Project Title	Hybrid Eco Responsible Optimized European Solution
Project Acronym	HEROES
Grant Agreement No.	956874
Start Date of Project	01.03.2021
Duration of Project	24 Months
Project Website	heroes-project.eu

D2.3 – Workflow Interfaces

Work Package	**WP 2, AI and HPC Workflows**
Lead Author (Org)	**Jorik Remy (UCit)**
Contributing Author(s) (Org)	**Elisabeth Ortega (HPCNow!), Benjamin Depardon (UCit)**
Reviewed by	**Elisabeth Ortega (HPCNow!), Davide Pastorino (Do IT Systems)**
Approved by	**Benjamin Depardon (UCit) – Management Board**
Due Date	**30.10.2022**
Date	**22.12.2022**
Version	**V1.1**

Dissemination Level

X	PU: Public
	PP: Restricted to other programme participants (including the Commission)
	RE: Restricted to a group specified by the consortium (including the Commission)
	CO: Confidential, only for members of the consortium (including the Commission)

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

Versioning and contribution history

Version	Date	Author	Notes
0.1	06.09.2022	Jorik REMY (UCit)	TOC and V0.1
0.2	14.09.2022	Jorik REMY (UCit)	Draft
0.3	15.09.2022	Jorik REMY (UCit)	Executive Summary, Introduction
0.4	16.09.2022	Jorik REMY (UCit)	
0.5	24.10.2022	Benjamin Depardon (UCit)	
0.6	26.10.2022	Benjamin Depardon (UCit)	Formatting, sections 2 & 3
0.7	29.11.2022	Elisabeth Ortega (HPCNow!)	First revision
	14.12.2022	Elisabeth Ortega (HPCNow!)	Review
	14.12.2022	Davide Pastorino (Do IT Systems)	Review
1.0	14.12.2022	Benjamin Depardon (UCit)	Final version for review
1.1	22.12.2022	Corentin Lefevre (Neovia)	Version approved by the Management Board

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

Table of Contents

List of Figures

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

Terminology

Terminology/Acronym	Description
API	Application Programming Interface
CI/CD	Continuous Integration/Continuous Delivery
CLI	Command Line Interface
CORS	Cross-Origin Resource Sharing
CPU	Central Processing Unit
HPC	High-Performance Computing
HTTPS	HyperText Transfer Protocol Secure
OOD	Open OnDemand
RAM	Random Access Memory
REST	REpresentational State Transfer
UI	User Interface
VM	Virtual Machine
VNC	Virtual Network Computing

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

References

Note: all URLs cited here have been verified in November 2022.

[1] D3.1 – Architecture Design

[2] D3.3 – Initial Platform Mock-up

[3] NextFlow Feature request: Add a web UI that allows a user to launch and monitor a workflow execution #88 – https://github.com/nextflow-io/nextflow/issues/88

[4] DolphinNext: A platform to create reproducible, portable and highly parallel pipelines – https://github.com/UMMS-Biocore/dolphinnext

[5] Hudak et al., (2018). Open OnDemand: A web-based client portal for HPC centers. Journal of Open Source Software, 3(25), 622, https://doi.org/10.21105/joss.00622

[6] Open OnDemand website – https://openondemand.org/

[7] NICE EnginFrame - https://aws.amazon.com/hpc/enginframe/

[8] Altair® Access™ - https://www.altair.com/access

[9] IBM Spectrum LSF Application Center - https://www.ibm.com/docs/en/slac/10.1.0?topic=center-introduction

[10] Bull extreme computing studio - https://atos.net/wp-content/uploads/2021/01/Factsheet_extreme_compting_studio_HPC.pdf

[11] Calegari, P., Levrier, M., & Balczyński, P. (2019). Web portals for high-performance computing: a survey. ACM Transactions on the Web (TWEB), 13(1), 1-36.

[12] React. A JavaScript library for building user interfaces - https://reactjs.org/

[13] Vue JS. The Progressive JavaScript Framework - https://vuejs.org/

[14] Angular. The modern web developer's platform - https://angular.io/

[15] Ruby on Rails. Compress the complexity of modern web apps - https://rubyonrails.org/

[16] Django. The web framework for perfectionists with deadlines - https://www.djangoproject.com/

[17] Flask. Web development one drop at a time - https://flask.palletsprojects.com/

[18] FastAPI - https://fastapi.tiangolo.com/

[19] VNC. Virtual Network Computing - https://en.wikipedia.org/wiki/Virtual_Network_Computing

[20] noVNC - https://novnc.com/

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

Executive Summary

The HEROES Project is aiming at developing an innovative European software solution allowing industrial and scientific user communities to easily submit complex Simulation and ML (Machine Learning) workflows to HPC (High Performance Computing) Data Centres and Cloud Infrastructures. It will allow them to take informed decisions and select the best platform to achieve their goals on time, within budget and with the best energy efficiency.

