



# FENIX

RESEARCH INFRASTRUCTURE

## D3.3

### Fenix User and Resource Management Service

<b>Work package:</b>	WP3 Technical specification and coordination	
<b>Author(s):</b>	Björn Hagemeier	JUELICH
<b>Reviewer #1</b>	Michele Carpené	CINECA
<b>Reviewer #2</b>	Javier Bartolomé	BSC
<b>Reviewer #3</b>	Stefano Gorini	CSCS
<b>Dissemination Level</b>	Public	
<b>Nature</b>	Other	

Date	Author	Comments	Version	Status
09.05.2019	Björn Hagemeier	Initial version	V0.1	Draft
29.05.2019	Björn Hagemeier	Added concept and specification documents	V0.2	Draft
06.09.2019	Björn Hagemeier	Incorporate updated timeline. Use ICEI template.	V0.3	Draft
22.04.2021	Björn Hagemeier	Updates: first Components deployed.	V0.4	Draft
18.06.2021	Björn Hagemeier	Answered reviewer feedback; updated "now" indicator in timeline.	V0.5	Draft
25.06.2021	Valentina Armuzza	Final editorial updates	V1.0	Final
25.02.2022	Björn Hagemeier	Revised version based on feedback at ICEI Intermediate Review 4 (RV4)	V1.1	Draft
14.03.2022	Björn Hagemeier	Updated version based on feedback from internal reviewers	V1.2	Draft
29.03.2022	Anne Nahm	Final editorial updates	V2.0	Final

The ICEI project has received funding from the European Union's Horizon 2020 research and innovation programme under the grant agreement No 800858.

© 2022 ICEI Consortium Partners. All rights reserved.



## Executive Summary

This updated version of D3.3 has been created after ICEI Intermediate Review 4 in November 2021 with the intention to have a more complete description of the developed solution. As compared to the previous version of the deliverable, the development process has advanced substantially. In Spring 2022, we are close to finalizing the development of the FURMS central instance. Similarly, all of the envisioned functionality is available for integration at the sites.

The central component, a web interface integrated with the central Fenix IdP and community and project management, has been deployed in February 2021 at JUELICH, the lead partner of this activity. Subsequent phases of development went according to plan. A training for site administrators to provide tools and documentation about the site-specific integration has been conducted in May 2021. This training marked the onset of site-local developments. Feature sets of each subsequent releases were chosen to always ensure a meaningful complete solution.

The final and complete implementation became available in February 2022, after which the contract entered the support and maintenance phase that will end with the end of the ICEI project in September 2023.

## Contents

Executive Summary.....	3
Acronyms.....	5
1. Introduction.....	8
2. FURMS components.....	8
2.1 Central FURMS instance.....	9
2.1.1 Views .....	9
2.1.2 SSH key management.....	10
2.1.3 User invitations.....	11
2.1.4 Fenix administrator view.....	11
2.1.5 Site administrator view.....	12
2.1.6 Community administrator view.....	13
2.1.7 Project administrator view.....	13
2.2 Site-local agents .....	14
2.2.1 Project provisioning and deprovisioning.....	14
2.2.2 User suspension .....	14
2.2.3 User SSH key (de)provisioning.....	14
2.2.4 Adding/removing user to project.....	14
2.2.5 Resource allocation to a project.....	14
2.2.6 Granting user access to allocation.....	14
2.2.7 Reporting resource consumption.....	14
2.2.8 Policy acceptance update.....	14
2.2.9 Ping agent.....	15
2.3 Message-oriented communication .....	15
3. Development.....	15
4. Procurement procedure and status .....	16
5. Deployment.....	16
6. Comparison to existing systems .....	17
6.1 EOSC.....	17
6.2 PRACE.....	17
6.3 Other activities .....	17
6.4 Local project management systems .....	17
7. Planned updates.....	18
8. Annexes.....	18

