HORIZON-EUROHPC-JU-2021-COE-01

EXCELLERAT P2

The European Centre of Excellence for Engineering Applications

Project Number: 101092621

D4.4
Data Management and Data Analytics in EXCELLERAT

The EXCELLERAT P2 project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101092621. The JU receives support from the European Union's Horizon Europe research and innovation programme and Germany, Italy, Slovenia, Spain, Sweden and France.

Work Package:	WP4	Workflow Development
Author(s):	Marvin Hubl	SSC
Approved by	Executive Centre Management	20.12.2023
Reviewer	Andrej Kosicek	ARCTUR
Reviewer	Claudio Arlandini	CINECA
Dissemination Level	Public	

Date	Author	Comments	Version	Status
2023-10-18	Marvin Hubl	Report initialisation	V0.1	Initial
2023-10-26	Marvin Hubl	First draft	V0.7	1st review
2023-11-27	Marvin Hubl	Second draft	V0.8	2nd review
2023-12-11	Marvin Hubl	Revised version	V0.9	Revision
2023-12-20	Marvin Hubl	Final version	V1.0	Final

List of abbreviations

CAD	Computer-Aided Design
HPC	High-Performance Computing
IT	Information Technology
OEM	Original Equipment Manufacturers
SME	Small and Medium-sized Enterprise
SSH	Secure Shell

Executive Summary

Access to high-performance computing resources for research and development partners is crucial for innovation. Furthermore, there is an inherent need for cooperative work in engineering applications. However, the technological potentials for cooperative work with high-performance computing resources are not fully exploited and, especially for small and medium-sized enterprises, there exists a high threshold to make use of high-performance computing. This document therefore firstly reports on an essential systems analysis for identifying relevant requirements for high-performance computing in engineering applications. Second, this document reports on ease of use and information technological security concerns as crucial problems for relevant stakeholders with a focus on small and medium-sized enterprises. Third, this document reports on the envisaged approach to develop further the software tool SCALES from the first EXCELLERAT project as portal with an integrated connector for a federated engineering ecosystem to foster cooperative work while maintaining data security and ease of use for high-performance computing.

Table of Contents

Table of Figures

1 Introduction

Providing effective access to exa-scale computing is becoming increasingly important in order to compete successfully in the innovation race. This importance is felt in both academic research and industrial development. One of the main concerns is to facilitate collaboration in basic academic research and industrial engineering development.

A basis for successful cooperation between engineering research and development partners is the well-managed provision of data. Therefore, a crucial requirement is reliable data exchange to integrate distributed engineering workflows. Engineering workflows typically comprise several legally and economically autonomous partners – e.g. in a supplier-customer relationship. Recent technological tendencies foster multilateral network structures, rather than linear bilateral relationships, to enhance cooperative processes in platform-based engineering ecosystems [1]. Platforms therefore logically include a standard or a common gateway to access digital services that support engineering processes.

For engineering development partners, a major barrier to using exa-scale or high-performance computing (HPC) capabilities is that the partners often have no explicit knowledge of how to use HPC resources. Using HPC resources requires specific, non-standard know-how for setting up, storing, and retrieving data to configure workflows, e.g. for simulation campaigns. The barrier is amplified when multiple HPC resources with distinct access and workflow specificities are included in a cooperative engineering process. Particularly, small and medium-sized enterprises (SMEs) often do not have the requisite knowledge capital or personnel resources to effectively integrate HPC in their product development processes. However, the appropriate usage of HPC can reduce the duration of development cycles and productions costs, e.g. by highly realistic simulations of mechanic properties or other technical attributes before constructing a respective physical prototype. Thus, HPC usage can enhance the innovativeness of engineering suppliers tremendously.

Against this backdrop, the objective is to help introducing HPC usage and ensure a broader access to a wider user community, in particular industry and SMEs. Since there are no uniform or standardised rules and it is left to each user how the data and the data storage is handled. Consequently, it is necessary to establish a standard for handling and storing data. Therefore, we contribute (1st) an initial problem analysis, (2nd) stakeholder requirements and (3rd) an architectural design based on standardisation endeavours for an open, data-based cooperative work in engineering applications for HPC.

The subsequent section reports on the basis for the technological development. Afterwards the methodological approach for identifying problems and requirements is described. Then, the initial architectural design conceptualisation based on the identified requirements will be outlined. Subsequently, the connection to other EXCELLERAT P2 tasks and an outlook on the work plan is provided. Then, the solution approach is briefly discussed with respect to expected merits. The last section concludes the report.

2 Technology basis

2.1 Data exchange and management with SCALES

The technological development is based on the data exchange and management tool SCALES, continuing developments from the first phase of EXCELLERAT [2]. In the following the state of SCALES as of the first phase is presented. The platform is not only used for data processing, but also enables a safe and traceable, online data transfer between the data generators and several HPC centres represented in the EXCELLERAT project. This data transfer will be highly automated to avoid duplication of the transferred content. This approach reduces the amount of transferred data, which can ultimately save time and costs. The portal divides uploaded or stored data into chunks. Then hash values of the chunks are calculated and used as the basis for comparisons of data chunks and duplication discovery. Data chunks that had already been uploaded or stored do not need to be uploaded or stored again. The portal provides relevant HPC processes for the end users, such as uploading input decks, scheduling workflows, or processing HPC jobs.

The added value of the workflow portal in relation to exa-scale data are the topics such as data reduction, volume reduction and data compression of input and output data. In concrete terms, this means that the data becomes smaller and data transport becomes more manageable for the data transport.

Figure 1 sketches a scheme of the logical components and used technologies for SCALES. The functionality of SCALES is described as a user process in the following subsection.