This document presents how a simple web interface can be built on top of HEROES APIs. Through an example of a Workflow Interface for the end-user, we present the prerequisites, technologies, and integration process of the interfaces within the HEROES Platform that have been elaborated during the development of the project.

This deliverable is accompanied by a tarball containing the current version of the code for the workflow interfaces.

The development of this document is closely linked to the ongoing developments of WP2 and WP3 of the HEROES project.

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

1 Introduction

Engineers and scientists seldom have the technical computer skills required to access a complex computing platform such as HEROES through a command line or REST APIs. They need to concentrate on their tasks and field of expertise, e.g., designing cars or planes or discovering new drugs... The best way to provide them access to the tools and computing power they need is through simple and intuitive web interfaces that will guide them through the process of submitting and managing their computational workflows.

The goal of this deliverable is not to implement a full-fledged web interface following the UX and UI principles but rather to present the means to realize such developments or to integrate an already existing web portal with the HEROES capabilities.

We present in this document an implementation of a workflow interface that could serve as an example in specific deployments or integrations at the request of a HEROES client. If a large company wishes to implement their own HEROES platform, either they already have web interfaces and, in this case, they can directly connect them to HEROES APIs, or they need new interfaces and in this case such development would most certainly be specific to the client's needs.

Through simple use cases and examples, the following sections present how workflow interfaces designed using ReactJS framework can interact with the HEROES platform to allow authenticated users to realize various operation concerning their workflows such as listing, describing, submitting, monitoring, or cancelling them, as well as accessing their data.

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

2 Workflow Interfaces

In this section, we present the use cases we want to be available in the workflow interfaces, and the prerequisites and constraints that we have for the HEROES platform. Finally, we present a few technologies that could be used to implement such interfaces connected to HEROES APIs to deliver a service to the end-users.

2.1 Use Cases

We only consider use cases for end-users (referred to as the "User" in the remainder of this document). HEROES or Organization administrators could also benefit from an interface for the day-to-day management of the platform, the users, the workflows… this would be dealt with in another portal. Currently, all administrative actions must go through the APIs.

2.1.1 Login/logout

The User must be able to login into the interface with their HEROES credentials (username, organization, and password).

Once logged in, through their active session they have access to all the features they are entitled to in their Organization.

The User must be able to manually logout, to prevent any further action with their current session. Logout can also happen automatically after a period of inactivity.

The following use cases all consider as a prerequisite that the User is connected and has an active session.

2.1.2 Data management

The User must be able to browse their data (stored in their personal bucket – see D3.1 Architecture Design) and any data shared among their Organization (stored in a shared bucket).

Through the interface, the User must be able to upload and download data to a specific folder. They must be able to create/rename/delete new folders/files.

Data can be either input data that will be used to submit new workflows, or output data produced by the workflows.

2.1.3 Submitting workflow

This use case is divided into several steps / sub-use-cases.

1. The Users must be able to see the list of workflows they have access to in their Organization.

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

2. They can select a workflow to submit it. They then need to specify:
    a. Any input parameters (e.g., data or configuration parameters for the workflow)
    b. Any resources requirements (e.g., number of CPU/RAM.)
    c. The placement policy selected from the ones provided by the decision module
3. Once all parameters validated, the users will get feedback from the decision module, presenting them with a list of potential clusters on which the workflow can run (e.g., resources constraints), ranked by decreasing order of the selected placement policy (along with estimated values for e.g., execution time, costs, or energy consumption). They need to select a cluster in this list to effectively submit the workflow for execution.
4. When the workflow runs, the User can monitor its status.
5. When the workflow has ended, the User can display and download the results (see Section 2.1.2) and display some information/metrics about the workflow and its jobs.

2.2 Prerequisites and constraints

Though the interface could be developed with any kind of technology permitting the User to interact with HEROES (CLI, "thick client" …), a web-based approach seems the most natural choice to allow pervasive access to the HEROES platform (e.g., no OS or device constraints).

The HEROES Initial Platform has been designed to integrate a single public point of access through the User Interface module. This design includes technological constraints coming from the project development methodology and tools as mentioned at the "Section 4: Containerized Platform and Automated CI/CD" in deliverable "D3.3 – Initial Platform Mock-up".

As part of the "User Interface module" the Workflow Interfaces must be integrated in a container that will be deployed in our Kubernetes cluster. Note that this is not a hard requirement, as HEROES provides APIs, any interface (even external to the HEROES platform) could use these APIs and provide the functionalities to the end-users. The interactions of the Workflow Interfaces are only linked to the FastAPI server embedded in the "Service Gateway", and accessible through REST APIs.

The selected technology to develop the Workflow interfaces must then comply with the following constraints:

- Must be able to communicate through HTTPS requests with FastAPI.
- Can be containerized to run in a Kubernetes cluster. The less external dependencies it has, the simpler it is to package it.
- Ideally the solution relies on open-source technologies to ease its adoption by a developer community. Licenses of the tools and dependencies must be compatible and ideally allow them to be embedded in commercial solutions to allow HEROES to develop different business models (e.g., MIT, BSD, Apache v2).

