Jupyter4NFDI

A central JupyterHub providing access to various software stacks and computing resources across the NFDI consortia

Proposal for the Initialization Phase of Base4NFDI
Submitted: February 15th, 2024
On behalf of: WG Research Software Engineering, Section Common Infrastructures

1 General Information

- Name of proposed Basic Service (in English)

  A Central JupyterHub for the NFDI

- Acronym of the proposed Basic Service

  JUPYTER4NFDI

- Service “subtitle” explaining key functionality

  A central JupyterHub providing access to various software stacks and computing resources across the NFDI consortia

- Lead institution:

  Forschungszentrum Jülich GmbH, 52425 Jülich

- Name of lead institution principal investigator

  Björn Hagemeier <b.hagemeier@fz-juelich.de>

- Participating institutions

Principal Investigator	Institution, location	Contact E-mail	Member in [consortium]
Björn Hagemeier	Forschungszentrum Jülich GmbH, Jülich Supercomputing Centre (JSC)	b.hagemeier @fz-juelich.de	Text+
Arnim Bleier	GESIS – Leibniz-Institute for the Social Sciences[1]	arnim.bleier @gesis.org	NFDI4DS
Bernd Flemisch	University of Stuttgart (US)	bernd.flemisch @iws.uni-stuttgart.de	NFDI4Ing
Matthias Lieber	Technische Universität Dresden (TUD), Center for Information Services and High Performance Computing	matthias.lieber @tu-dresden.de	NFDI4DS
Klaus Reuter	Max Planck Computing and Data Facility (MPCDF)[1]	klaus.reuter @mpcdf.mpg.de	FAIRmat
George Dogaru	Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG)	george.dogaru @gwdg.de	Text+

Table 1: List of participating institutions

- Planned runtime of the project: **1 year**

- Summary of the proposal in English and German

English: The widespread usage of Jupyter notebooks across scientific disciplines highlights their fundamental importance in current research practices. However, their deployment across various NFDI consortia currently occurs through individual JupyterHubs, resulting in access barriers to computational and data resources. Only some of the services are widely available, and others are barricaded within VPNs, creating a fragmented landscape that complicates access. Our proposal aims to unify these efforts by offering a centralized service. We will

[1] The participating institution does not apply for funding in this proposal.

simplify access, significantly improve the user experience, and extend the reach of Jupyter to a broader audience within the NFDI and beyond. We will integrate with IAM4NFDI to govern access to the central service and external resources, with future integrations planned with upcoming basic services such as Multi-Cloud or nfdi.software. Moreover, HPC resources from the GCS and the NHR Alliance will be accessible via the centralized Jupyter service, which lowers HPC entry barriers for a wide range of users. Importantly, the centralized system will not only simplify access but also support the import of projects along with their necessary dependencies, fostering an ecosystem conducive to creating reproducible FAIR Digital Objects (FDOs), possibly along with notebook identifiers supported by PID4NFDI. By ensuring a consistent user experience across all consortia, we will promote continued innovation. Our commitment is to a collaborative and centralized approach that brings Jupyter notebooks within everyone's reach and democratizes access to the infrastructures for reproducible computational research within the NFDI and beyond.

German: Die weit verbreitete Nutzung von Jupyter-Notebooks in allen wissenschaftlichen Disziplinen unterstreicht ihre grundlegende Bedeutung in der aktuellen Forschungspraxis. Allerdings erfolgt ihr Einsatz in verschiedenen NFDI-Konsortien derzeit über einzelne JupyterHubs, was zu Zugangsbarrieren zu Rechen- und Datenressourcen führt. Nur einige der Dienste sind weithin verfügbar, andere sind innerhalb von VPNs verbarrikadiert, wodurch eine fragmentierte Landschaft entsteht, die den Zugang erschwert. Unser Vorhaben zielt auf eine Vereinheitlichung der Bemühungen ab, indem wir einen zentralisierten Dienst anbieten. Wir werden den Zugang vereinfachen, die Benutzerfreundlichkeit deutlich verbessern und die Reichweite von Jupyter auf ein breiteres Publikum innerhalb der NFDI und darüber hinaus ausdehnen. Wir werden IAM4NFDI integrieren, um den Zugriff auf den zentralen Dienst und externe Ressourcen zu regeln, wobei zukünftige Integrationen mit kommenden Basisdiensten wie Multi-Cloud oder nfdi.software geplant sind. Darüber hinaus werden HPC-Ressourcen des GCS und des NHR-Verbunds in den zentralen Jupyter-Dienst eingebunden, wodurch die HPC-Einstiegshürden für eine breite Nutzerschaft reduziert werden. Wichtig ist, dass das zentralisierte System nicht nur den Zugang vereinfacht, sondern auch den Import von Projekten mit ihren notwendigen Abhängigkeiten unterstützt und so ein Ökosystem fördert, das die Erstellung reproduzierbarer FAIR Digital Objects (FDOs) ermöglicht, möglicherweise zusammen mit Notebook-Identifikatoren, die von PID4NFDI unterstützt werden. Durch die Gewährleistung einer konsistenten Benutzererfahrung über alle Konsortien hinweg werden wir kontinuierliche Innovation fördern. Wir setzen uns für einen kollaborativen und zentralisierten Ansatz ein, der Jupyter-Notebooks zur rechnergestützten, reproduzierbaren Forschung für jedermann innerhalb der NFDI und darüber hinaus zugänglich macht.

2 State-of-the-Art of Proposed Basic Service

2.1 Background and Motivation

The use of Jupyter notebooks is ubiquitous in almost all fields of science today[2]. Jupyter notebooks provide the ability to create fully documented code and how it processes associated data. Sharing a notebook enables its recipients to understand the sender's code and intentions by allowing them to follow along the steps of data processing. The provision of an executable notebook can therefore be a major ingredient for ensuring the transparency of digital research processes and guaranteeing the reproducibility of computational results [6].