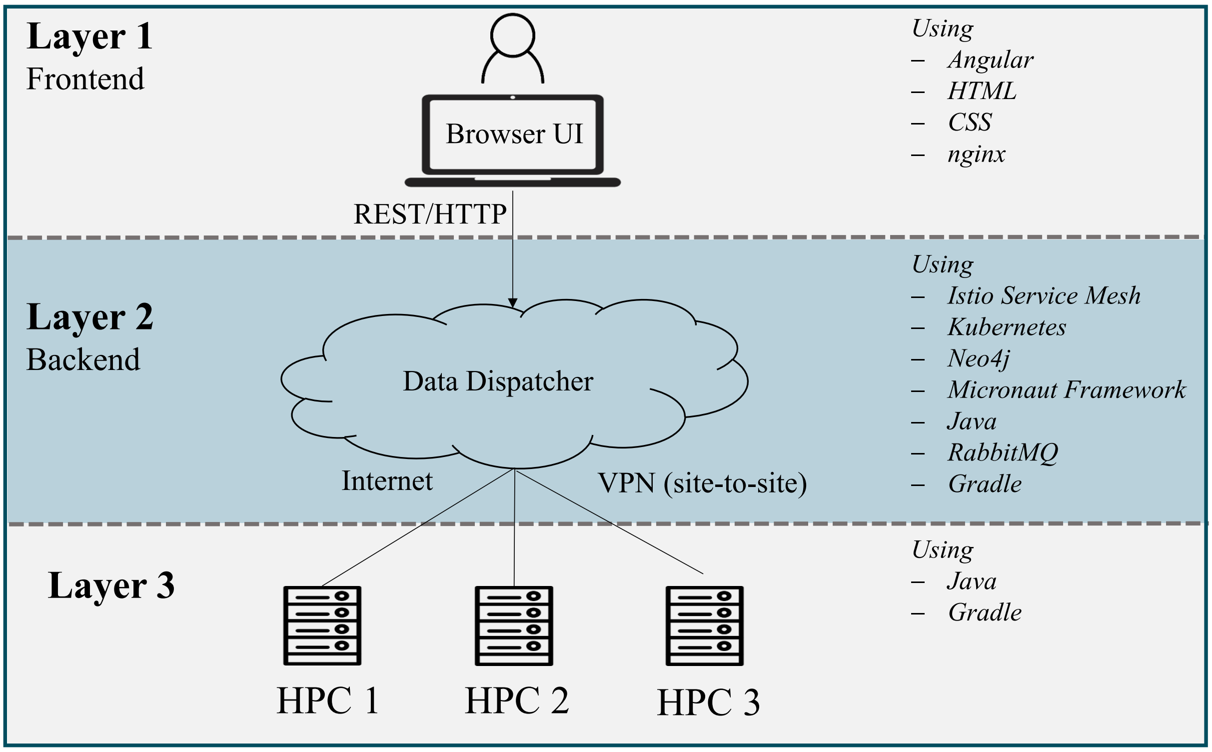
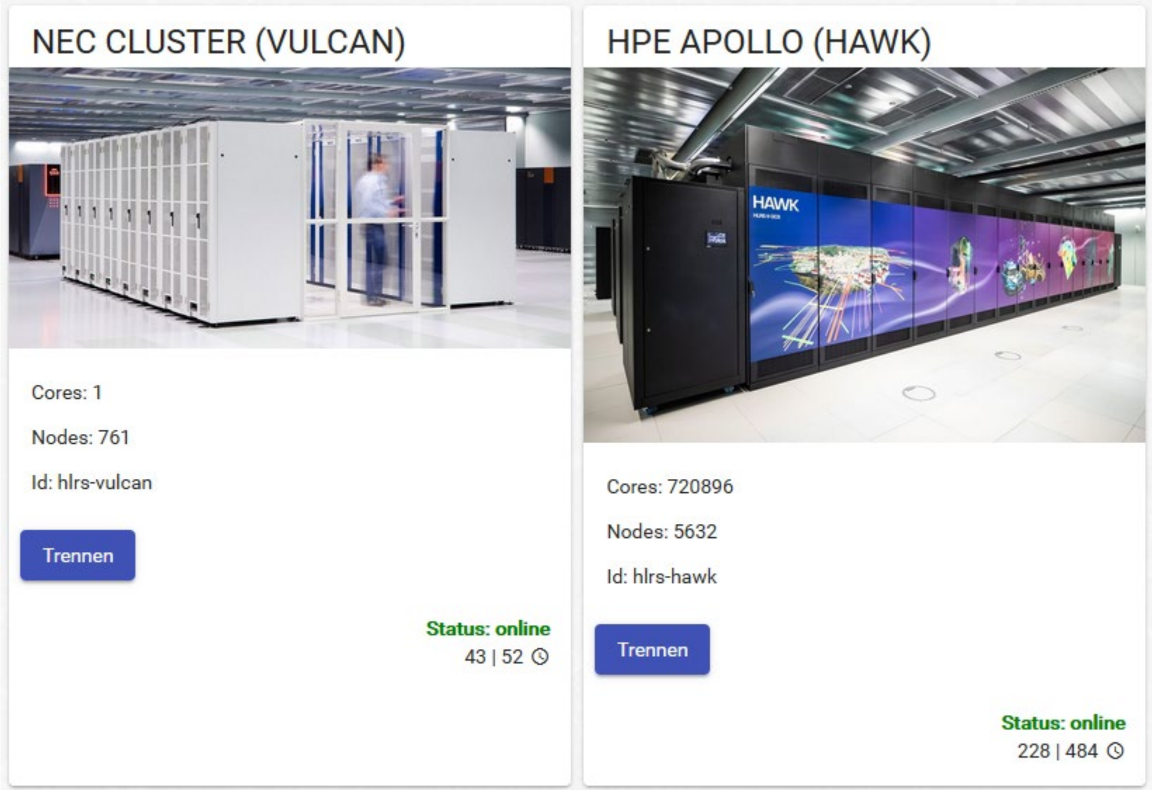


Figure 1: Scheme of SCALES as the technological basis

2.2 SCALES user process

After a successful login, the user starts on a web-based dashboard page. To use the corresponding HPC resources, a connection to the cluster, on which the calculation is to be performed, is required, see Figure 2. To connect the platform with the machine user, the Secure Shell (SSH) access data of the corresponding cluster must be entered once.