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

2.3 State of Art: Web UIs

This section describes different technologies that can be used to develop the HEROES Workflow Interface: “Open OnDemand” and some web frameworks that can be used for specific developments.

It is worth noting that NextFlow (the workflow manager that HEROES uses internally) does not provide any standard Web UI [3], this is left to users to develop their own (e.g., DolphinNext [4]). Though this might have been interesting to have such NextFlow interfaces, this wouldn’t have fitted exactly what we want to do as we wouldn’t have been able to map the functionalities directly only on HEROES APIs.

There are many other HPC web portals that exist in the market, most of them commercial, but they do not offer the required level of customization required to integrate them with the HEROES APIs. Some examples are: NICE EnginFrame [7], Altair® Access™ [8], IBM Spectrum LSF Application Center [9], Bull extreme computing studio [10]; additional HPC web portals can be found in [11].

2.3.1 Open OnDemand

Open OnDemand [5][6] (OOD) is an NSF-funded open-source HPC portal. The goal of Open OnDemand is to provide an easy way for system administrators to provide web access to their HPC resources. The code is available as open source under the terms of the MIT License which is compatible with developments we could make in HEROES and the will to potentially resell the platform or its components.

OPEN
OnDemand

Figure 1. Open OnDemand logo

OOD has many advantages such as: an existing and user-friendly interface, a large user & developer community, extensible to add new features/applications and backends.

Though OOD is a very good candidate, we decided not to use it in our prototype as it was complicated to use it in our development and backend infrastructure: OOD couldn’t be run within a Kubernetes container, thus making it complicated to deliver it with all the other components of HEROES (this is not a hard requirement, but this was helpful for our developments). The solution we took was to develop our own web portal with a standard web-framework, but with limited features, as a demonstrator of what a HEROES portal could be and as an example of how another portal could be integrated with HEROES APIs.

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

2.3.2 Web Framework

There are many frameworks available to develop web portals. To cite a few:

- Frontend frameworks: React JS [12], Vue JS [13], Angular [14]
- Backend frameworks: Ruby on Rails [15], Django [16], Flask [17], FastAPI [18]

As in HEROES we already have a backend framework (FastAPI), we only needed to select a frontend framework to develop our HEROES portal, namely React JS. It is an open-source framework for JavaScript for building user interfaces and their components, part of the overall React Native framework for building mobile applications. React is maintained by Facebook and the wider community and is currently the most popular front-end framework in the past year, experiencing rapid growth due to its ease of use and flexibility. We also selected this technology as it can easily be containerized in our Kubernetes infrastructure.

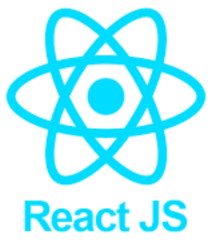


Figure 2. React JS logo

We present in Section 3.3 the developments made with React JS and the integration with FastAPI.

3 HEROES web interfaces

3.1 Integration in the architecture

The development of Web interfaces relies on the developments made on the "Workflow and Job Management" module and the Service Gateway (FastAPI), described in deliverable "D3.3 – Initial Platform Mock-up". The Service Gateway (FastAPI) is the entry point of the whole HEROES architecture, it provides APIs that allows to interact with the different modules of HEROES.

The User Interfaces have been integrated all along the HEROES Initial Platform development because of its frontal role in the architecture design. The User Interface has been designed to communicate with FastAPI into the platform to perform various actions including but are not limited to the workflow interactions. We present in Figure 3 where the user interfaces are integrated in the HEROES architecture.