While it is possible to instantiate a Jupyter server and execute notebooks locally, this process might be cumbersome in the case of complex hardware and software requirements. Infrastructure providers can host so-called JupyterHub instances to give researchers access to computational environments and resources for executing notebooks without burdening them with installation and maintenance tasks. The necessity for and benefits of JupyterHubs have been recognized by several NFDI consortia which decided to connect their communities to already existing institutional services or to build up and offer access to new instances. A non-exhaustive list of available hubs in the context of the NFDI is collected at https://nfdi-jupyter.github.io/services/. As can be observed there, the listed hubs are very heterogeneous in various aspects such as access permission, computational resources or featured kernels. For a researcher in the NFDI, it might be impossible to find an accessible instance meeting their requirements.

With the proposed NFDI-wide Jupyter basic service, user access will be unified at a single point of entry. It will provide access to the same resources currently accessible through dispersed JupyterHub instances. A researcher trying to execute a Jupyter notebook through the central service will be directed to resources meeting their requirements and conforming to their respective access rights. Participating infrastructure providers will be able to connect their resources to the central service, enabling them to focus on their particular expertise and to operate their resources at an optimized capacity, while their researchers get access to the full spectrum of available software kernels and hardware architectures. By providing the technical means to execute virtually every notebook, their reproducibility will be largely supported. Existing JupyterHubs can optionally be maintained during the transition phase to the central service. Moreover, the basic service will exploit the fact that the central JupyterHub facilitates access to large-scale high-performance computing (HPC) systems.

More added value of the proposed Jupyter service will be substantiated by interactions with existing and prospective other basic services. This is obvious for IAM4NFDI and the provision of the NFDI Authentication and Authorisation Infrastructure, enabling a single sign-on experience and the management of resource allocations to users and user groups. Integration with the

[2] https://www.nature.com/articles/d41586-019-03366-x and https://www.nature.com/articles/d41586-018-07196-1

planned basic service nfdi.software and potentially any repository hosting Jupyter notebooks will automate the step from finding a notebook to its execution and possible adaptation. This step may be even more streamlined by employing notebook identifiers in the context of PID4NFDI. Moreover, it would be natural to utilize a possible Multi-Cloud service for identifying and connecting to remote resources.

The basic service will be made available at a well known address[3] with immediate login options at least for all interested NFDI consortia. Documentation and support information would be readily available including guides for onboarding new users.

2.2 State of the art

JupyterHub is a multi-user server for Jupyter notebooks, enabling concurrent work on Jupyter notebooks and other Jupyter-based documents. It's useful for organizations providing shared computing environments, allowing multiple users to access a Jupyter notebook server simultaneously. Each user has a personal workspace with notebooks, files, and settings, enabling easy collaboration and access to powerful computing resources. JupyterHub is highly customizable, integrating with authentication systems like OAuth, LDAP, and Kerberos for secure access. It supports various computing environments, including local servers, cloud platforms, and high-performance computing clusters.

Despite its utility, JupyterHub deployments have limitations. Users depend on administrators to configure the required computing environments and software libraries. This can create issues as the specific needs of users often vary based on their respective use cases. A statically configured JupyterHub deployment can limit usability and scalability, as any new requirements must be communicated to and implemented by the administrators. Even if administrators are willing to globally update the deployment, the changes can lead to dependency conflicts for other users.

BinderHub[4] offers a first step towards a solution to these limitations by expanding the conventional JupyterHub model. Rather than relying on privileged administrators for configuration, BinderHub allows its users to specify the configuration and dependencies. It enables a cloud-based environment for executing Jupyter notebooks packaged via repositories following the Reproducible Execution Environment Specification (REES)[5]. MyBinder[6], a public deployment of BinderHub, then supports the interactive execution of these environments with certain restrictions, such as being limited to public repositories and having moderate resource demands.

[3] For example https://jupyter.nfdi.de/ or https://nfdi-jupyter.de/
[4] https://binderhub.readthedocs.io/
[5] https://repo2docker.readthedocs.io/en/latest/specification.html
[6] https://mybinder.org/

2.3 Own Preparatory Work for the Basic Service

In the following, we list three measures of preliminary work for the basic service: JupyterHub instances and services as part of institutional and consortial efforts, a spawner for the execution of various backend services, and activities within the Jupyter ecosystem.

2.3.1 JupyterHub instances and services

JSC's JupyterHub service[7] has been in operation for several years and provides access to various backends. Users can run JupyterLabs on five HPC systems' login or compute nodes [1]. Executing on compute nodes deducts computing time from the user's HPC project budget. Alternatively, JupyterLabs can be run in a Kubernetes environment with limited resources, enabling access without a full HPC account for users within the Helmholtz Association. The service caters to 150-250 unique users weekly, utilizing Cloud and supercomputing resources according to their requirements.

GESIS Notebooks enables reproducible analysis in Social Sciences. The service supports researchers, enhances visibility, and promotes open-science practices. It is updated and extended as part of NFDI4DataScience. The service handles about 40,000 sessions per week, each with 1 CPU and 4 GB of memory. However, for complex data science projects, more resources are needed. To address this[8], GESIS developed Persistent BinderHub[9], combining features from BinderHub and JupyterHub. It is already adopted by several institutions and showcased at relevant conferences[10].

The MPCDF offers Jupyter services to Max Planck Society scientists and collaborators, including options to run Jupyter notebooks on Slurm-based HPC clusters, supercomputers, as well as on Cloud environments based on BinderHub deployments[11] managed by Kubernetes. Domain scientists routinely use the platform for workshops, interactive material for publications, and easy access to new computational methods.

The Jupyter service at TUD offers JupyterLab on dedicated HPC resources (compute nodes with CPU/GPU) using a Slurm integration[12]. The service is only available for registered HPC users. It is used frequently for training and workshops but also as a low entry barrier interface for HPC, especially for the data science community. For the proposed service, TUD will make its NHR compute resources available for the NFDI community.

