through supporting:

- project meetings
- site visits between project members to foster collaboration
- conference attendance for dissemination
- open access publication fees and communication activities
- equipment for carrying out the work (high performance laptop computers for each FTE)
- hosting workshops for dissemination (in budget of lead site **Simula**)
- cloud computing costs for testing outputs and supporting development and workshops (**Simula** budget)
- Certificate on Financial Statements (CFS) (**Simula** only)

3.2 Capacity of participants and consortium as a whole

3.2.1 Consortium composition

The SOURCE consortium spans the broad spectrum of actors required for successfully developing and disseminating tools and infrastructure for open and reproducible computational science, catering to the needs of the European and global scientific community. It is composed of one academic institution (University of Oslo), three research organisations (Max Planck Gesellschaft, Ifremer, Simula), and one SME (QuantStack) based in three different countries (Norway, France, Germany).

The consortium has developed through collaborations and common interests. Some partners have been working together on different aspects of Jupyter development (**QuantStack**, **Simula**), software for education (**QuantStack**, **Simula**, **MPG**) and use of Jupyter tools for reproducible science (**Simula**, **MPG**) for many years. Others contribute significant expertise in the practice of open science and training (**Ifremer**, **UiO**).

Many participants (**MPG**, **UiO**, **Ifremer**) have scientists involved who work on facilitating computational and open science for scientists in their institutions. As such, each of them has experience and a good overview of the requirements for effective science and reproducible science from the many research projects they are connected to. In addition, several of them are research active in scientific domains, reproducibility and education.

The existing Binder tools – which are the baseline for this project – originate from Project Jupyter. We have core Jupyter and Binder developers in our team, and thus direct access to developer expertise and experience.

Finally, we note that all project partners are long time passionate advocates of open and reproducible science; building on highly successful past experience with OpenDreamKit, they *have chosen to write this proposal fully in the open* on GitHub (https://github.com/minrk/horizon-widera-2022) for maximum transparency and engagement of the community. We have used the same open source collaboration tools and practices as the open source open science community.

3.2.2 Complementarity and interdisciplinarity

For the successful delivery of this project and its mission to enable better reproducibility and science through better software (and services built upon it), we need complementary expertise from researchers and research software engineers. As we build on, improve, and advance existing software tools from Project Jupyter, it will be essential to know these well. As our approach will provide automatic reproducible computational environments if best practice is followed by the researchers, the education and training aspect for best practice is also vital for this project.

The chosen consortium ensures a critical mass of interdisciplinary expertise and excellence in key areas (such as natural sciences, education, software engineering, Project Jupyter) with research organisations and SMEs of recognised international reputation:

- A set of use cases that cover several application domains and users, and that impose very diverse requirements on open tools (**MPG**, **Ifremer**, **UiO**);
- Lead developers in the Jupyter Ecosystem, including IPython, the Jupyter notebook, JupyterLab, JupyterHub, Binder, mybinder.org, Jupyter Widgets (**Simula**, **QuantStack**)
- Experts and major promoters of the Jupyter collaborative user interfaces for interactive, exploratory and reproducible computing in a variety of scientific domains (**MPG**, **Ifremer**, **UiO**);
- A long experience and proven track record of success with large and complex collaborative projects, including European E-Infrastructure projects (**MPG**, **Simula**), projects focused on large-scale infrastructures and large experimental services (**MPG**, **Ifremer**), as well as experience in running large scale open source projects (Jupyter project, **Simula**, **QuantStack**);
- Experience in educating students and experienced researchers on computational methods and open science (**Simula**, **MPG**, **Ifremer**, **UiO**);
- A comprehensive range of skill sets and competencies in several relevant domains, from applied research to standardisation to business analysis.

We have budgeted travel funds to visit each other for short periods (of a few weeks) where this is helpful to work more effectively and improve our ability to work together within the interdisciplinary team of participants.

3.2.3 Capacities and roles of participants

Simula Research Laboratory (**Simula**) is an internationally-leading Norwegian research institute in the key ICT areas: communication systems, scientific computing, and software engineering. Dedicated to tackling scientific challenges with long-term impact and of genuine importance to real life, Simula offers an environment that emphasises and promotes basic research. This translates into numerous projects funded by the EU, Norwegian government, or regional institutions, that Simula was involved in.