## Acronyms

AAI	Authentication and Authorization Infrastructure
ACD	Active Data Repositories
ACL	Access Control List
API	Application Programming Interface
ARD	Archival Data Repositories
BSC	Barcelona Supercomputing Center
CapEx	Capital Expenditure
CDP	Co-design Project
CEA	Commissariat à l'énergie atomique et aux énergies alternatives
CINECA	Consorzio Interuniversitario
CLI	Command Line Interface
CSCS	Centro Svizzero di Calcolo Scientifico
DL	Data Location Service
DM	Data Mover Service
DT	Data Transfer Service
FPA	Framework Partnership Agreement
FURMS	Fenix User and Resource Management Services
GoP	Group of Procurers
GUI	Graphical User Interface
HBP	Human Brain Project
HPAC	High Performance Analytics and Computing
HPC	High Performance Computing
HPDA	High Performance Data Analytics
HPST	High-Performance Storage Tier
IaaS	Infrastructure as a Service
IAC	Interactive Computing Services
ICCP	Interactive Computing Cloud Platform
ICEI	Interactive Computing E-Infrastructure for the Human Brain Project
ICN	Interactive Computing Node

IdP	Identity Provider
IPR	Intellectual Property Rights
JP	Joint Platform
JSC	Jülich Supercomputing Centre
LCST	Large-Capacity Storage Tier
MS	Monitoring Services
NDA	Non-Disclosure Agreement
NETE	External Interconnect
NETI	Internal Interconnect
NMC	Neuromorphic Computing
NVM	Non-Volatile Memory
NVRAM	Non-Volatile Random Access Memory
OIDC	OpenID Connect
OpEx	Operational Expenditure
PaaS	Platform as a Service
PCP	Pre-Commercial Procurement
PI	Principal Investigator
PID	Persistent Identifier
PIE	Public Information Event
PRACE	Partnership for Advanced Computing in Europe
Q&A	Questions and Answers
QoS	Quality of Service
R&D	Research & Development
R&I	Research & Innovation
RBAC	Role-Based Access Control
RFI	Request For Information
SCC	Scalable Computing Services
SGA	Specific Grant Agreement
SIB	Science & Infrastructure Board
SLA	Service Level Agreement
SP	Subproject
TCO	Total Cost of Ownership

TGCC	Très Grand Centre de calcul du CEA
UI	User Interface
US	User Support Services
VM	Virtual Machine Services

## 1. Introduction

The ICEI architecture foresees a central system to manage membership and roles of users within certain stakeholder groups. The system is also expected to allow for resource allocation and accounting within these groups. Additional aspects, such as the management of policy documents or central management of SSH keys must also be covered. What distinguishes this e-infrastructure service from prior approaches is the integration of an authentication and authorization infrastructure (AAI) and resource management in a central system that we call Fenix User and Resource Management Service (FURMS).

FURMS has been procured as a development service from an external supplier. The contract pertaining to this targets the development of the software implementing the central service and interfaces for site local components. The site local components facilitate the integration of participating sites with the Fenix infrastructure. They handle aspects such as user management, resource allocation, and accounting.

Support and maintenance for the central service as well as a template for the site local components is covered by the contract. The full implementation of the site local components must be performed by the individual sites. Sustainability of the software beyond the end of the ICEI project has been assessed as part of the evaluation of the bids.

Originally, the initial components of the FURMS software were scheduled to be available in M18. However, due to delays in the schedule the procurement process started only in M21. This document now finally reports about the availability of the first software component.

The remainder of this document is structured as follows. First, we'll introduce FURMS and the intended functionality, in particular the Web UI, site-local agents, and message oriented middleware, followed by a status update regarding the development, contract execution and deployment. Finally, as requested during the last review, we argue why existing systems did not provide the functionality that we intend for the Fenix infrastructure. Further details of the entire system can be found in the annexes.

## 2. FURMS components

At the logical level, FURMS comprises two components: the central FURMS instance and the site-local agents. Technically, the central instance can be broken down into the Unity identity management<sup>1</sup> and FURMS itself. However, for the purpose of this document, we consider both as a single entity. Additionally, there is a message broker through which the central FURMS instance and the site local agents communicate.

The general model of the FURMS system is that sites provide resources to the Fenix infrastructure, which are managed by FURMS. These resources are distributed to communities according to fixed ratios. These communities in turn distribute their resources to projects based upon a review process that is out of scope of FURMS. Figure

---

<sup>1</sup> <https://unity-idm.eu/>



1 captures this flow and includes the central IdP which governs the access to the FURMS system as well as those participating services that can actually make use of the IdP.