NEC CLUSTER (VULCAN)

Cores: 1

Nodes: 761

Id: hlrs-vulcan

Status: online

43 | 52

HPE APOLLO (HAWK)

Cores: 720896

Nodes: 5632

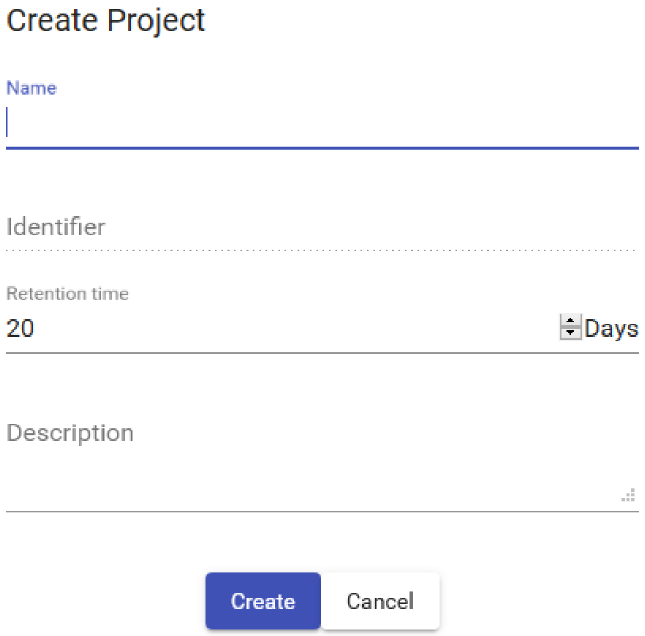
Id: hlrs-hawk

Status: online

228 | 484

Figure 2: Connection to example HPC cluster

The basic structure of the platform consists of projects corresponding to the workspaces on the clusters. This means, that when creating a project, a workspace with the same name is created on the cluster. When a new project is initially created, a name must be assigned, see Figure 3. In addition, you must specify how long the retention period (1-30 days with a possible extension of three times, which corresponds to a total of 120 days) should be. A description is optional.



Create Project

Name

Identifier

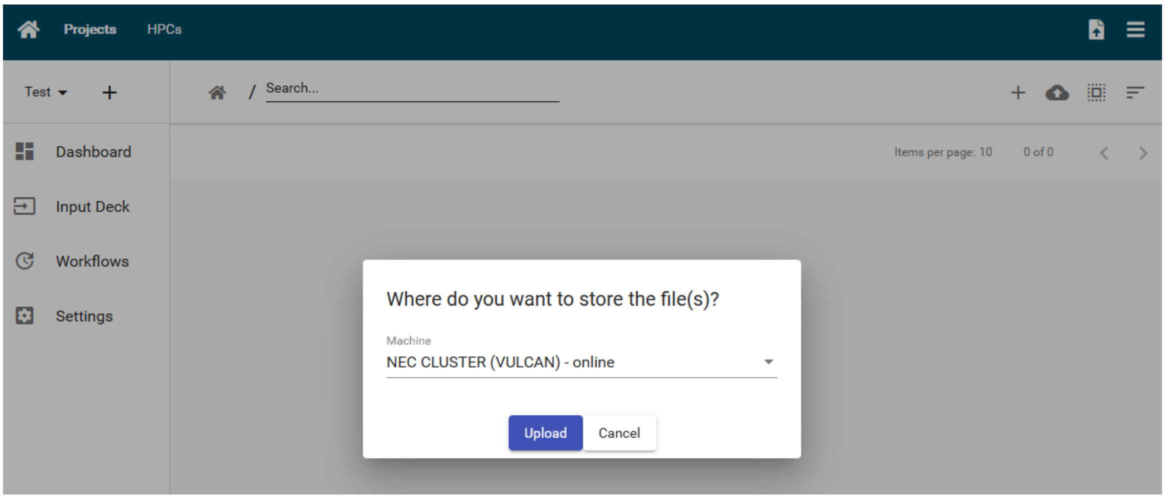
Retention time

20 Days

Description

Figure 3: Creation of a SCALES project

The project consists of a dashboard, an input deck, workflows, and project settings. The dashboard shows the most recently executed workflows and the most recent notes. To prepare the simulation, data is created locally and can be uploaded to the corresponding cluster through the input deck menu, see Figure 4. After the appropriate files have been selected, the user is asked on which machine these files should be uploaded. As part of further development, the aim is for the platform to decide where which code is stored for the best calculation. The progress of uploading the local files is displayed to the user in the interface. During the upload, the system checks whether there are identical file pieces, that are already in use and do not need to be uploaded again.

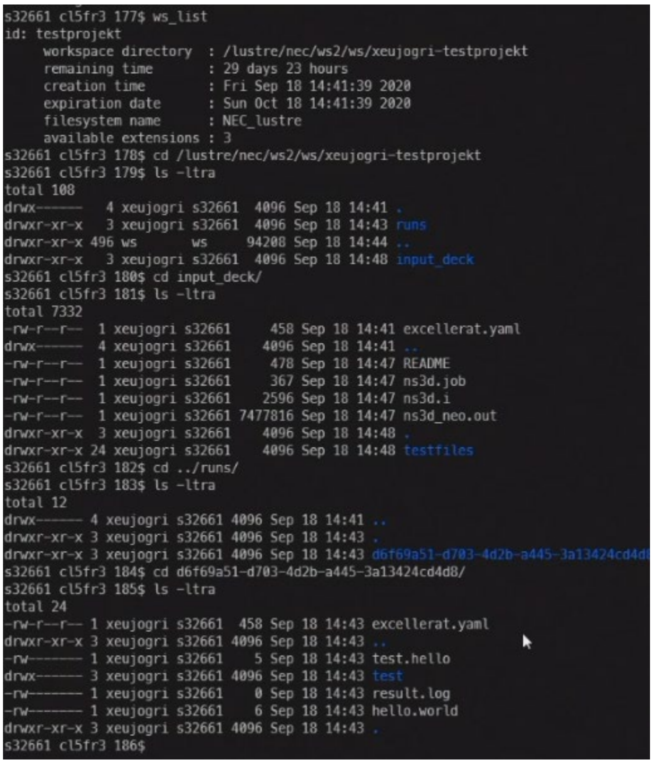


Where do you want to store the file(s)?