Benjamin Ragan-Kelley has contributed to the Jupyter Project since its inception as a lead developer, and headed the Numerical Analysis and Scientific Computing department at Simula from 2018-2021. While continuing to contribute to and maintain the open source software around which SOURCE centers, he also researches the effectiveness and usefulness of such tools for education [33] and reproducible science [18, 11, 8, 27, 4].

Simula's role – in addition to managing the overall project – is to provide technical leadership and in-depth expertise of the Jupyter and Binder project, which will be instrumental in the execution of this project. As lead partner of the project, Simula is the largest beneficiary, both as the largest technical contributor, and for project-wide administrative support, as well as the host of some project activities such as workshops, and cloud computing resources.

The **Max Planck Society** (Max Planck Gesellschaft, MPG) is a non-profit research organisation with 86 research institutes and nearly 24,000 staff. For this project, we have representatives from the **Max Planck Computing and Data Facility (MPCDF)** – the organisation's cross-institutional competence centre for computational and data sciences – and staff from the **Max Planck Institute for Structure and Dynamics of Matter (MPSD)**, who are active in condensed matter research and reproducibility research.

The MPCDF operates large state-of-the-art supercomputers, several mid-range compute systems and data repositories for various Max Planck institutes and provides an up-to-date infrastructure for data management including long-term archival. The MPCDF is a member of major European exascale projects, particularly the BioExcel and Novel Materials Discovery (NOMAD) CoE, and of projects such as Big data driven material science (BIGMax), FAIR data infrastructure for Condensed-Matter Physics and the Chemical Physics of Solids (FAIRmat), and Data Infrastructure Capacity for EOSC (DICE).

The MPSD enjoys an international reputation in the field of the ultrafast structural dynamics. The MPSD is currently comprised of 3 departments and several independent research groups, and is one of the partners of the cluster formed with the Center for Free-Electron Laser Science (CFEL), DESY, and the University of Hamburg.

MPCDF and MPSD have in-depth experience in delivering training and workshops on computational methods, including best practice and reproducibility. One team member (Hans Fangohr, MPSD) has a long-standing collaboration with **Simula** and the Jupyter project, and a research interest in the use of open source tools, Jupyter, and Binder for research and reproducible research [6, 10, 8, 4]. Moreover, Hans Fangohr has won awards repeatedly for excellence in design and delivery of teaching activities at different universities. The MPCDF has multi-year expertise in deploying and using Cloud-based JupyterHub and BinderHub installations for various use cases from different scientific domains.

In this project, the team will co-design Binder-based services for reproducible science and data publishing. They will draw from the wide research activities of scientists in the Max Planck Society – including social sciences, humanities and HPC-based activities – to evaluate, improve and apply the Binder tools for use cases such as data publishing and better reproducibility in HPC.

QuantStack is a France-based software corporation specialising in open-source scientific computing. Clients and partners of QuantStack range from financial software companies to robotics startups and public research institutions.

QuantStack's team comprises maintainers and contributors to open-source technologies considered as industry standards and adopted by millions in the world, such as Jupyter, conda-forge, mamba, and many more. It is home to some of the most prolific contributors to the ecosystem.

QuantStack is responsible for some of the main innovations in the Jupyter ecosystem of the past few years. Features developed by the team include the support for *collaborative editing* in JupyterLab, the development of the *JupyterLab Visual Debugger*, *JupyterLite* (an in-browser distribution of JupyterLab leveraging WebAssembly for language kernels), the *xeus Jupyter kernels* (xeus-robot, xeus-cling, xeus-sql, xeus-lua, xeus-python), and many data visualisation libraries such as ipygany, ipyleaflet, and ipycanvas, as well as the Voilà dashboarding system. Beyond the new developments, QuantStack takes a large part of the maintenance burden of the underlying Jupyter components.