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

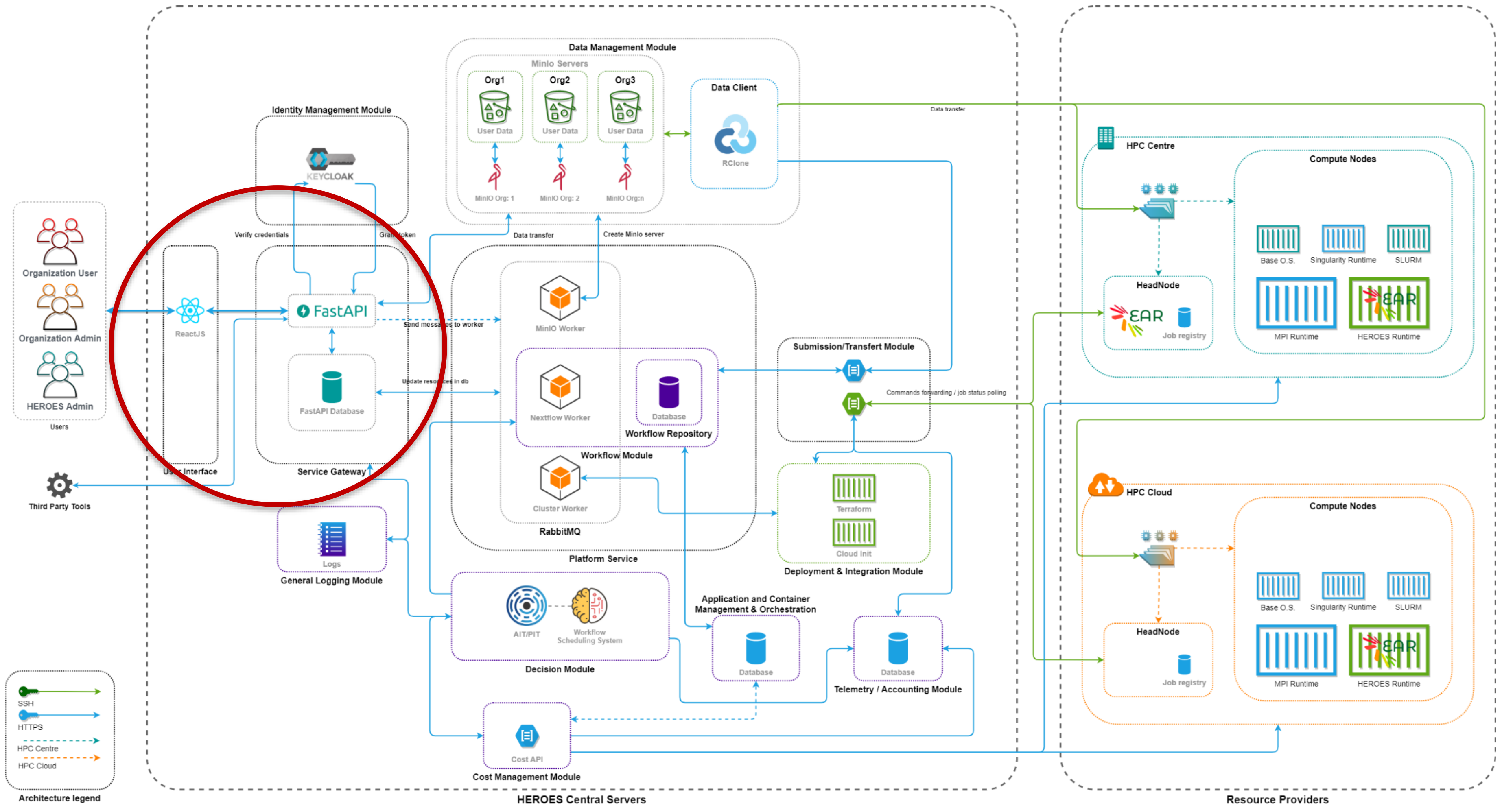


Figure 3. Integration of the user interfaces in the HEROES Architecture

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

3.2 React JS and FastAPI integration

As React JS is a purely client-side JavaScript based solution, meaning that all actions/requests come from the users' browsers. This has an implication on how FastAPI must be accessible: it is not the React JS server that communicates with FastAPI (which would allow to keep FastAPI APIs private, and only have React JS server public), but the user browser that directly communicate with FastAPI. Thus, the users need to be able to directly access the FastAPI APIs. The implication is twofold:

1. FastAPI server must be accessible from all HEROES users (IP/hostname accessible from where the users are, and firewall rules opened for HTTPS).
2. To allow the communication between FastAPI (Service Gateway) and the Workflow Interfaces, we need to enable the "Cross Origin Resource Sharing" (CORS[1]) in FastAPI. This is required for the integration of a middleware component to handle the requests with the Workflow Interfaces as their origin. To do so, we integrate the `CORSMiddleware` component from the `starlette` package in the FastAPI code:

```
middleware = [
    Middleware(
        CORSMiddleware,
        # Allow all origins as users can be anywhere, or just a subset of
DNS names in case of a private deployment
        allow_origins=['*'],
        # Indicate that cookies should be supported
        allow_credentials=True,
        allow_methods=['*'],
        allow_headers=['*']
    )
]
```

With these modifications, React JS and FastAPI can communicate, making it possible to develop the web interface. Note that this modification on FastAPI configuration for CORS is not specific to React JS but to any frontend framework that would be used to develop another web interface. This wouldn't have been needed if a backend framework had been used, as in this case both the backend framework and FastAPI would both have been backends, users' browsers interacting only with the backend framework.

3.3 HEROES web interface prototype

The prototype web interface provides the following features:

1. User authentication through username and password
2. Job/workflow submission, monitoring and management
3. Data browsing, upload/download, and management
4. Access to a remote visualization platform for pre/post-processing