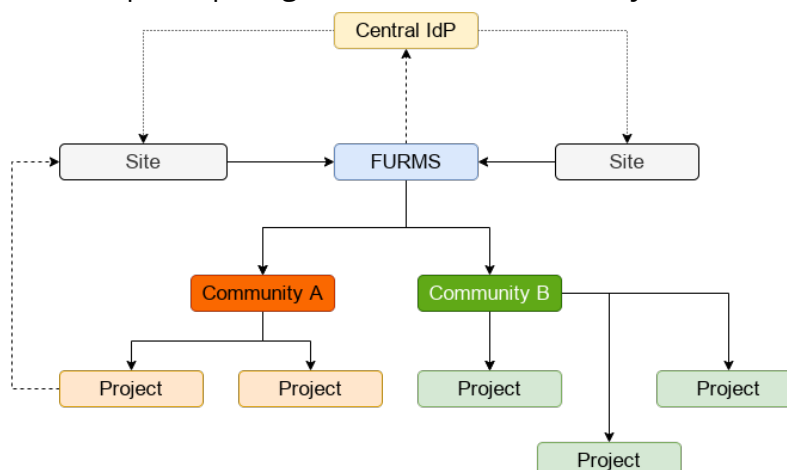


Figure 1: Users and administrators of FURMS authenticate via the central proxy IdP using their home or hosting organization's credentials. Participating sites provide resources to FURMS, which distributes them through communities and projects that use these resources. The consumption of resources at the sites is reported back to FURMS.

## 2.1 Central FURMS instance

The central FURMS instance keeps the intended state of the infrastructure and propagates it to the participating sites. It is represented by a web interface for users and administrators.

### 2.1.1 Views

The interface is centred around the roles that users may have within the infrastructure. Figure 2 displays the view-picker that lets users select the next perspective according to their roles and permissions. In the background, there is the dashboard for the Fenix administrator view. End users would typically only see their "User settings" view. Other available views are those for:

- Site administrators
- Community administrators
- Project administrators

In each of these, administrators at the respective level can configure resources pertaining to their scope. A full set of wireframes can be found in the appendix.

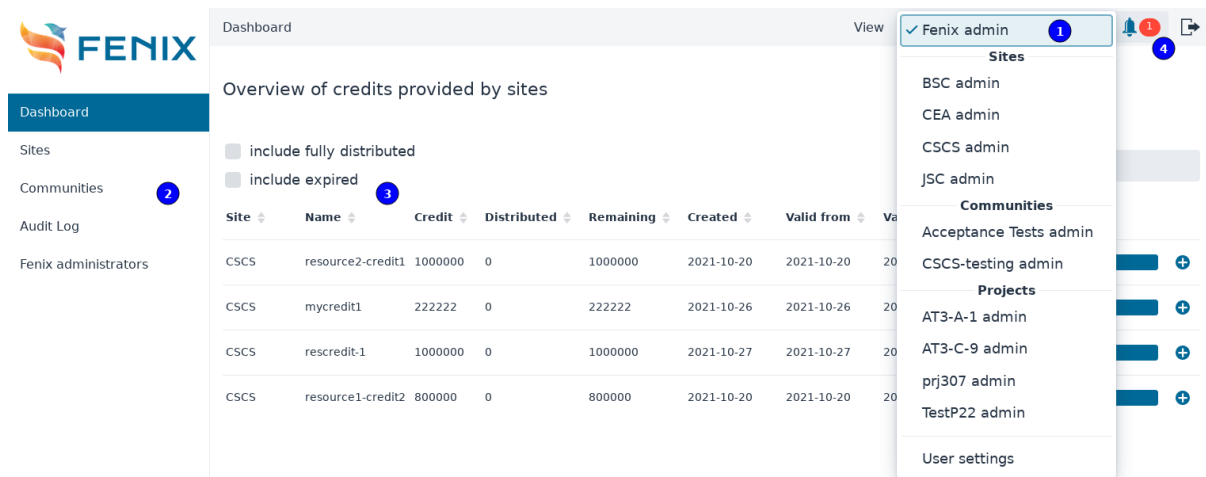


Figure 2: As an administrator of the Fenix infrastructure (cf. selected role (1)), one can see resources provided by the sites (3). Additional views are available for this role (2). Moreover, a notification system provides users with a direct link to pending actions (4).