Machine: NEC CLUSTER (VULCAN) - online

Figure 4: Uploading an input deck with SCALES

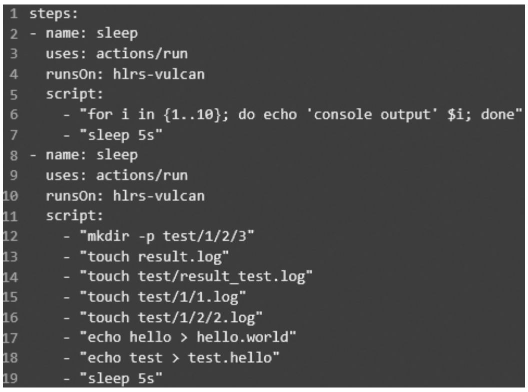
All uploaded files end up in the editor's workspace. For the platform, a specific folder structure is created on the HPC cluster so that all files can easily be found again, see Figure 5. However, all modifications, that take place outside the platform are not monitored, since no direct accesses are integrated into the cluster. The following folder structure is used: the "*input_deck*" contains all input data, that have been uploaded. "*runs*" contains all executions with the resulting data.



```
s32661 cl5fr3 177$ ws_list
id: testprojekt
     workspace directory  : /lustre/nec/ws2/ws/xeujogri-testprojekt
     remaining time       : 29 days 23 hours
     creation time        : Fri Sep 18 14:41:39 2020
     expiration date      : Sun Oct 18 14:41:39 2020
     filesystem name      : NEC_lustre
     available extensions : 3
s32661 cl5fr3 178$ cd /lustre/nec/ws2/ws/xeujogri-testprojekt
s32661 cl5fr3 179$ ls -ltra
total 108
drwx------   4 xeujogri s32661  4096 Sep 18 14:41 .
drwxr-xr-x   3 xeujogri s32661  4096 Sep 18 14:43 runs
drwxr-xr-x 496 ws       ws     94208 Sep 18 14:44 ..
drwxr-xr-x   3 xeujogri s32661  4096 Sep 18 14:48 input_deck
s32661 cl5fr3 180$ cd input_deck/
s32661 cl5fr3 181$ ls -ltra
total 7332
-rw-r--r--  1 xeujogri s32661     458 Sep 18 14:41 excellerat.yaml
drwx------  4 xeujogri s32661    4096 Sep 18 14:41 ..
-rw-r--r--  1 xeujogri s32661     478 Sep 18 14:47 README
-rw-r--r--  1 xeujogri s32661     367 Sep 18 14:47 ns3d.job
-rw-r--r--  1 xeujogri s32661    2596 Sep 18 14:47 ns3d.i
-rw-r--r--  1 xeujogri s32661 7477816 Sep 18 14:47 ns3d_neo.out
drwxr-xr-x  3 xeujogri s32661    4096 Sep 18 14:48 .
drwxr-xr-x 24 xeujogri s32661    4096 Sep 18 14:48 testfiles
s32661 cl5fr3 182$ cd ../runs/
s32661 cl5fr3 183$ ls -ltra
total 12
drwx------ 4 xeujogri s32661 4096 Sep 18 14:41 ..
drwxr-xr-x 3 xeujogri s32661 4096 Sep 18 14:43 .
drwxr-xr-x 3 xeujogri s32661 4096 Sep 18 14:43 d6f69a51-d703-4d2b-a445-3a13424cd4d
s32661 cl5fr3 184$ cd d6f69a51-d703-4d2b-a445-3a13424cd4d8/
s32661 cl5fr3 185$ ls -ltra
total 24
-rw-r--r-- 1 xeujogri s32661  458 Sep 18 14:43 excellerat.yaml
drwxr-xr-x 3 xeujogri s32661 4096 Sep 18 14:43 ..
-rw------- 1 xeujogri s32661    5 Sep 18 14:43 test.hello
drwx------ 3 xeujogri s32661 4096 Sep 18 14:43 test
-rw------- 1 xeujogri s32661    0 Sep 18 14:43 result.log
-rw------- 1 xeujogri s32661    6 Sep 18 14:43 hello.world
drwxr-xr-x 3 xeujogri s32661 4096 Sep 18 14:43 .
s32661 cl5fr3 186$
```

Figure 5: Workspace structure on HPC filesystem

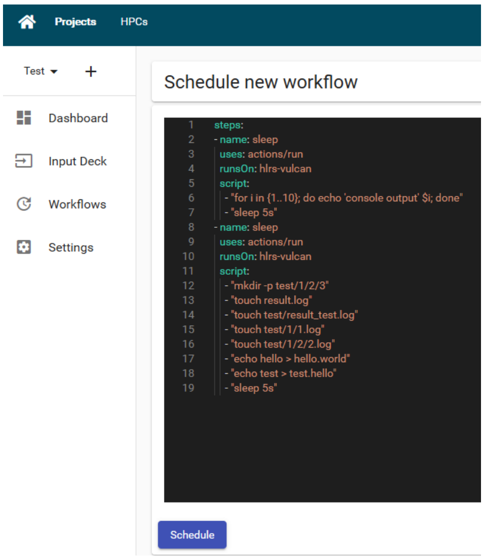
So far you can specify an "excellerat.yaml" file for each project, see Figure 6. This control file describes the workflows, how the simulation should look like as well as what should be done with it in the workspace. Additionally, you can specify scripts that may run in pre- and post-processing here. Those could be included in the YAML file as batch scripts or uploaded at the beginning and called in YAML.



```
steps:
- name: sleep
  uses: actions/run
  runsOn: hlrs-vulcan
  script:
    - "for i in {1..10}; do echo 'console output' $i; done"
    - "sleep 5s"
- name: sleep
  uses: actions/run
  runsOn: hlrs-vulcan
  script:
    - "mkdir -p test/1/2/3"
    - "touch result.log"
    - "touch test/result_test.log"
    - "touch test/1/1.log"
    - "touch test/1/2/2.log"
    - "echo hello > hello.world"
    - "echo test > test.hello"
    - "sleep 5s"
```

Figure 6: Example for an excellerat.yaml file

After creating a new workflow, the uploaded excellerat.yaml is automatically loaded into this workflow, see Figure 7. In the background, the file is loaded from the cluster workspace to be displayed in the browser. Furthermore, the workflow can be changed manually. There is a validation of the commands and an auto-completion. Around the control file the input data is added to run the simulation. The corresponding workflow can be scheduled and started after pressing "Schedule". The platform component in the corresponding cluster executes the command and prepares the run script ("run.sh") from the input files and passes it to the simulation.



```
steps:
- name: sleep
  uses: actions/run
  runsOn: hlrs-vulcan
  script:
    - "for i in {1..10}; do echo 'console output' $i; done"
    - "sleep 5s"
- name: sleep
  uses: actions/run
  runsOn: hlrs-vulcan
  script:
    - "mkdir -p test/1/2/3"
    - "touch result.log"
    - "touch test/result_test.log"
    - "touch test/1/1.log"
    - "touch test/1/2/2.log"
    - "echo hello > hello.world"
    - "echo test > test.hello"
    - "sleep 5s"
```

Figure 7: Workflow scheduling

After the run has been successfully scheduled, each step is processed in the background, executed automatically and the user receives a browser pop-up notification at the end communicating that the run has been successfully completed, see Figure 8. In this example, a few files and a folder structure have been created in the workspace, which the user can now download.