The main JupyterHub provided by GWDG is accessible via AcademicID, one of the Community AAIs to be part of the IAM4NFDI infrastructure, and can already be used by Text+ members. Several Docker images are available when starting the JupyterLab environment and it is possible to make further images with preinstalled software available. Notebook-based solutions

[7] https://jupyter-jsc.fz-juelich.de/

[8] The need to resolve these limitations has also been substantiated via a user survey https://t.ly/Z-NFm. A preliminary analysis of the open-ended responses to the survey is available in the binder-ready repository https://github.com/sgibson91/mybinder.org-user-survey-nlp .

[9] https://github.com/gesiscss/persistent_binderhub

[10] Talk at JupyterCon 2020, Persistent BinderHub: https://t.ly/rUaY1

[11] https://notebooks.mpcdf.mpg.de/binder/

[12] https://jupyterhub.hpc.tu-dresden.de/

suitable for tasks in Text+ are being developed within the project's "Infrastructure/Operations" task area, in which GWDG and JSC are involved.

Several other institutions and NFDI consortia have established services from the Jupyter ecosystem and were actively involved in the discussions leading up to this proposal. In a joint effort, we have started a collection of available JupyterHub instances[13] within the consortia. The resources used for these existing, so far unrelated services, can potentially be used as external resources for the proposed basic service.

2.3.2 OutpostSpawner

JupyterHub manages JupyterLabs and other backend services through so-called spawners. In the trivial case, this translates into the local execution of a process running the JupyterLab. In distributed or HPC environments, the execution of such backend services becomes more complex and specific spawners need to be developed to support a variety of environments. The original idea of the JupyterHub developers is to have a spawner implementation per backend type. However, only a single spawner can be configured within each JupyterHub instance, in turn allowing only a single backend type per instance. In order to overcome this limitation, the OutpostSpawner[14] has been developed at JSC. Its predecessor has been in operation for over five years and facilitates access to JupyterLabs based on Kubernetes or HPC. The latest version of it enables the use of every JupyterHub spawner, which follows the JupyterHub standards, at connected JupyterHub Outposts[15]. The communication between User, JupyterHub and notebook server is managed by OutpostSpawner and JupyterHub Outpost, enabling an easy integration of external systems to the central JupyterHub.

2.3.3 Activities within the Jupyter ecosystem

Some of the participating institutions of this proposal are dedicated to advancing the Jupyter ecosystem. For example, GESIS has been an active participant in the mybinder.org federation[16] and contributed regularly to the development of the BinderHub software. Furthermore, GESIS developed the Persistent BinderHub software. In a crucial step towards ensuring long-term sustainability, the stewardship of Persistent BinderHub's development has been transferred to core developers within the Jupyter ecosystem to enable a JupyterHub-Binder Integration, in collaboration with 2I2c[17]. This collaboration promises continued evolution and robust support for the platform, leveraging the broad expertise and community support within the Jupyter ecosystem.

2.4 Current Technical Readiness Level of the proposed Basic Service

The following table lists the essential components that will be used to provide the basic service. Almost all of them are in use in a production environment at JSC, serving numerous users as

---

[13] https://nfdi-jupyter.github.io/services/
[14] https://github.com/kreuzert/jupyterhub-outpostspawner
[15] https://github.com/kreuzert/jupyterhub-outpost
[16] https://mybinder.readthedocs.io/en/latest/about/team.html#partner-institutions
[17] https://2i2c.org/blog/2024/jupyterhub-binderhub-gesis/

described above. We consider industry standard solutions with a large user base and large number of deployments to be TRL 9. The components we plan to use as the foundation for the basic service have been in use for several years now, but have a smaller user base.

Component	TRL	Description
OpenStack, Kubernetes, Rancher, Fleet, JupyterHub	9	External developments
UNICORE	8	Development for over 25 years at JSC and external partners
JuypterHub @JSC and OutpostSpawner	7	JupyterHub instance and custom Spawner (in operation since 2017)
JupyterHub Outpost	7	RestAPI to manage notebook-server and other web services for every JupyterHub spawner (in operation since 2017 and 2021 respectively with a different name)
JupyterHub-Binder Integration	6	This integration allows reproducibility and the import of binder-ready repositories from platforms such as GitHub
IAM4NFDI	5	External development, average of components as indicated by the IAM4NFDI proposal with technical solutions already at TRL 9

Table 2: Technology Readiness Levels

3 SWOT Analysis

Internal	Strengths 1. State of the art standards and software stacks (OpenStack, Kubernetes, Django, continuous delivery) 2. Over six years of operational experience	Weaknesses 1. Single point of failure 2. Centralization at a single site 3. Lack of testing by a wider community
External	Opportunities 1. Single landing page for all of NFDI 2. Access to various systems at multiple sites 3. Low entry barrier for additional resources 4. Increase number and amount of available total resources 5. Little requirements from user’s point of view 6. Gather scattered Jupyter community within NFDI	Threats 1. Access via username/password 2. Disparity of integrated resources may be challenging 3. Complex full stack HPC setup 4. Technological lock in 5. Lack of adoption by consortia

Table 3: SWOT Analysis

In the following paragraphs, we discuss some of the weaknesses and threats including potential countermeasures.

We acknowledge that the central service will be a single point of failure. However, in the current setup of multiple JupyterHubs per consortium there is a single point of failure in each setup. The difference is when users would be affected by potential outages, not necessarily in the duration of this outage. We believe that a single, central service can be offered at higher individual availability than many unrelated services, because of reduced overall effort in its operation. Potentially being able to switch from using one service to another may not help during outages, because the services may differ in essential aspects, such as resource types and permissions, availability of data or permissions to access the front end. The central approach will actually provide an opportunity to aim for a highly available system crossing site boundaries, an effort that can hardly be handled by many concurring unconnected instances.

The service will be hosted centrally at a single site. Initially, this will be JSC, because the JupyterHub service that has been running there for several years serves as a template for the NFDI base service. The potential risk of a partner lock-in will be countered by ensuring an open and generic development of the infrastructure. This will allow for other partners to take over the central role should this ever be required. Moreover, such possibility may pave the way for a true high-availability (HA) setup in the future.