### 2.1.2 SSH key management

SSH key management is part of the „User settings“ view, as shown in Figure 3. This is where users can register SSH keys to be distributed to the sites. Managing SSH keys in FURMS allows users to distribute SSH keys to all sites to which they have access from a single, central location. In particular, each site has a different local user interface through which SSH keys may be uploaded. Thus, FURMS improves usability by providing a single interface for all participating sites.

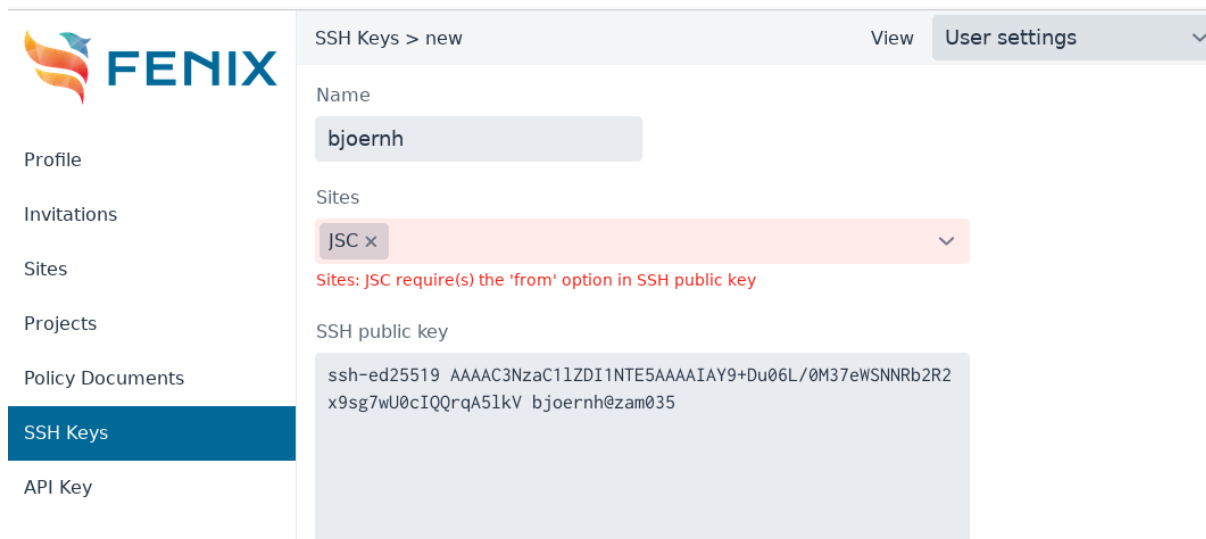


Figure 3: Users can register public SSH keys for SSH authentication at those sites where they have access. Specific requirements for these keys can be configured by the site administrators. The key will then be communicated to the site via the site-local agent.

Additionally, users can view their profile, respond to invitations, check sites and projects to which they have access, manage policy documents and create an API key to access the RESTful API.

### 2.1.3 User invitations

Users can be invited to the infrastructure already in a specific role, such that they can immediately start working in this role when accessing the system. An example of this is shown in Figure 4. Users who are already in the system can also be selected for their roles and receive a notification, as indicated by mark 4 of Figure 2. These invitations are available at each administrative level:

- Fenix infrastructure
- Community
- Project
- Site

Existing administrators can invite additional administrators to this role.