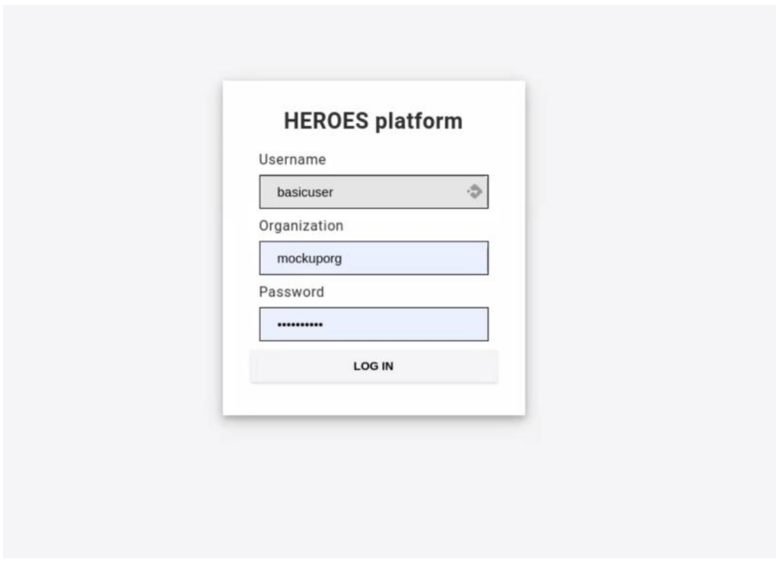
[1] CORS or "Cross-Origin Resource Sharing" refers to the situations when a frontend running in a browser has JavaScript code that communicates with a backend, and the backend is in a different "origin" than the frontend. Which is the case here as the users can be "anywhere". See https://fastapi.tiangolo.com/tutorial/cors/

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

Features 1 to 3 rely directly on HEROES APIs, while feature 4 is currently an additional feature that is not managed by HEROES. The remote visualization part relies on a centralized server/VM on which a VNC [19] server is started. The VNC protocol is proxied to make it accessible through a browser (via WebSockets thanks to noVNC [20]).

We present below the interfaces that have been developed for the prototype, along with some examples of code to explain how React JS was interfaced with FastAPI.

3.3.1 User authentication through username and password


HEROES platform

Username: basicuser

Organization: mockuporg

Password: ••••••••••

Figure 4. Login page

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

```
import axios from 'axios';

const ENDPOINT = 'https://fastapi.heroes.doit.priv'

export default async function login ({ username, organization, password })
{
    const data = {
        "username" : username,
        "organization" : organization,
        "password" : password,
    }

    const url = `${ENDPOINT}/organization/auth/login`

    const res = await axios.post(url, data, {
        headers: {
            "accept": "application/json",
            "Content-Type": "application/json",
        },
    });
    if (!res)
        throw new Error("Response is not OK");

    let token = res.data
    let org = res.data.organization
    return [token, org];
}
```

Once logged in, the user has access to the home page presented in Figure 5. The left-hand menu gives access to all the features.

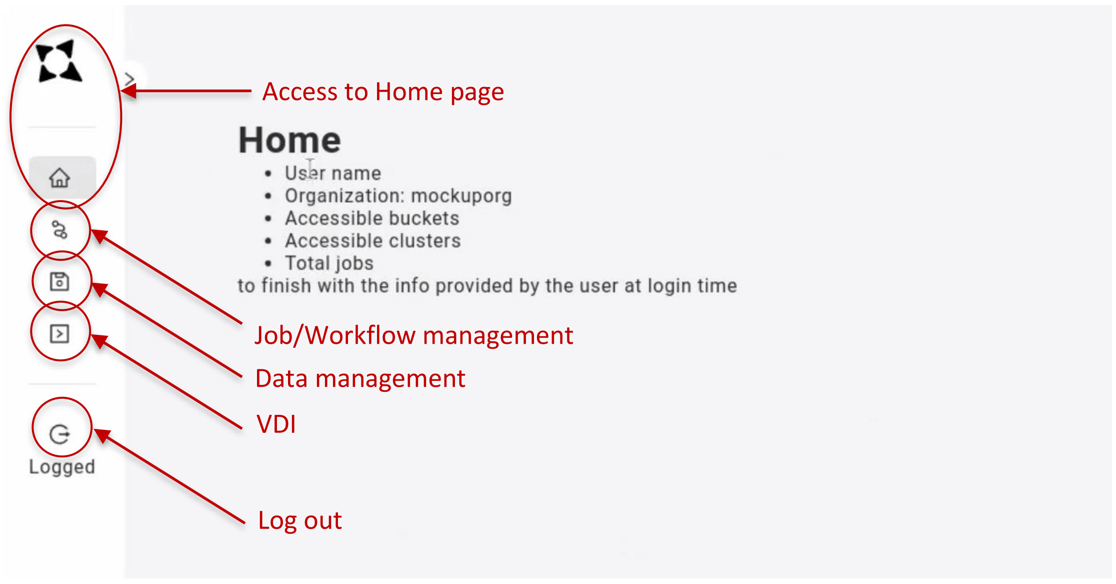
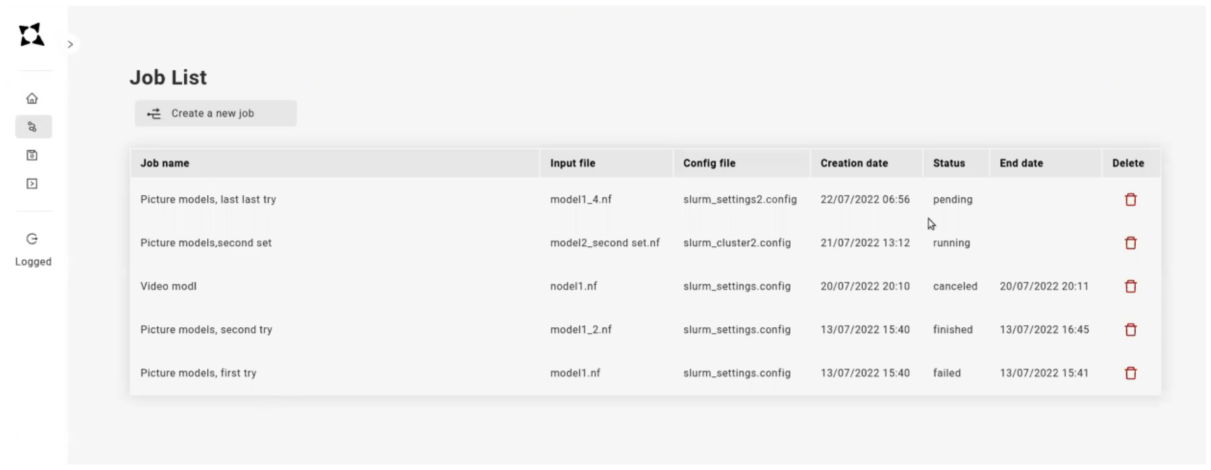