Accessing computational resources via username and password alone has been largely rejected by supercomputing sites in the past. In recent years, however, some of the restrictions have been lifted in order to allow for service offerings such as Jupyter. This may pose a risk if users employ weak passwords. The addition of multi-factor authentication (MFA) will mitigate this risk and preparatory work has already been conducted. Currently, Jupyter users at JSC can opt-in to MFA.

The disparity of integrated external resources is an opportunity and a threat at the same time. Bringing in a larger variety of resources offers the opportunity for users to access the resources most appropriate for their use cases. On the other hand, the integration itself and the variation among the resources introduces additional complexity. Capturing the variety of resources (HPC and cloud) in the user interface has already been solved at JSC. We will tackle the additional complexity introduced by accessing external resources at various sites by providing a well-defined API for such integration, that is also already in use at JSC.

There is a risk of a lack of adoption by the consortia. By the time we will be ready to include any resources they may already use within their consortium, they may have become accustomed to using their existing solutions. Migrating a large user base from one service to another without incentivizing such migration has proven to be a challenging task in the past. Related to this, the timelines of existing consortia, which are executed as projects, may not be in line with the timeline of this basic service. We believe that the existing JupyterHub services can remain connected to their resources during the initial phase and users migrated to the central service subsequently, thus offering a smooth transition to the central service. From the point of view of the resource providers, an overall reduced maintenance effort can be expected. Users can expect an increased quality of service from a frequently used, centrally managed service.

4 Working Concept for the development of the Basic Service

The concept of the central service is to provide a single JupyterHub entry point into a distributed infrastructure of disparate resources. The central JupyterHub will initially co-exist with any previously existing or otherwise required community-specific JupyterHubs. The ultimate goal is to centrally provide a compelling user experience while still offering the accustomed versatility of the participating resources.

A central JupyterHub reaching out to various external resources, cloud-based and HPC, exists at JSC and has been in operation since 2017. It has been integrated with an OIDC-based authentication solution and maps users of the web interface to their UNIX user names in the HPC area. This will be the starting point of the development of the basic service. We will tailor the already existing JupyterHub to the needs of an NFDI central offering. Such customization has already proven to be a successful approach, when enabling access to the Jülich Quantum Computing Infrastructure (JUNIQ)[18], which has been implemented as another endpoint in the same underlying infrastructure. By this approach we avoid duplication of essential components.

In addition to the central hub, which is a technical requirement for the central service, we aim to provide the following features of interest to many current user communities. A connection to existing data and software repositories, conforming to the REES specification, will allow users to immediately start working with their relevant data. The JupyterHub-Binder Integration will allow users to share Jupyter notebooks and import notebooks with their needed dependencies. In support of making scientific results reproducible, we consider participating in the MyBinder federation. FAIR digital objects (FAIR DO) should become importable as easily as possible.

Finally, because the Jupyter ecosystem has become a widely used, invaluable tool to modern science, we will liaise with the wider Jupyter community and feed back any relevant developments upstream to ensure the sustainability of our solutions.

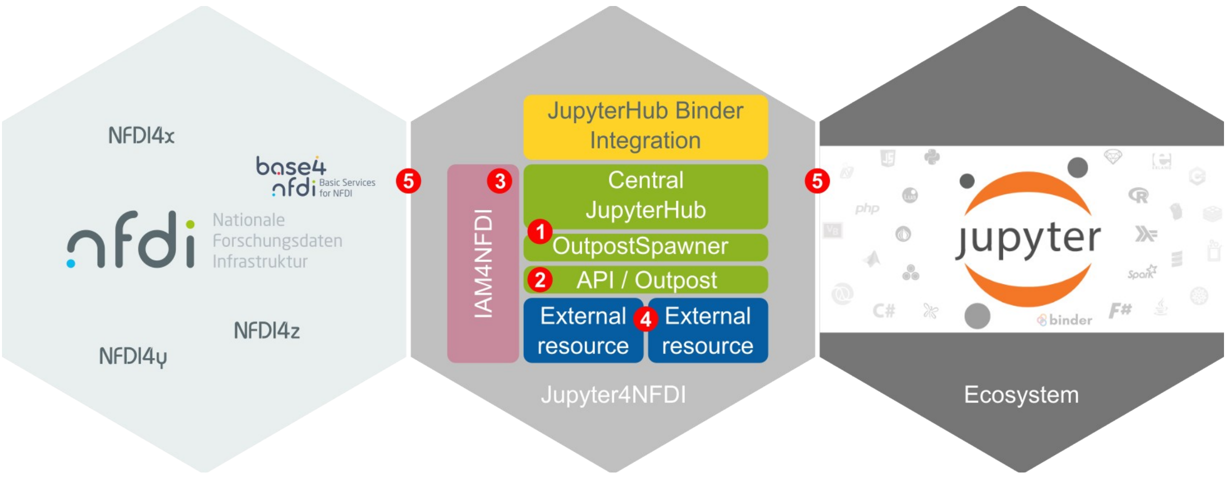


Figure 1: The central service deployed by Jupyter4NFDI comprising the central JupyterHub, versatile OutpostSpawner and API towards external resources. It will be using the IAM as provided by IAM4NFDI as well as external resources provided by various consortia and potentially the upcoming Multi-Cloud. We intend to integrate with other Base4NFDI services as well as the overall Jupyter ecosystem. Work package numbers in red.

[18] https://juniq.fz-juelich.de/

4.1 Service initialisation concept

Initially, the central service will be set up and hosted at JSC. However, we will ensure that the service setup is sufficiently generic to possibly migrate to another central hosting provider in the future. This will be achieved by using widely used standard software stacks and standard deployment methods for the service. Early on, the central JupyterHub will be complemented by connecting two external resources, a cloud based Kubernetes cluster and UNICORE based HPC systems, followed by external resources hosted at the institutions of the further PIs. The entire deployment will be sufficiently documented to transfer the service to alternative or even additional providers in the future.

Initial measures have already been taken to avoid a lock-in to any particular provider to host the central service. In particular, the currently envisioned solution will not require any local modifications of the mainline JupyterHub code, but only use appropriate configuration. Moreover, all developments will be openly available, as is already the case for the installation at JSC, as detailed in the section about WP1.