QuantStack's open-source development is not limited to the Jupyter ecosystem, as the team is also very active in the conda-forge project, a community-maintained distribution of packages for scientific computing, with tens of thousands of packages available, and hundreds of millions of package downloads monthly. QuantStack is also responsible for the development of the mamba package manager, which has been adopted by the conda-forge and Binder projects, among others.

QuantStack's team will provide expertise to the consortium as core Jupyter developers, and will contribute to the project by improving the performance and reliability of the Binder project for building software environments.

The **French National Institute for Ocean Science (Ifremer)** is a French public scientific and technological institution that works for exploring, understanding and predicting the ocean. A pioneer in ocean science, Ifremer's cutting-edge research is grounded in sustainable development and open science. Ifremer's vision is to advance science, expertise and innovation by creating and sharing ocean data, information & knowledge. Ifremer hosts more than 1,500 personnel spread along the French coastline in more than 20 sites.

Ifremer has a marine scientific computing center, hosting various world-class data for oceanography for different national, European, and international projects. The Pangeo platform is already deployed using JupyterHub and Python environment over HPC resources at Ifremer. Within this project, Ifremer will focus on testing, validating, improving, and applying the developed tools for practical reproducibility in real-world research contexts, such as ocean physics data analysis on satelite to model data and biologging data analysis on fish to track its behaviour and environment. This data and analysis enable biologists to gain a better understanding of fish movement, their preferred habitats, and the environmental conditions they need to thrive, all of which are essential for the future protection of natural resources.

The **University of Oslo** (UiO) is Norway's oldest institution for research and higher education, with 28,000 students and 6,000 employees. UiO aspires to be an international hub for the research-based integration of computing into science education and has financed a university-wide hosting service for Jupyter notebooks through JupyterHub to introduce a computational aspect to all curriculum programs in all science disciplines from bachelor to postdoctoral studies.

The University of Oslo is a Silver Partner to The Carpentries, an international successful community driven

project with Instructors, Trainers, Maintainers, helpers, and supporters who share a mission to teach foundational computational and data science skills to researchers. It is also actively involved in the CodeRefinery initiative that acts as a hub for FAIR (Findable, Accessible, Interoperable, and Reusable) software practices. Twice a year, CodeRefinery organises big online training events with more then 300 attendees each.

The focus of UiO is to use their vast and leading experience in training and communication to educate researchers globally about open science and practical reproducibility, to help translate the technical advances of this project into wide-spread impact.

The University Center for Information Technology (USIT) that represents the University of Oslo in this project is part of the Norwegian Research Infrastructure Services (NRIS), a collaboration between highly qualified IT staff at the four Norwegian universities (NTNU), the Universities of Bergen, Oslo and Tromsø and employees at Sigma2 (the Norwegian National e-infrastructure provider), to pool competencies, resources and services. USIT also organises with NRIS community-specific outreach events to connect with local communities and collaborates with other European initiatives that offer Galaxy training/mentoring efforts in EOSC (EOSC-Nordic, RELIANCE, EuroScienceGateway), and ELIXIR. USIT actively contributes to Nordic e-Infrastructure Collaboration (NeIC) projects such as the Nordic distributed tier-1 facility for the worldwide computing grid serving the large hadron collider at CERN and leads the Nordic Collaboration on e-Infrastructures for Earth System Modeling Tools (NICEST2).

3.2.4 Connections beyond project partners

As our ambition is to **improve practical reproducibility for the global community of researchers**, we need to be well connected to understand requirements and constraints from many domains. To improve our networking and information gathering, we have started to compose our Community Engagement Panel (Section 1.2.8, Task **T1.4**) with the aim to bring together representatives from diverse research domains, research infrastructure providers, research funders, publishers, educators, and policy makers. We expect to also be able to use that network to support communication, dissemination and exploitation of our results.

The project partners are research active in current topics of open science, and are members in various of research activities and organisations, including BIGMax, NOMAD, FAIRmat, DICE, EOSC-NORDIC, RELIANCE, EuroScienceGateway, The Carpentries, CodeRefinery and Software Sustainability Institute (SSI).

References

[1] ACM Software System Award. 2017. URL: https://awards.acm.org/software-system/award-winners?year=2017&award=149.