Figure 5. First version of the home page once logged-in

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

3.3.2 *Job/workflow submission, monitoring and management*

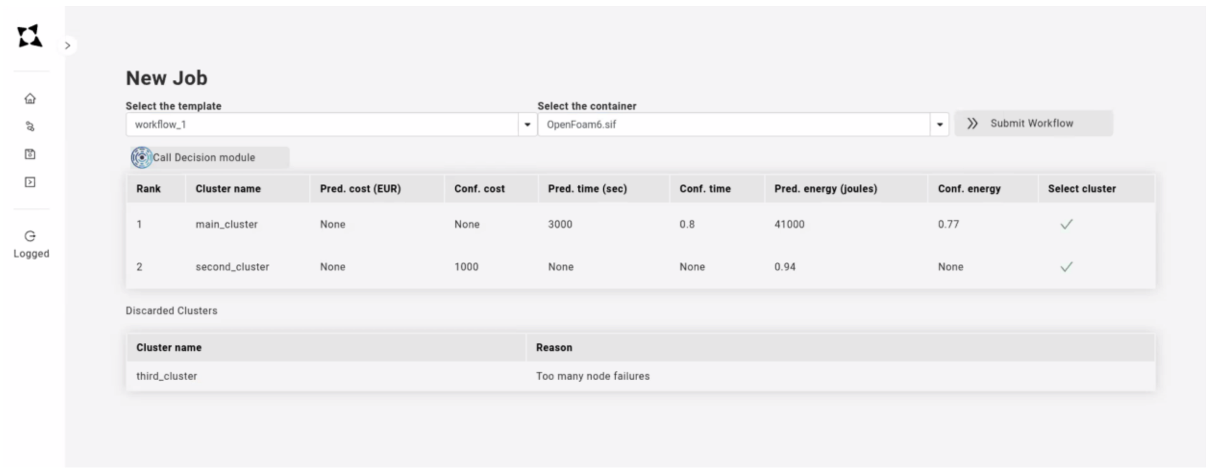
Through the interface the user can list/monitor their current workflows and delete them (see Figure 6) and can submit new workflows (see Figure 7). During the submission process, the user will be guided in the selection of the target cluster for their workflow: this is shown in Figure 7 through the "Call Decision module" button, the response is displayed below in a table listing the clusters in the order the Decision Module has ranked them. A table also displays a list of clusters that have been discarded during the decision process, with the reason why it has been discarded (in this example, the "third_cluster" has been discarded because recently there has been too many jobs killed due to a "node failure" – threshold defined in the Decision Module).



Job List

Job name	Input file	Config file	Creation date	Status	End date	Delete
Picture models, last last try	model1_4.nf	slurm_settings2.config	22/07/2022 06:56	pending		
Picture models,second set	model2_second set.nf	slurm_cluster2.config	21/07/2022 13:12	running		
Video modl	nodel1.nf	slurm_settings.config	20/07/2022 20:10	canceled	20/07/2022 20:11	
Picture models, second try	model1_2.nf	slurm_settings.config	13/07/2022 15:40	finished	13/07/2022 16:45	
Picture models, first try	model1.nf	slurm_settings.config	13/07/2022 15:40	failed	13/07/2022 15:41	

Figure 6. List of job/workflow executions



New Job

Select the template: workflow_1

Select the container: OpenFoam6.sif

Rank	Cluster name	Pred. cost (EUR)	Conf. cost	Pred. time (sec)	Conf. time	Pred. energy (joules)	Conf. energy	Select cluster
1	main_cluster	None	None	3000	0.8	41000	0.77	✓
2	second_cluster	None	1000	None	None	0.94	None	✓