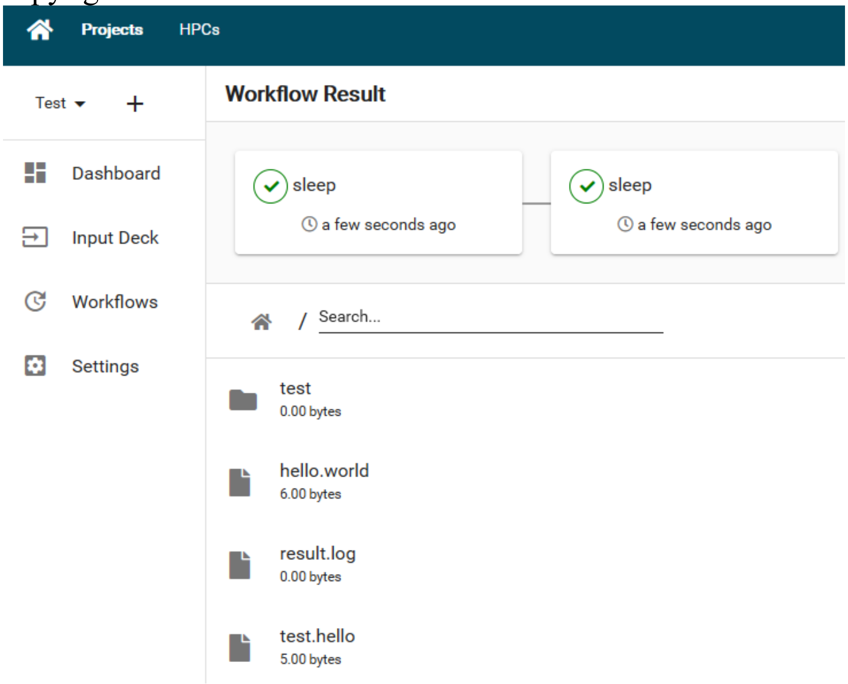


Figure 8: Workflow overview

3 Methodology for requirements analysis

In the first year of the project, the focus was on understanding challenges in cooperative engineering for industrial users, particularly for SMEs. Our goal is to establish a standard for cooperative work in an open infrastructure and to ensure broader access to a broader user community, in particular industry respectively SMEs. To this end, we analysed data management issues that partners encounter in HPC for engineering applications. We are pursuing a method that enables the target solution to be generic with regard to the variable boundary conditions (hardware, software, access requirements) of different HPC centres. The methodology is based on the so called “essential systems analysis” [3][4] and is executed both, literature-based with respect to recent standards and pertinent developments as well as based on discussions with the HPC experts from the project consortium during our regular project meetings. Furthermore, we took opportunities in the meetings with at least three of our industrial business partners to identify their prerequisites or barriers in using HPC resources when there was an appropriate context. We instantiate the essential systems analysis technique with modern technological background. An advantage of this technique is that it provides guidance in identifying a generic target model for heterogeneous systems, as shown in Figure 9.

Initially, information on a selected existing system and corresponding processes are gathered. With this information, the current as-is state of the system can be modelled. The result of the modelling process is a model of the existing physical system. From this model, the so-called system essences must be identified, i.e. the logical components, functionalities, and procedures in abstracted relations. The result of this activity is a logical as-is model of the system. As a “key feature” of the essential systems analysis methodology, the desired changes are defined with respect to the logical as-is model. This determines the logical target model.

The logical target model is then a common basis to design different physical target models. Therefore, the logical target model is, by conjunction, brought together with the boundary conditions of the system’s execution environment. With the logical target model and the boundary conditions, restrictions for the system instance are selected, resulting in the physical target model of the system in its specific execution environment.

For example, specific systems and processes run on each of the several HPC centres involved in EXCELLERAT P2. During the essential system analysis, several different physical as-is models are determined. However, the physical as-is models have common essences when software frameworks or tools and hardware arrangements are abstracted to their logical core functionalities and relationships to each other. Even if there are several logical as-is models, the goal is to define a logical target model that ultimately reflects the standardisation efforts. Therefore, the desired changes for the different logical as-is models need to be defined accordingly. Then, the standardized logical target model can be instantiated with respect to HPC systems or use case boundary conditions. Ideally, the selection of instance restrictions will be a configuration of the logical model serving as a standard respectively as a template.

Figure 9 sketches the standardisation objective in the analysis process by the number of parallel activities and outputs respectively inputs. The depiction of three existing systems in this diagram is exemplary. The depiction of two essence derivations is exemplary to indicate that the quantity of essences is lower than the number of the concerned physical systems. This is where we are expecting to make use of a particular benefit of the essential systems analysis. The definition of a single logical target model reflects the objective to establish a standard. Since the selection of instance restrictions can be imagined as setting parameters to instantiate a standard or template for certain boundary conditions, instance selection is also presented a logical activity. The essential activity of selecting instance constraints may not differ for different boundary conditions. Only different boundary conditions lead to different physical target models.

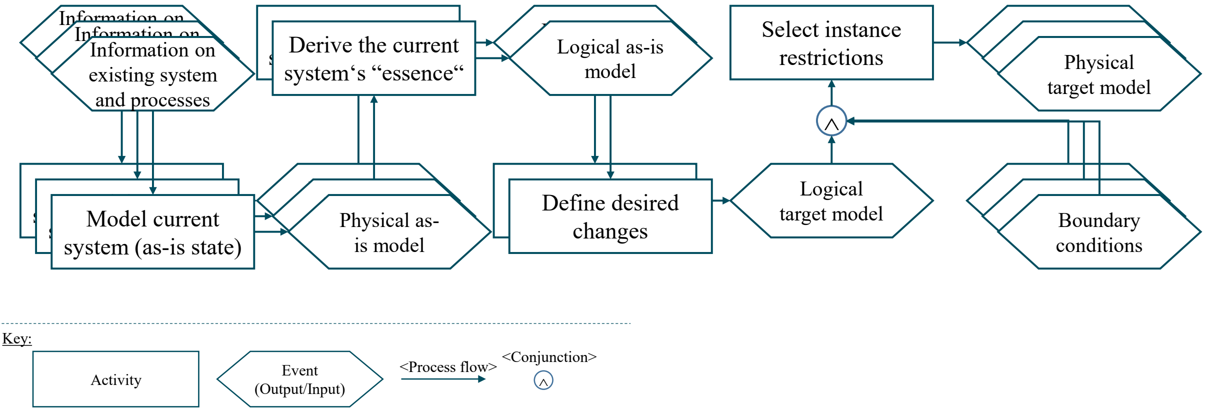


Figure 9: Structure of the analysis process for HPC usage with standardisation objective