[2] E. Afgan et al. “The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update”. In: *Nucleic Acids Research* 46.W1 (May 2018), W537–W544. DOI: 10.1093/nar/gky379.

[3] M. Albert, H. Fangohr, and P. J. Metaxas. *Supplementary Data And Code For Paper: ”Frequency-Based Nanoparticle Sensing Over Large Field Ranges Using The Ferromagnetic Resonances Of A Magnetic Nanodisc”*. https://github.com/maxalbert/paper-supplement-nanoparticle-sensing. 2016. DOI: 10.5281/ZENODO.60605.

[4] M. Beg, J. Taka, T. Kluyver, A. Konovalov, M. Ragan-Kelley, N. M. Thiery, and H. Fangohr. “Using Jupyter for Reproducible Scientific Workflows”. In: *Computing in Science & Engineering* 23.2 (Mar. 2021), pp. 36–46. DOI: 10.1109/mcse.2021.3052101.

[5] Binder-supported Workshops. *Over 60 requests from organisers of workshops to increase mybinder.org resources temporarily for their workshop deliveries*. https://tinyurl.com/mr36bfzu. 2022.

[6] H. Fangohr et al. “Data Analysis Support in Karabo at European XFEL”. In: *Proc. of International Conference on Accelerator and Large Experimental Control Systems (ICALEPCS’17), Barcelona, Spain, 8-13 October 2017*. International Conference on Accelerator and Large Experimental Control Systems 16. https://doi.org/10.18429/JACoW-ICALEPCS2017-TUCPA01. Geneva, Switzerland: JACoW, 2018, pp. 245–252. DOI: https://doi.org/10.18429/JACoW-ICALEPCS2017-TUCPA01.

[7] H. Fangohr. *Introduction to Python for Computational Science and Engineering*. https://github.com/fangohr/introduction-to-python-for-computational-science-and-engineering/blob/master/Readme.md. 2022. DOI: 10.5281/ZENODO.5887400.

[8] H. Fangohr, V. Fauske, T. Kluyver, M. Albert, O. Laslett, D. Cortes-Ortuno, M. Beg, and M. Ragan-Kelly. *Testing with Jupyter notebooks: NoteBook VALidation (nbval) plug-in for pytest*. https://github.com/computationalmodelling/nbval. 2020. DOI: 10.48550/ARXIV.2001.04808.

[9] H. Fangohr, R. Rosca, and Y. Kirienko. *Open Science COVID Analysis (OSCOVIDA)*. 2022. URL: https://oscovida.github.io.

[10] H. Fangohr et al. "Data Exploration and Analysis with Jupyter Notebooks". en. In: *Proceedings of the 17th International Conference on Accelerator and Large Experimental Physics Control Systems* ICALEPCS2019 (2020), USA. DOI: 10.18429/JACOW-ICALEPCS2019-TUCPR02.

[11] J. Z. Forde, T. D. Head, C. Holdgraf, Y. Panda, G. Nalvarete, B. Ragan-Kelley, and E. Sundell. "Reproducible Research Environments with Repo2Docker". In: 2018. URL: https://api.semanticscholar.org/CorpusID:49255709.

[12] L. Gerhardt, W. Bhimji, S. Canon, M. Fasel, D. Jacobsen, M. Mustafa, J. Porter, and V. Tsulaia. "Shifter: Containers for HPC". In: *Journal of Physics: Conference Series* 898 (Oct. 2017), p. 082021. DOI: 10.1088/1742-6596/898/8/082021.

[13] B. E. Granger and F. Perez. "Jupyter: Thinking and Storytelling With Code and Data". In: *Computing in Science & Engineering* 23.2 (Mar. 2021), pp. 7–14. DOI: 10.1109/mcse.2021.3059263.

[14] Gravitational Wave Open Science Center. *Gravitational Wave Open Science Center*. URL: https://www.gw-openscience.org/about/.

[15] M. Hansen. *How We Got Published in The New York Times*. 2018. URL: https://journalism.columbia.edu/nyt-twitter-story.

[16] S. Hettrick. *Software In Research Survey*. 2018. DOI: 10.5281/ZENODO.1183562.