Discarded Clusters

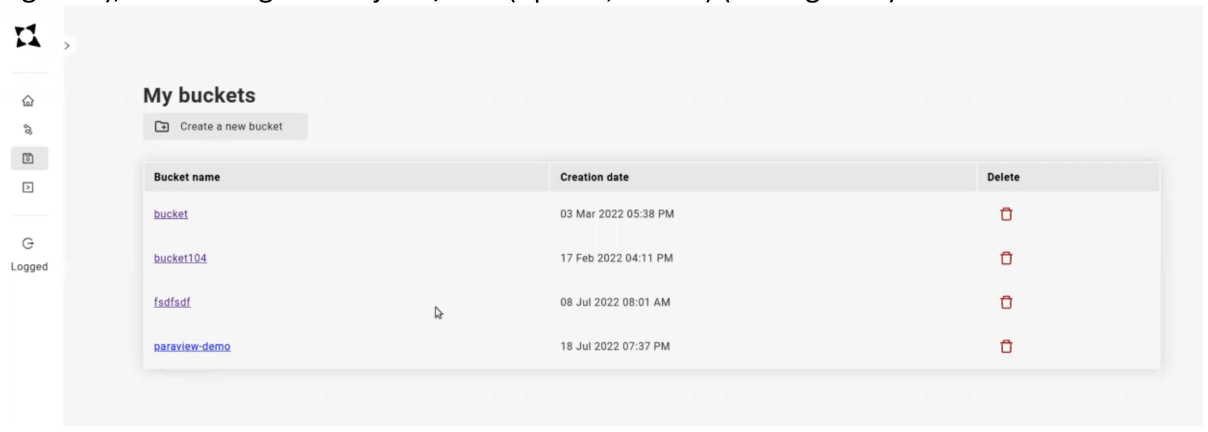
Cluster name	Reason
third_cluster	Too many node failures

Figure 7. Job/Workflow submission

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

3.3.3 Data browsing, upload/download, and management

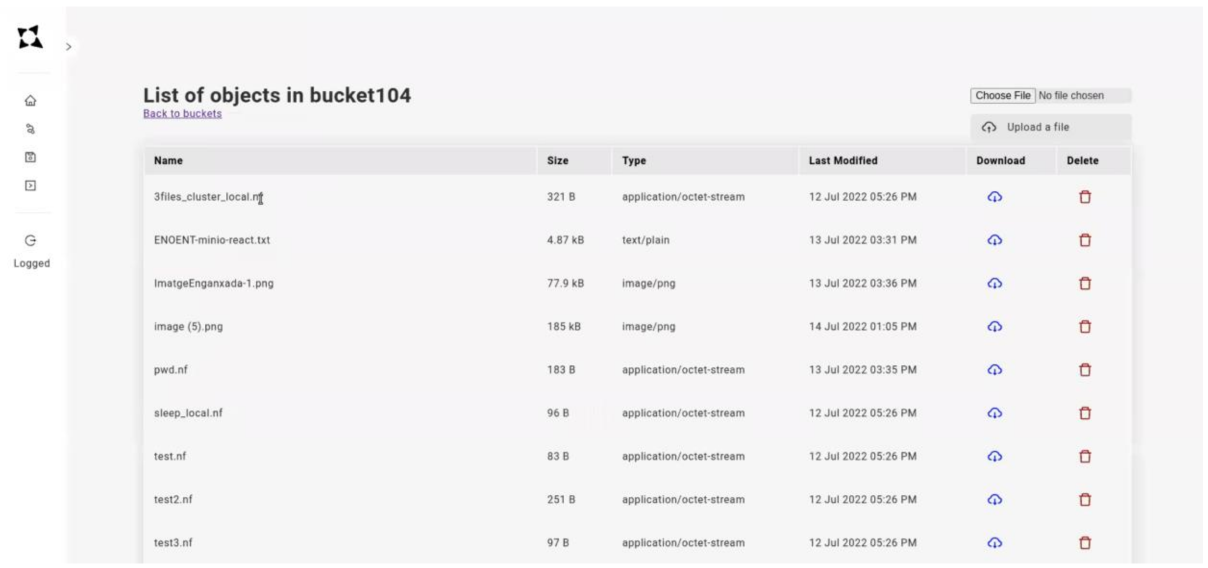
Through the interface the users can list the buckets/directories they have access to (see Figure 8), and manage his objects/files (upload, delete) (see Figure 9).