4 User and system requirements

Engineering stakeholders from original equipment manufacturers (OEM) and suppliers to the automotive industry served as representatives for an analysis of requirements concerning the access and usage of HPC resources in engineering domains. We discussed industry challenges and expectations with partners in qualitative consultation meetings. The industry perspective has been complemented by discussions with HPC experts in the regular EXCELLERAT P2 meetings. For potential industry users, especially for SMEs, the following problems are salient.

The access to appropriate HPC resources has high perceived thresholds for industrial companies, especially for SMEs. Three barriers exist:

(1) Insecurities with respect to data transfer: For conducting simulation use cases of a SME on HPC resources, massive amounts of data must be transferred between the SME and an HPC resource. Often the duration of the data transmission is tedious and exposed to the risk of data losses or inconsistencies. Furthermore, severe IT security concerns, such as confidentiality, integrity, authenticity, availability, and non-repudiation prevent SMEs from transferring data to an external computing resource. HPC centres in

EXCELLERAT already have solutions in production to address those issues. However industrial companies often require an adaption with their respective IT governance respectively with their IT landscape, which usually varies from company to company. Hence a solution should provide a kind of a technical broker to mediate individual technical requirements and the technical solutions provided by HPC centres.

(2) Difficulties to access and use HPC resources: For using a particular HPC resource it is usually necessary to have specific knowledge about the corresponding software and hardware architecture as well as about specific scripting languages and scripts to interact with the HPC resource's operating system. Often, a training is necessary before being able to effectively use an HPC resource.

(3) Difficulties to identify an HPC resource which is optimised for the specific simulation use case of an SME: Simulation use cases are optimised with distinct algorithms which in turn are optimised for the HPC hardware where they are executed. Example use cases with distinct simulation models, algorithms and hardware are [5]:
    a) External aerodynamics shape optimisation in automotive,
    b) Aircraft simulations with emission models,
    c) High fidelity simulation of rotating parts,
    d) Large eddy simulation for confined explosions,
    e) Hydrogen combustion,
    f) Mitigation of aeroacoustic noise,
    g) Digital twin simulation of a wall of a tokamak fusion reactor.

Figure 10 outlines the crucial IT security requirements for engineering applications, referring to the problem respectively barrier (1) stated above. Product development in engineering is an important cooperation process between partners. Within a product development process, the substantial outputs which are supplied to each other are intangible goods. Not physical prototypes are common in this early phase of value creation, but information on how to eventually construct a product and product properties. This information reflects the core competencies which expose the partners in economic competition. The information includes crucial know-how of the partners. Hence the corresponding engineering data is critical and exchanging them therefore requires highest IT security. It is undoubtedly an economic imperative for every engineering partner to protect their critical engineering data as effectively as possible. This also applies to collaboration with each other, e.g. by running a cooperative simulation campaign on a shared HPC resource. Therefore, an essential prerequisite is data sovereignty. Each partner who provides data retains control over who uses the data, the period during which the data may be used, in which locations the data may be used and how the data may be used, including, where applicable, the obligation to delete the data [6].

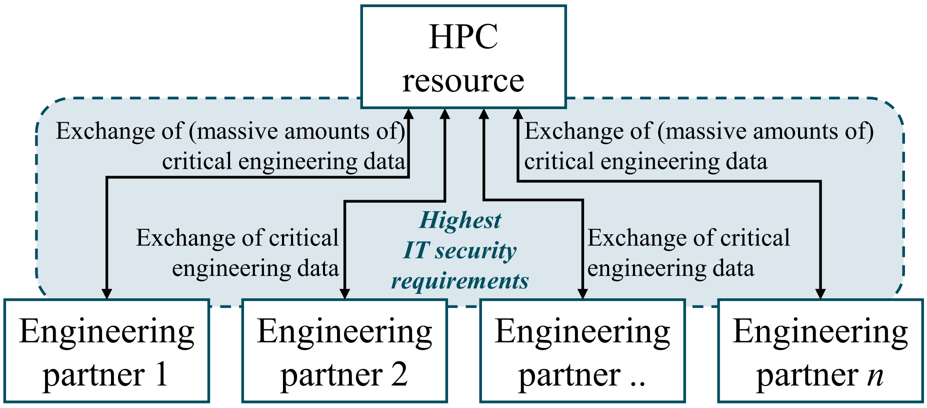


Figure 10: Cooperation in engineering applications with an HPC resource

Literature suggests a reference solution where data usage policies are determined separately from the actual data transfer and in a federated paradigm [7]. A rational of the paradigm is that data federation – i.e. data provision and consumption in a decentralised approach – prevents loss of data sovereignty. Centralised data nodes would inevitably shift control of the data provided to the centralised nodes. Note that this approach does not exclude the involvement of HPC nodes for certain data processing steps, but the approach will ensure data sovereignty, if HPC nodes are included. This addresses industrial engineering partner's IT security concerns with respect to data protection in terms of economic competition.

Based on the paradigm prior work in research and practice [8][9] developed a connector-based approach. Figure 11 outlines a basic architecture of a corresponding connector. The connector is a software module that connects nodes, data providers and data consumers with each other while maintaining data sovereignty for data exchange. To be able to send or receive data the node must be authenticated with an authentication service. This makes sure that nodes are the nodes that they purport to be. In a control layer the policies for data usage and storage are determined by a set of rules. In this phase meta-data, but no payload is exchanged. Only after the determination of the policies the actual data transfer is authorised. The transfer of the data payload is executed on a separated transfer layer that implements a data transfer protocol.