The Jupyter4NFDI basic service will be able to include external resources. We aim to use the IAM4NFDI basic service to authenticate and authorize users at both the JupyterHub as well as at the external resources. This will allow for a consistent user experience and avoid additional measures to propagate identity information or the storage of user credentials in the central service. We will start with a small number of external resources during the initialization phase to ensure proper documentation that will serve as a guide to the integration of comparable additional resources. To this end, the integration will be based on a concise API that is already in use at JSC and can easily be implemented by any provider of external resources. The existing implementations allow to access any type of cluster as external resources, if a corresponding spawner exists.

The Multi-Cloud concept [2] clearly states the idea of providing access to cloud resources through Jupyter. A central cloud offering within the NFDI opens up two opportunities of integrating such resources. For one, the above stated idea could be implemented by integrating resources of the Multi-Cloud as external resources in the basic service once they become available. The other opportunity would be to use the Multi-Cloud as a hosting environment for the central service. The timeline for the availability of the NFDI Multi-Cloud environment is yet unclear. We will monitor the developments in this working group and intend to evaluate a potential integration of resources of the Multi-Cloud. To this end, we are in contact with TA4 of PUNCH4NFDI to establish synergies.

We will make every effort to ensure the sustainability of our solution. The measures we will take in this regard will be:

1. Staying in contact with the broader Jupyter community by participating in workshops and regular meetings, taking these opportunities to ensure that our solutions are compatible

with the wider Jupyter ecosystem. This activity will be coordinated in WP5 and implemented in the respective technical WPs 1 and 2.

2. Make as much use as possible of available generic facilities for customization rather than modifying code of the basic service, thus avoiding the creation of a fork that would be difficult to maintain.
3. Feeding back any unavoidable code changes upstream to make them available to the broader Jupyter community.
4. The information and continuous involvement of providers of external resources, i.e. providers of Kubernetes environments or HPC centres.

4.2 Development and integration outlook

The goal of the subsequent integration phase will be to stabilize the environment and include more resources from external providers. Such external providers may be those that already offer Jupyter services within their consortia and would now like to put their offering on a more generic and sustainable foundation, while still accommodating specific requirements of their respective community. This will also be a means to support additional communities with similar specific requirements that may not yet have been served elsewhere, e.g. in terms of hardware accelerators or other specific infrastructure elements. This can be particularly useful for resources supporting artificial intelligence and machine learning, which is relevant in most fields of science nowadays. Moreover, locally available data at external resources may be relevant for several communities and providing access to the data from a central hub facilitates and enhances their common usage.

Finally, the capacity to import code along the necessary dependencies that adhere to the REES specification paves the way for a host of opportunities. Specifically, Fair Digital Objects (FDOs) hosted in open-access repositories can be seamlessly imported and executed with just a single click. This is strongly connected to a possible collaboration with PID4NFDI regarding notebook identifiers. As a result, the basic service initiative nfdi.software can function as an effective showcase for reproducible software available to our proposed infrastructure.

4.3 Ramping up for operation

Once consortia have migrated to the basic service, a business model will have to ensure the service’s sustainability. Funding will have to be secured to cover the cost of operating the service including the external resource integration. Governance among the key stakeholders, consortia, resource and service providers, will have to be established to ensure effective coordination, decision-making, and resource management across the various entities. Furthermore, the efficient operation and evolution of the service will be a topic.

On the technical side, the central service will be professionalized by implementing easy to reach support channels, an integration of further external compute and data resources including specific recipes for recurring tasks. Support would be organized through a distributed team involving representatives of the central JupyterHub, external resources as well as communities.

4.4 Risks and challenges

Risks and challenges have been discussed as part of the SWOT analysis in section 3.

5 Work Programme

With this proposal, we apply for an initialization phase of a duration of one year. During this phase, we will deploy a central JupyterHub instance with a basic attached resource pool and create proper documentation to guide providers of additional external resources to join the central service. Furthermore, the central JupyterHub service will be integrated with the IAM4NFDI central service to support the management of users and access permissions. This integration comprises authentication and authorization at the central service as well as the external resources, which in turn should also be integrated with the IAM4NFDI service.

The connection to NFDI and other data and software repositories will be prepared. The full implementation of such connections will be done as part of a possible subsequent integration phase.

5.1 Overview of work packages

There are two essential work packages that directly address the Jupyter basic service on the technical level. In WP1, Deployment of central JupyterHub, we aim to provide the central infrastructure by installing a central JupyterHub instance. In WP2, Resource Manager, we will deal with managing external resources to run user workloads such as Jupyter Labs. In WP3, Integration of the AAI, we will keep close contact to the IAM4NFDI[19] basic service partners to integrate both the central JupyterHub as well as the external resources (if not already the case) with the central identity and access management. All developments in WPs 1-3 will be evaluated and improved by connecting first external resources from participating partners by means of WP4. Finally, we will guarantee the sustainability of our solutions and their embedding into the wider Jupyter community by making them available beyond the scope of the NFDI in WP5, Ecosystem Integration, and also conceptualize the connection to data and software repositories.

[19] https://www.nfdi.de/identity-und-access-management-startet-weg-zu-nfdi-weitem-basisdienst/

Work package	Deliverables (D) and milestones (M)	Responsible partner
1. Deployment of central JupyterHub	M1.1 Central JupyterHub deployed D1.1 REST API documentation for resource manager	JSC
2. Resource Manager	M2.1 JSC resources available in central hub D2.1 External resource API implementation template	JSC
3. Integration of the AAI	M3.1 Integration with IAM infrastructure proxy D3.1 Profile for the integration with IAM infrastructure proxy	GWDG
4. Connecting external resources	M4.1 Initial external resources available in central hub	US
5. Ecosystem Integration	D5.1 Use-case for binder based FDOs D5.2 Development of user stories D5.3 Best practice guidelines for reproducibility	GESIS[20]

Table 4: Overall work programme with work packages, deliverables, milestones and responsible partners

The timeline and dependencies of the deliverables and milestones mentioned in Table 4 are available in the Gantt chart in the appendix.