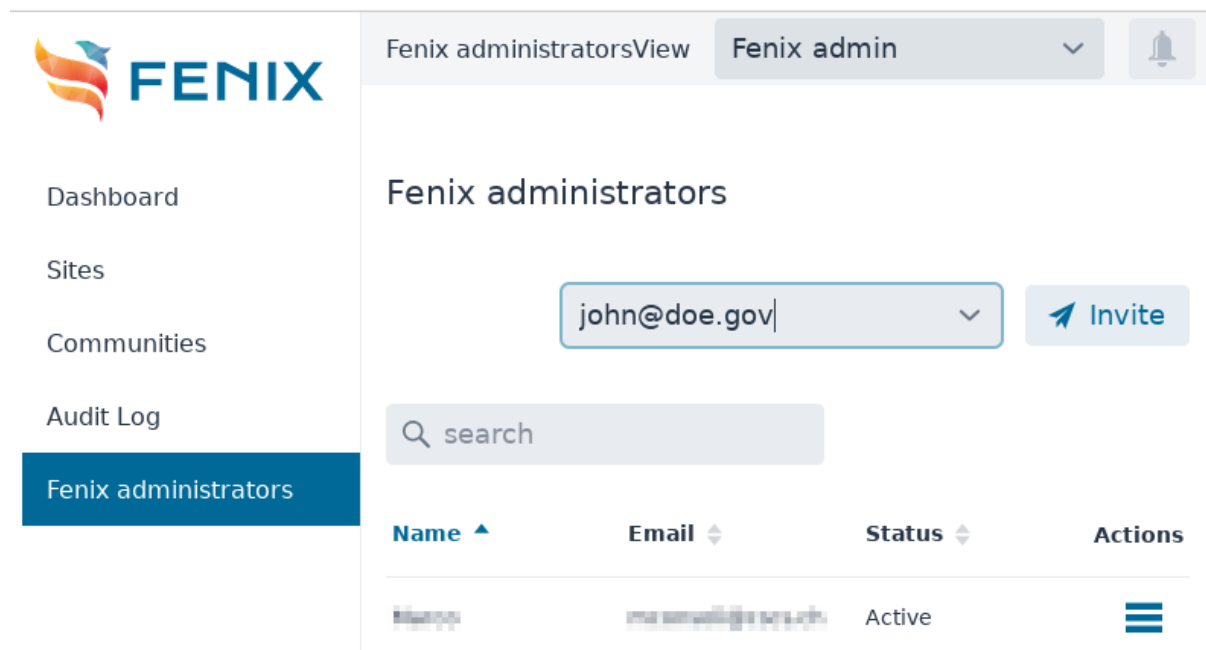


Figure 4: Users can be invited to certain roles in the system.

In the following, we will provide a summary of the remaining views of the FURMS UI. Further details can be found in the appendix.

### 2.1.4 Fenix administrator view

The Fenix administrator view allows to manage sites by creating them and assigning initial administrators who then take over the further management from the site administrator view. Fenix administrators also manage communities. Also in this case, initial community administrators will be assigned, who take over the management of the communities. Moreover, Fenix administrators can assign resources provided by sites to communities. Finally, the consumption of resources can be displayed.