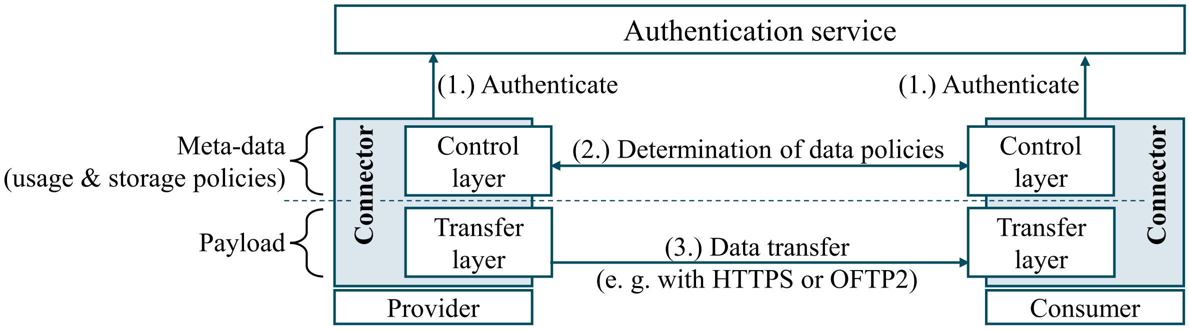


Figure 11: Connector-based approach to enhance cooperation in engineering applications for HPC

For the purpose at hand, the connector implements the requirements for functionalities of the logical target model and serves as gateway that encapsulates the respective boundary conditions of the partners' systems. The details of the encapsulated boundary conditions and the functionalities of the connector, specifically for HPC resources, still need to be elaborated. Relevant functionalities refer to information about the exact storage locations of data, the data types, the data age or the frequency of data use within the HPC resources.

With these requirements and solution approach, we are further developing the software tool "SCALES" from the first EXCELLERAT project as a data management portal [10]. Figure 12 shows how SCALES is planned to be integrated to be a gateway that provides the functionalities of a connector as described above as a managed service. This means, industrial engineering partners or HPC resources can make use of the connector functionalities in a low-threshold way by accessing SCALES. SCALES already is designed to enhance ease of use of HPC resources. In the further development, it is planned that SCALES implements a standardised policy framework concerning data usage and data storage for HPC in engineering applications.

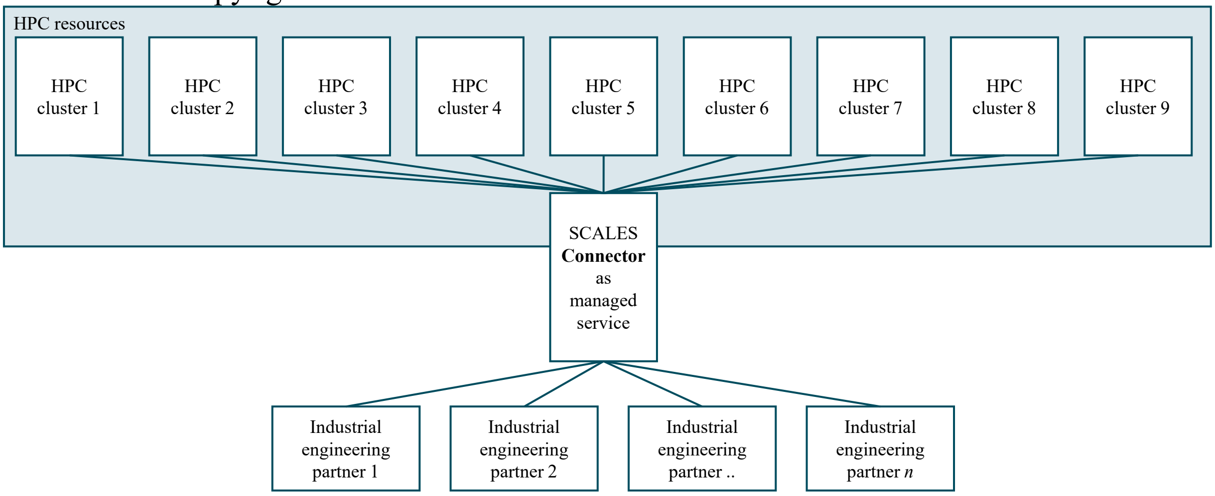


Figure 12: SCALES as connector for managed data exchange in cooperative engineering applications with HPC resources.

5 Connection to EXCELLERAT P2 tasks and work plan

The development described above is connected to other tasks within EXCELLERAT P2 as follows:

The analysed requirements and the developments referring to data management and data transfer in task 4.4 may provide inputs for the determination of use cases and workflow configurations in task 2.1 as well as for corresponding workflow demonstrations in task 2.2.

For data management problems, the results of task 4.4 may also provide a perspective on testing in task 3.3.

Service provisioning in task 5.1, training and education in task 5.2 and further applications in task 5.4 may rely on solutions elaborated in task 4.4 concerning data management and data transfer issues.

Since the development in task 4.4 seeks to tie up with recently emerging technologies for data management, results of task 4.4 are closely connected to business development in task 6.3 and market exploitation in task 6.4.

To achieve the objectives the further work plan is as follows:

First, the connector-based approach is developed. The connector serves as data gateway to different selected HPC resources via a portal. The timeline for this development is end of the year 2024 (project month 24, milestone 7).

When there is the proof of concept of the connector's gateway functionalities, the specific and differentiated requirements for data management and handling in HPC resources, pertaining to data handling, archiving, versioning, and reproducing are added to the gateway's connecting functionality for usage over the portal. The timeline for this development is end of the year 2026 (project month 48, milestone 10).

6 Discussion

The data management portal SCALES is expected to enhance efficiency in several ways but also to be an enabler for novel business services. First, the portal's capabilities are expected to reduce the amount and therefore time and ultimately cost of data transfer by identifying data chunk duplicates, which eventually cumulate to massive, wasteful redundancies.

Second, the portal's features are expected to accelerate development cycles for SMEs by providing them a low-threshold access to HPC capabilities. Hence SMEs will be able to conduct several highly realistic simulations before constructing physical prototypes. The portal automates the handling of the required computing infrastructure such that companies do not need further service providers for executing their simulations. In this sense the portal promotes the innovative ability of SMEs.

The further portal development endeavour aims at facilitating cooperative processes, for example during product engineering. The idea is that several product developers work together on or with simulations in a federated sense. This means, simulation input and output data are stored and provided in a distributed fashion and are integrated for the joint development tasks. Thereby the portal ensures compliance with data sovereignty and information security concerns, such as confidentiality, integrity, authenticity, availability, or non-repudiation. This is a new type of business service that we combine with corresponding European research and development activities such as Gaia-X [11].