5.2 Detailed work programme

5.2.1 WP1: Deployment of central JupyterHub

WP lead: JSC

Contributors: JSC, US

This work package will provide a central JupyterHub web service (M1.1) to provide users access to external resources. In this context, external refers to all resources made accessible regardless of their location and type.

To provide this central service, we will set up a Kubernetes cluster with Rancher on top of an OpenStack cloud environment. Rancher's Fleet[21] service will be a central component of this cluster, deploying necessary software packages using a GitOps[22] approach. The software packages that will be deployed include the JupyterHub instance, which uses the custom OutpostSpawner, and common core services such as database, proxy, persistent storage, and monitoring.

JupyterHub uses spawners[23] to start web services (such as single-user notebook servers) for a user. Spawners represent an abstract interface to a process. A custom spawner needs to be able to take three actions: start, stop and poll a process. We will employ the above mentioned OutpostSpawner to control JupyterLabs on remote resources (HPC, Cloud, etc.), which will be equipped with a JupyterHub Outpost accepting forwarded commands from the OutpostSpawner. The JupyterHub Outpost provides a REST API (D1.1) that also provides start, stop and poll functions.

[20] Funding for the deliverables of GESIS D5.1-3 stems from NFDI4DS - NFDI for Data Science and Artificial Intelligence deliverable D3.6.4.1 (DFG project number 460234259).
[21] https://fleet.rancher.io/
[22] https://github.com/FZJ-JSC/jupyter-jsc-deployment/
[23] https://jupyterhub.readthedocs.io/en/stable/reference/spawners.html

In this work package we will define and document the required configuration of a JupyterHub Outpost, to use external resources of a given type. Two reference JupyterHub Outposts implementations will be provided, using the KubeSpawner[24] and the UnicoreSpawner[25]. For this work package, a JupyterHub Outpost using the KubeSpawner will be installed which makes the local resources of the cluster it is installed on available to users.

The identification and implementation of custom features required by third parties (NDFI consortia, communities, etc.) will be envisaged, but likely be realized only in subsequent project phases.

During the initialization phase, JSC will provide a general resource pool comprising 100 CPUs, 750GB main memory and 100TB of storage. This will allow for about 200 simultaneous JupyterLab sessions, e.g. when conducting training courses via the basic service. Users can apply for additional resources, e.g. in the HPC area, and access them through the same basic service.

5.2.2 WP2: Resource Manager

WP lead: JSC

Contributors: JSC, MPCDF, TUD

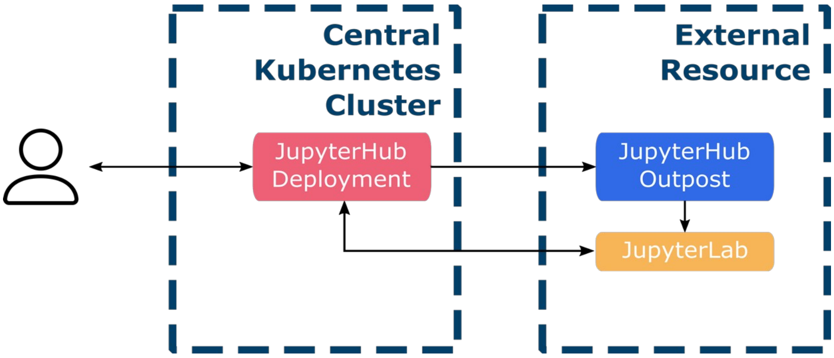


Figure 2: Interaction between JupyterHub and JupyterHub Outpost

JupyterHub Outposts are responsible for the life cycle of the user's JupyterLabs on a specific resource. While the specific implementation may differ for each resource, all JupyterHub Outposts offer a unified REST API for the start, stop and poll functions. Therefore, it can be understood as a skeleton implementation for all JupyterHub spawners. Two reference configurations, using the KubeSpawner and the UnicoreSpawner, provide a starting point for installations and configurations of JupyterHub Outposts. While the front-facing REST API of a JupyterHub Outpost strictly follows the necessary specifications, the backend is fully configurable and customizable. D2.1 will provide an improved version of the existing documentation and a template for configuring the JupyterHub Outpost for additional external resources, which will be created in an iterative manner with connecting first instances in WP4.

[24] https://github.com/jupyterhub/kubespawner
[25] https://github.com/kreuzert/jupyterhub-unicorespawner

Each JupyterHub Outpost should offer persistent storage for the users. The realization of storage availability and its user-specific quotas is the responsibility of each site. Additional features may be offered in consultation with the central JupyterHub, using the close collaboration between OutpostSpawner and JupyterHub Outpost. These could be the import of existing projects (JupyterHub-Binder Integration) or Fair Digital Objects (FDO, cf. WP5). Moreover, we envisage a variety of versions of JupyterLabs and pre-installed Kernels as well as specific JupyterLab extensions.

When receiving a request to perform an action for a user, such as starting a JupyterLab, JupyterHub Outpost may receive an access token. The access token will then be used to validate the information given to it by the central JupyterHub with the AAI service to authorize the user. Information about actual access permissions to external resources will be exchanged prior to the user accessing the infrastructure, such that it is ensured that users will only be able to attempt running JupyterLabs on resources where they have access. How this information will be exchanged will be a topic of WP3.

The JupyterHub Outpost allows a fine-grained offering of all connected resources. It is possible to support multiple JupyterHubs, each with different flavors. Administrators can ensure an optimized use of all resources. Various limitations per JupyterHub, user or virtual organization can be configured.[26]

5.2.3 WP3: Integration of an AAI

WP lead: GWDG

Contributors: GWDG, JSC

The basic service IAM4NFDI, which started its implementation in April 2023, will provide consistent user IDs across all NFDI consortia and participating organizations. It will be the single identity and access management service within NFDI. After preliminary discussions with key personnel of the project we identified the *IAM infrastructure proxy* as the primary component to connect to with the central Jupyter service. Furthermore, the IAM4NFDI WP3 “Incubator” allows for working on specific advanced requests from the community and is structured in 4 cycles of 6 months each. The integration of Jupyter services can be addressed in IAM4NFDI M3.3 “Incubator Cycle 2”.