My buckets

Bucket name	Creation date	Delete
bucket	03 Mar 2022 05:38 PM	
bucket104	17 Feb 2022 04:11 PM	
fsdfsdf	08 Jul 2022 08:01 AM	
paraview-demo	18 Jul 2022 07:37 PM	

Figure 8. Data management: list of buckets



List of objects in bucket104

Name	Size	Type	Last Modified	Download	Delete
3files_cluster_local.nf	321 B	application/octet-stream	12 Jul 2022 05:26 PM		
ENOENT-minio-react.txt	4.87 kB	text/plain	13 Jul 2022 03:31 PM		
ImatgeEnganxada-1.png	77.9 kB	image/png	13 Jul 2022 03:36 PM		
image (5).png	185 kB	image/png	14 Jul 2022 01:05 PM		
pwd.nf	183 B	application/octet-stream	13 Jul 2022 03:35 PM		
sleep_local.nf	96 B	application/octet-stream	12 Jul 2022 05:26 PM		
test.nf	83 B	application/octet-stream	12 Jul 2022 05:26 PM		
test2.nf	251 B	application/octet-stream	12 Jul 2022 05:26 PM		
test3.nf	97 B	application/octet-stream	12 Jul 2022 05:26 PM		

Figure 9. Data management: list of objects in a bucket

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

```
const listObjectsOfBucket = async () => {

fetch(`${ENDPOINT}/organization/data/bucket/${bucketName}/list?bucket_recur
sive=true`, {
        method: 'GET',
        headers: {
            "accept": "application/json",
            "Content-Type": "application/json",
            "token": `${jwt}`,
        },
        }).then((response) => response.json())
        .then((objects) => {
            setObjects(objects)
            setIsLoading(false)
        })
    }
```

3.3.4 Access to a remote visualization platform for pre/post-processing

Finally, the user can request to connect to a remote desktop for pre/post-processing their data (see Figure 10). The interactive window can be shown both embedded into the web interface or in a new tab of the browser.

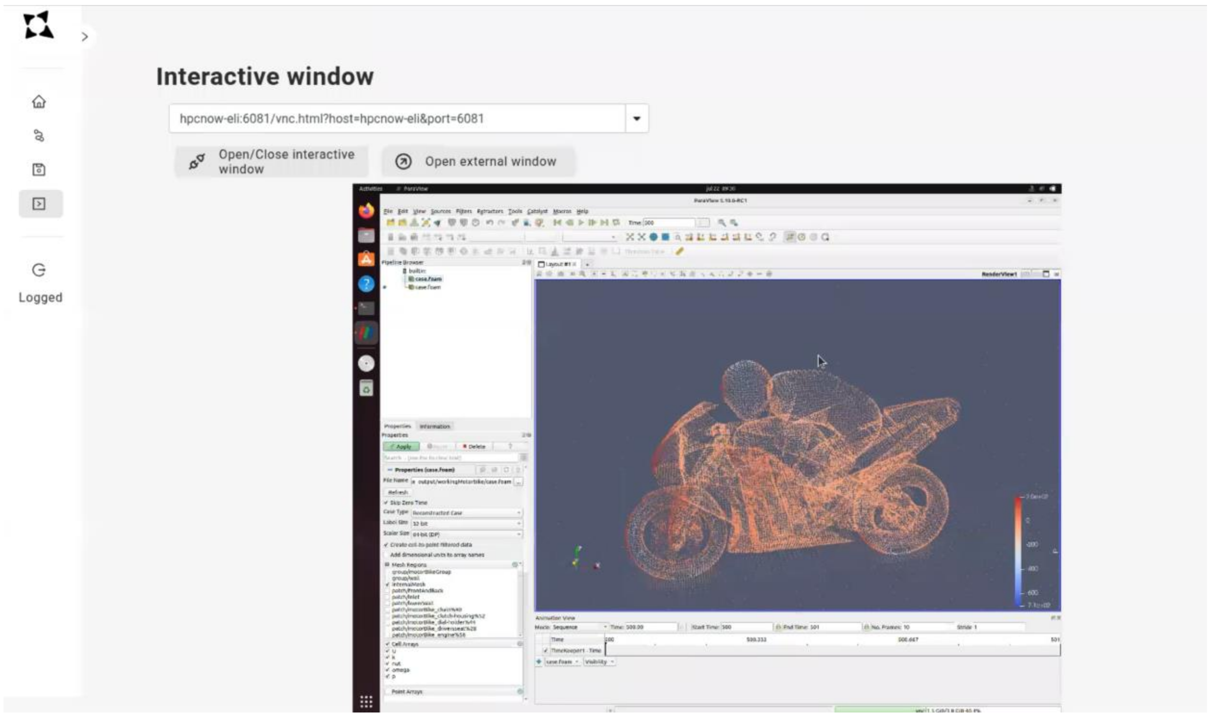

Interactive window
hpcnow-eli:6081/vnc.html?host=hpcnow-eli&port=6081
Open/Close interactive window
Open external window
Logged

Figure 10. Remote visualization feature through embedded VNC

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.

4 Conclusion

Accessing complex HPC or AI workflows can be made easy by providing the end-users a simple and user-friendly user interface. Building a new web interface or integrating HEROES features within an existing interface will be use-case dependent (large clients tend to already have interfaces in place, and completely changing them might be difficult).

In this document, we present an example implementation of such web interface, based on React JS, an open-source web frontend framework that we plugged on top of HEROES APIs.

The HEROES project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956874.