[17] T. Hirst. "The Rise of Transparent Data Journalism – The BuzzFeed Tennis Match Fixing Data Analysis Notebook". In: (2016). URL: https://blog.ouseful.info/2016/01/18/the-rise-of-transparent-data-journalism-the-buzzfeed-tennis-match-fixing-data-analysis-notebook/.

[18] P. Jupyter et al. "Binder 2.0 - Reproducible, interactive, sharable environments for science at scale". In: *Proceedings of the 17th Python in Science Conference*. Ed. by F. Akici, D. Lippa, D. Niederhut, and M. Pacer. 2018, pp. 113–120. DOI: 10.25080/Majora-4af1f417-011.

[19] T. Kluyver et al. "Jupyter Notebooks – a publishing format for reproducible computational workflows". In: *Stand Alone* 0.Positioning and Power in Academic Publishing: Players, Agents and Agendas (2016), pp. 87–90. DOI: 10.3233/978-1-61499-649-1-87.

[20] G. M. Kurtzer, V. Sochat, and M. W. Bauer. "Singularity: Scientific containers for mobility of compute". In: *PLOS ONE* 12.5 (May 2017), pp. 1–20. DOI: 10.1371/journal.pone.0177459.

[21] LA Times Data Desk Jupyter Notebooks. 2018. URL: https://github.com/datadesk/notebooks.

[22] P. Liu and J. Guitart. "Performance characterization of containerization for HPC workloads on InfiniBand clusters: an empirical study". In: *Cluster Computing* 25.2 (Nov. 2021), pp. 847–868. DOI: 10.1007/s10586-021-03460-8.

[23] G. Maze and K. Balem. "Argopy: A Python Library for Argo Ocean Data Analysis". In: *Journal of Open Source Software* 5.53 (Sept. 1, 2020), p. 2425. DOI: 10.21105/joss.02425.

[24] F. Mölder et al. "Sustainable data analysis with Snakemake". In: *F1000Research* 10 (Apr. 2021), p. 33. DOI: 10.12688/f1000research.29032.2.

[25] MyBinder team. *MyBinder.org Events Archive*. 2022. URL: https://archive.analytics.mybinder.org.

[26] T. E. Odaka, A. Banihirwe, G. Eynard-Bontemps, A. Ponte, G. Maze, K. Paul, J. Baker, and R. Abernathey. "The Pangeo Ecosystem: Interactive Computing Tools for the Geosciences: Benchmarking on HPC". In: *Tools and Techniques for High Performance Computing*. Ed. by G. Juckeland and S. Chandrasekaran. Vol. 1190. Communications in Computer and Information Science. Cham: Springer International Publishing, 2020, pp. 190–204. DOI: 10.1007/978-3-030-44728-1_12.

[27] V. D. Øvreeide and B. Ragan-Kelley. *Measuring notebook reproducibility with repo2docker*. JupyterCon, Oct. 2020. DOI: 10.6084/m9.figshare.19604239.v1.

[28] P. Parente. *Estimate of Public Jupyter Notebooks on GitHub: Latest Report*. 2022. URL: https://github.com/parente/nbestimate.

[29] F. Pérez and B. Granger. *Project Jupyter: Computational Narratives as the Engine of Collaborative Data Science*. 2015. URL: http://archive.ipython.org/JupyterGrantNarrative-2015.pdf.

[30] J. M. Perkel. "Why Jupyter is data scientists' computational notebook of choice". In: *Nature* 563.7729 (Oct. 2018), pp. 145–146. DOI: 10.1038/d41586-018-07196-1.

[31] R. Priedhorsky and T. Randles. "Charliecloud: Unprivileged Containers for User-Defined Software Stacks in HPC". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC '17. Denver, Colorado: Association for Computing Machinery, 2017. DOI: 10.1145/3126908.3126925.

[32] Project Jupyter. 2022. URL: http://jupyter.org.

[33] B. Ragan-Kelley, K. Kelley, and T. Kluyver. *JupyterHub: Deploying Jupyter Notebooks for students and researchers*. PyData London. Tutorial. 2016. URL: http://pydata.org/london2016/schedule/presentation/59/.