The integration with IAM4NFDI does not stop at the central JupyterHub, but should also involve the external resources that run the actual workloads. While a similar setup is already working in production at JSC, the actual integration with a new identity and access management will be quite involved.

Deliverable 3.1 will provide a profile for the integration with the IAM infrastructure. In particular, it will document the required and available attributes from the central IAM service. Moreover, it will specify the interplay of IAM, JupyterHub and external resources. There are several options

[26] https://jupyterhub-outpost.readthedocs.io/en/latest/usage/configuration.html

how this could be done and they may be influenced by the implementation of the IAM basic service. An initial implementation of this specification will be available as M3.1, with optional improvements and optimizations subject to a subsequent project phase.

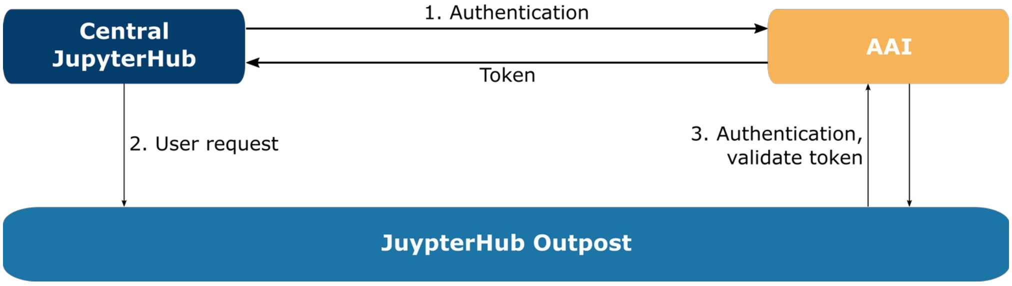


Figure 3: User authentication with AAI

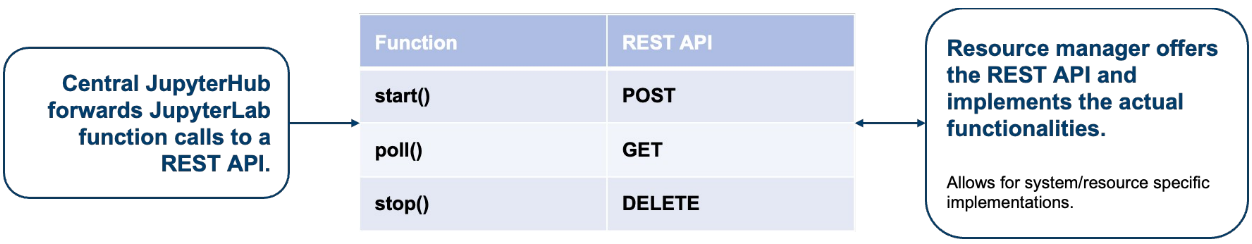
5.2.4 WP4: Connecting external resources

WP lead: US

Contributors: MPCDF, TUD, GWDG

For evaluating and optimizing our efforts, it is important to test the integration of external resources early on. The partnering groups will install a JupyterHub Outpost locally. This can be based on one of the two reference configurations, or a custom configuration according to the local requirements at each site. Specifically, all partners will connect their cloud or HPC resources to the central hub, with their explicit contributions listed in the appendix. With TUD and GWDG, two NHR centres will make their HPC resources accessible via the central JupyterHub. We see this integration as a strategic step to promote NHR and to lower the HPC entry barriers for a wide range of user groups. As indicated in the attached Letter of Support by the NHR association, a corresponding information exchange with other NHR centres is planned.

A prototypical configuration is sufficient to test the integration. The completed backend logic would be implemented in a subsequent project phase. The location of the JupyterHub Outpost installation can be freely chosen by the individual sites. The API and other relevant services must be available to the central JupyterHub.



Function	REST API
start()	POST
poll()	GET
stop()	DELETE

Figure 4: Expected REST API JupyterHub Outpost implementation

The experiences gained thereby will help to refine the JupyterHub Outpost configuration template as delivered by WP2. Connecting external resources is also an important step for validating the integration of the AAI mechanism developed in WP3. The key result of this workpackage will be available with M4.1 at the end of the initialization phase.

5.2.5 WP5: Organization & Ecosystem Integration

WP lead: GESIS

Contributors: JSC, US

The Jupyter ecosystem, including tools and frameworks, has become a cornerstone in modern scientific computing and data analysis across almost all academic fields. To foster and sustain this utility and versatility, the Jupyter ecosystem aligns to reusable standards and adheres to open-source software development practices. Similar to the Jupyter community as well as across research domains, the NFDI-consortia researchers and service managers require services that will facilitate quality and long-term maintainability amidst limited resources.

In this necessarily cross-sectional project, we will build on the practices and standards that have enabled the success and scalability of the Jupyter ecosystem and extend them when necessary to facilitate collaboration among the consortia. To this end, we plan three forms of outreach and integrations based on the practices and standards of successful open source software development:

- Engagement with the international Jupyter community: We have already taken the initial steps in this direction by sharing our proposed work at the monthly JupyterHub team meeting https://github.com/jupyterhub/team-compass/issues/644.
- Collaboration with service managers of related services: here, our focus will be on the planned software galleries and marketplaces across the consortia. We are also planning to integrate with the basic service nfdi.software, which will allow for binder-ready import of FAIR Digital Objects (FDOs) into the JupyterHub.
- Involvement of end-users and their requirements: We plan to take a two-tier approach. First, we will develop tutorials and guidelines for best practices, accompanied by examples to demonstrate the computational aspects of research. Second, we will gather feedback through monthly open team meetings to establish an easily accessible point of contact for users and developers. These meetings will be coordinated using a discussion forum modeled after the JupyterHub Team Compass.[27].