The overall endeavour pertains to the vision that particularly SMEs can easily make use of HPC systems, without needing to bother about architectural, implementation-related or platform specificities. An important basic requirement, however, is to understand how HPC use case stakeholders manage the corresponding data and product life cycles. The envisaged outcome is to leverage potentials for automation to facilitate access to HPC, particularly for SMEs.

Imagine a small engineering service provider designs a specific aircraft component and creates a three-dimensional computer model for it with standard computer-aided design (CAD) tools. To enhance the efficiency in collaborative engineering – in our vision – the engineering service provider can simply upload the aircraft component to the portal. The portal allocates the model to an appropriate HPC resource and integrates the aircraft component with the computer models of other aircraft parts which are relevant for the performance figures. After the simulation run, the performance figure is returned to the engineering service provider who then revises the component model. For uploading the revised model, the portal recognises that only parts of the computer model had been changed and transmits only the part with the relevant changes. This is an example of how the portal can enhance efficiency and cooperation in engineering services for SMEs.

A customary alternative solution today is the usage of standard computer simulation tools, like MATLAB Simulink on ordinary computing resources. The value proposition of SCALES is to be a tool for managed secure, reliable, and fast data exchange with higher ease of use than the solutions before. With SCALES, HPC users do not need to know software or hardware specificities of the used HPC resource. As an outlook, users do not even have to know, which HPC resource they must use, because the portal appropriately allocates data to available suitable HPC resources.

The idea is that engineers with a simulation use case log in to SCALES as a portal to HPC resources. In SCALES, they transmit required simulation information, like input data and possibly a standardised use case class description, e.g. simulation of a rotating part. Then SCALES allocates the simulation data to the appropriate sites, securely, reliable, and as fast as possible.

The core target user group of SCALES are small and medium-sized mechanical engineering providers with simulation use cases, such as engineering suppliers for automotive or aerospace OEM who want to validate the expected physical behaviour of their CAD models in complex assemblies. Early adopters may be automotive suppliers with respect to the transformation towards cooperative engineering in emerging digital ecosystems, using novel technologies like Gaia-X federation services.

There is much effort in several mechanical engineering industries, like manufacturing, automotive, aerospace, and agricultural engineering, towards federated cooperation in engineering tasks. Efforts can promote HPC adoption among SMEs, if HPC resources are integrated into the federated services and corresponding engineering processes. SCALES can enhance innovative capabilities by enabling more efficient development cycles in collaborative engineering tasks. Engineering providers can use HPC to validate and revise their CAD models in a highly realistic manner before delivery and before constructing a material prototype. We are aware of software development risks but, as a software developing software tool provider, we have the required experience and methods to keep these risks under control. One method for difficult technological streams with uncertainties is to first work on a small proof of concept (Minimum Viable Product-approach). On this basis then, a decision on how to proceed is made within an experienced software developer team, providing suggestions for alternative streams, too. Besides technological risks, there may be a risk of low adoption. We try to mitigate the risk by observing and evaluating ongoing developments in the industry and, of course, by back coupling with the partner HPC centres in the project.

7 Conclusions

The report described the essential systems analysis approach as general methodology for identifying requirements of HPC stakeholders with a focus on SMEs.

Important requirements pertain to

(a) IT security: The IT security requirements in engineering applications are particularly crucial for SMEs to ensure the protection of their market-relevant know-how to sustain in economic competition. The requirement is addressed by providing “SCALES” as a portal for secure and reliable data management for HPC in engineering applications.

(b) User friendliness: The requirement for ease of use shall mitigate thresholds for potential engineering partners to benefit from available HPC capacities in their engineering processes. However, for using HPC often very specific knowledge about the resources to use is needed. “Hiding” technical software, hardware, and architecture specificities of HPC resources in SCALES as an HPC access portal promotes the perceived utility of HPC for potential engineering partners, especially for SMEs.

(c) Facilitation of cooperation in engineering processes: The requirement to facilitate cooperative work reflects an inherent need in engineering processes which currently is not fully satisfied due to IT tool restrictions in connection with economic constraints. The IT tool restrictions mainly refer to a focus on bilateral data exchange in existing engineering tools. The economic constraints refer to the primacy of protecting engineering know-how in economic competition. The incorporation of novel emerging data federation technologies in HPC engineering ecosystems addresses both, the IT restrictions as well as the economic constraints.

8 References

[1] H. Haße. H. van der Valk, F. Möller, B. Otto, "Design Principles for Shared Digital Twins in Distributed Systems," Bus Inf Syst Eng, vol. 64, no. 6, pp. 751-772, 2022. DOI: 10.1007/s12599-022-00751-1.

[2] EXCELLERAT P2, "Success Story: Enabling High Performance Computing for Industry through a Data Exchange & Workflow Portal", online, https://www.excellerat.eu/success-story-enabling-high-performance-computing-for-industry-through-a-data-exchange-workflow-portal/, 20th December 2023.

[3] K. Pohl, "Requirements Engineering," dpunkt.verlag, Heidelberg, 2008.

[4] S. M. McMenamin, J. F. Palmer, "Strukturierte Systemanalyse," Prentice Hall, London, 1988.

[5] EXCELLERAT Services, "Industrial use cases", online, https://services.excellerat.eu/searchusecases, 20th December 2023.

[6] A. Reiberg, C. Niebel, P. Kraemer, „Was ist ein Datenraum? Definition des Konzeptes Datenraum," Gaia-X Hub Germany (Whitepaper), 2022.

[7] S. Scheider, F. Lauf, F. Möller, B. Otto, "A Reference Systems Architecture with Data Sovereignty for Human-Centric Data Ecosystems," Bus Inf Syst Eng, 2023. DOI: 10.1007/s12599-023-00816-9.

[8] B. Otto, "A Federated Infrastructure for European Data Spaces," Communications of the ACM, vol. 65, no. 4, pp. 44-45, 2022. DOI: 10.1145/3512341.

[9] Catena-X Automotive Network Consortium, "EDC – The Central Component," online, https://catena-x.net/en/offers/edc-the-central-component, 20th December 2023.

[10] EXCELLERAT Services, "Data Management Platform", online, https://services.excellerat.eu/viewcode/9, 20th December 2023.

[11] Gaia-X "A Federated and Secure Data Infrastructure", online, https://gaia-x.eu/what-is-gaia-x/about-gaia-x/, 20th December 2023.