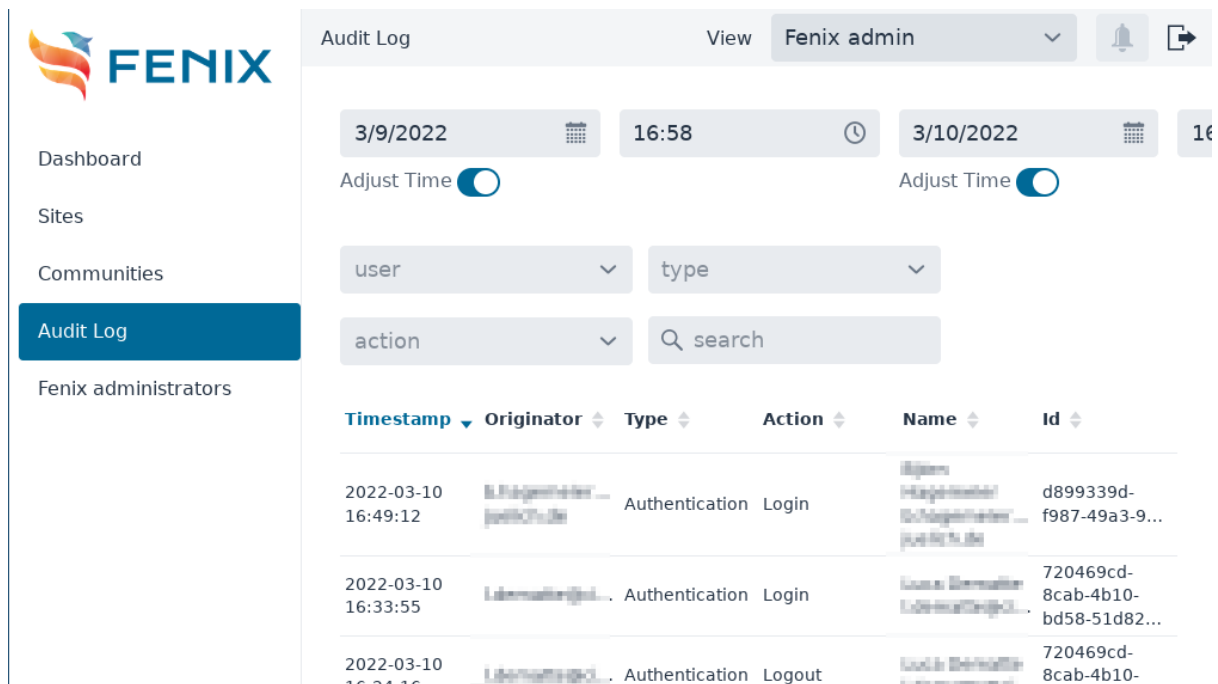


Figure 5: Fenix administrator view with the Audit Log selected.

### 2.1.5 Site administrator view

The site administrator view supports the management of policy documents, services, resource types, resource credits, site administrator management, and general settings. A special view of pending site actions allows to repeat certain site actions or cancel them. Policy documents can be linked to either the site or individual services. A policy is optional in either of these two cases. The resource model can best be understood by considering the respective UML diagram in annex 2.



Figure 6: Site administrator view with resource credits selected.

### 2.1.6 Community administrator view

The community administrator view provides a dashboard with an overview of resources allocated to projects, projects and groups views, community administrator management, as well as general community settings.

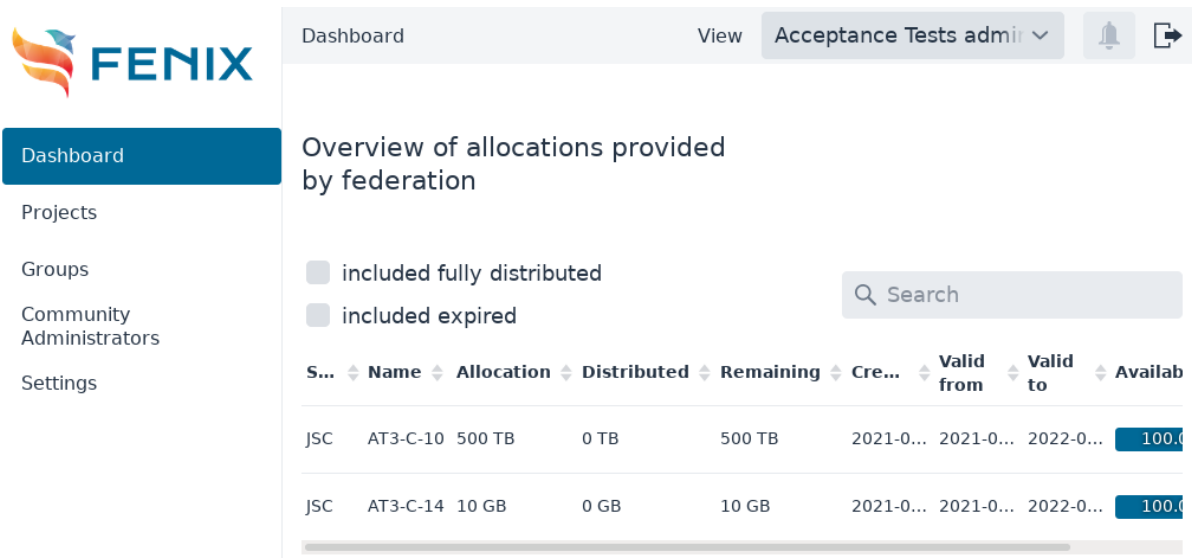


Figure 7: Community administrator view with the dashboard showing the available resources and their distribution to projects.

### 2.1.7 Project administrator view

The project administrator view provides information about project users and administrators, sites, resource allocations and access, resource consumption alarms, as well as general project settings. Through this, project administrators can add and remove users to the project and assign resources to them. This assignment will eventually be communicated to the site-local agents and translated to the site-specific user, project and resource management.