These forms of outreach will help us to identify and communicate the necessary standards and interfaces to develop a successful Jupyter service offering for the entire NFDI.
The three forms of outreach and integrations guide and inform our deliverables. Specifically, in D5.1, we will investigate the use-case for Binder-based FDOs. D5.2 will cover the development of corresponding user stories. Finally, these activities will culminate in D5.3, which will provide best practice guidelines to enable the envisioned adoption of Jupyter-based computational reproducibility across the NFDI ecosystem.

[27]https://github.com/jupyterhub/team-compass

8 Appendix

8.1 Bibliography and list of references

[1] J. H. Göbbert, T. Kreuzer, A. Grosch, A. Lintermann, and M. Riedel (2018). "Enabling Interactive Supercomputing at JSC Lessons Learned," in Lecture Notes in Computer Science, Springer International Publishing,, pp. 669–677.

[2] Marius Alfred Dieckmann, Alexander Goesmann, Kilian Schwarz, Christoph Wissing, Alexander Sczyrba, & Sebastian Jünemann (2022). NFDI Multi-Cloud concept (1.0.1). Zenodo. https://doi.org/10.5281/zenodo.6510971

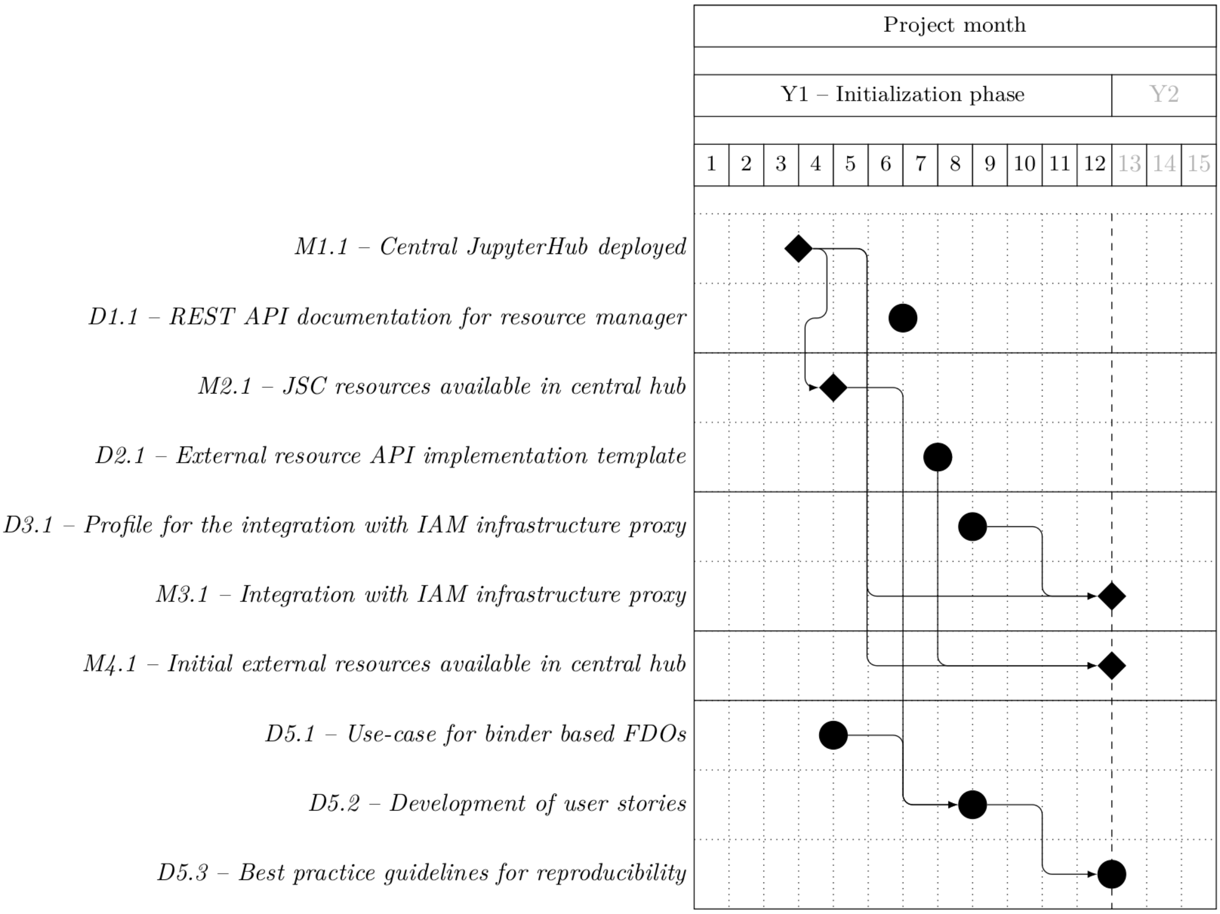
[3] Tim Kreuzer, & Alice Aiko Grosch (2023). kreuzert/jupyterhub-outpost: 0.9.4 (0.9.4). Zenodo. https://doi.org/10.5281/zenodo.10577756

[4] Tim Kreuzer, & Alice Aiko Grosch (2023). kreuzert/jupyterhub-outpostspawner: 0.0.23 (0.0.23). Zenodo. https://doi.org/10.5281/zenodo.10577760

[5] Tim Kreuzer, & Alice Aiko Grosch (2023). kreuzert/jupyterhub-unicorespawner: 0.9.0 (0.9.0). Zenodo. https://doi.org/10.5281/zenodo.10577763

[6] David Schoch, Chan Chung-hong, Claudia Wagner, and Arnim Bleier (2023). "Computational Reproducibility in Computational Social Science." arXiv preprint arXiv:2307.01918

8.2 Gantt Chart



8.3 Resource contributions

Several partners are dedicated to contribute an indicative amount of shared resources to the initial infrastructure as listed in the following table.

Provider	CPU cores	RAM (GB)	Storage volume (TB)
JSC	100	750	100
MPCDF	32	128	10
US	84	1536	22
GWDG	100	750	100
T**UD**	64	256	32

In addition, users with access to HPC resources (CPU or GPU) at participating GCS or NHR centres (JSC, GWGD, TUD) will be able to access these through the central Jupyter service.