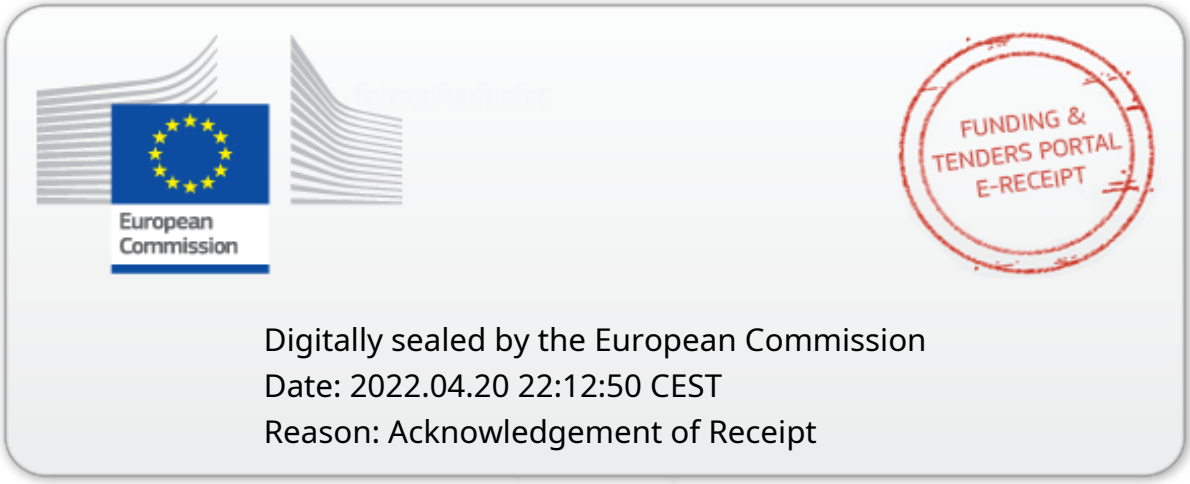
[34] Ø. Seland et al. "Overview of the Norwegian Earth System Model (NorESM2) and key climate response of CMIP6 DECK, historical, and scenario simulations". In: *Geoscientific Model Development* 13.12 (Dec. 2020), pp. 6165–6200. DOI: 10.5194/gmd-13-6165-2020.

[35] Simko, Tibor, Heinrich, Lukas, Hirvonsalo, Harri, Kousidis, Dinos, and Rodríguez, Diego. "REANA: A System for Reusable Research Data Analyses". In: *EPJ Web Conf.* 214 (2019), p. 06034. DOI: 10.1051/epjconf/201921406034.

[36] SOURCE team. *Example Repository for Reproducibility. Can be re-executed with Binder.* https://github.com/fangohr/reproducibility-repository-example. 2022.

[37] M. Woillez, R. Fablet, T.-T. Ngo, M. Lalire, P. Lazure, and H. de Pontual. "A HMM-based Model to Geolocate Pelagic Fish from High-Resolution Individual Temperature and Depth Histories: European Sea Bass as a Case Study". In: *Ecological Modelling* 321 (Feb. 2016), pp. 10–22. DOI: 10.1016/j.ecolmodel.2015.10.024.

[38] A. Zeller, R. Gopinath, M. Böhme, G. Fraser, and C. Holler. *The Fuzzing Book*. https://www.fuzzingbook.org. 2022.



This electronic receipt is a digitally signed version of the document submitted by your organisation. Both the content of the document and a set of metadata have been digitally sealed.

This digital signature mechanism, using a public-private key pair mechanism, uniquely binds this eReceipt to the modules of the Funding & Tenders Portal of the European Commission, to the transaction for which it was generated and ensures its full integrity. Therefore a complete digitally signed trail of the transaction is available both for your organisation and for the issuer of the eReceipt.

Any attempt to modify the content will lead to a break of the integrity of the electronic signature, which can be verified at any time by clicking on the eReceipt validation symbol.

More info about eReceipts can be found in the FAQ page of the Funding & Tenders Portal.

(https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/support/faq)