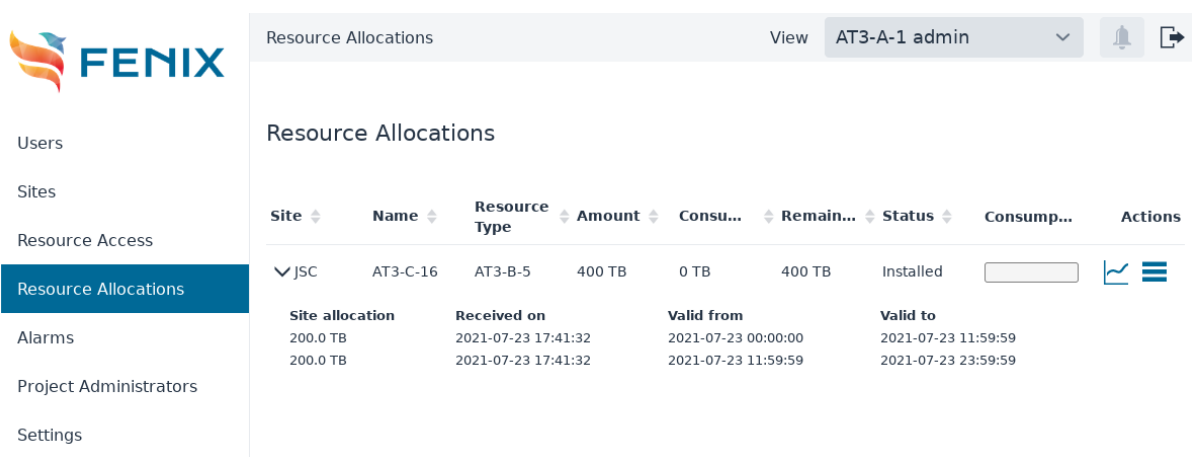


Figure 8: Project administrator view with available resources per site. The reported consumption can be viewed via the diagram icon on the right.

## 2.2 Site-local agents

The site-local agents are software components that translate the model of the central FURMS instance into site-specific actions as laid out in the following paragraphs. Communication is generally mediated through a message broker, allowing for asynchronous communication and decoupling the distributed components in cases of temporary failure.

### 2.2.1 Project provisioning and deprovisioning

Once a project has received resources at one of the participating sites for the first time at the FURMS level, this information must be communicated to that site. The site-local agent will then trigger the creation of a local project equipped with the resources as indicated by FURMS. Users who are members of the project at the FURMS level and have access to this site's resources must also be provisioned during the process.

### 2.2.2 User suspension

Certain services are detached from the web based authentication protocol through the central proxy IdP. This is the case for SSH based authentication. Therefore, additional measures must be taken to inform sites of user suspension, allowing sites to take appropriate action.

### 2.2.3 User SSH key (de)provisioning

It is difficult to authenticate SSH users via OIDC. We therefore decided to distribute SSH keys to the sites through FURMS.

### 2.2.4 Adding/removing user to project

These actions take place when a user is first granted access to project resources at a site or when the last of such grants is removed.

### 2.2.5 Resource allocation to a project

Once resources have been allocated to a project within FURMS, this must be communicated to the sites, which will then equip the corresponding local project with the same amount of resources.

### 2.2.6 Granting user access to allocation

In the FURMS model as well as most of the sites' models, project membership is decoupled from resource access within the project. Therefore, granting a user access to specific project resources is captured in this interaction.

### 2.2.7 Reporting resource consumption

Once resources have been consumed at a site, the site reports this back to FURMS. This allows project and community administrators to have an overview of resource consumption across the infrastructure.

### 2.2.8 Policy acceptance update

For legal purposes, most sites require users to sign or acknowledge policy documents. In order to save users from having to access site-specific interfaces, this process is handled within FURMS. Site administrators are able to define site and service specific policies. In

case of paper-based workflows for policies, there is a specific site staff role for staff members of the sites who receive and acknowledge such policy documents and enter the information in FURMS.

### 2.2.9 Ping agent

Through this action, site administrators can trigger a test of the connection between the central FURMS instance and the site local agent. This check is mediated through the central message broker and thus represents a check of the full interaction as required by all other interactions above.

## 2.3 Message-oriented communication

A message-oriented communication architecture has been chosen due to the inherently distributed nature of the entire system. It decouples the components and ensures message delivery despite potential intermittent outages of the communication partners. Particularly long-running operations benefit from the asynchronous communication patterns offered by this approach.

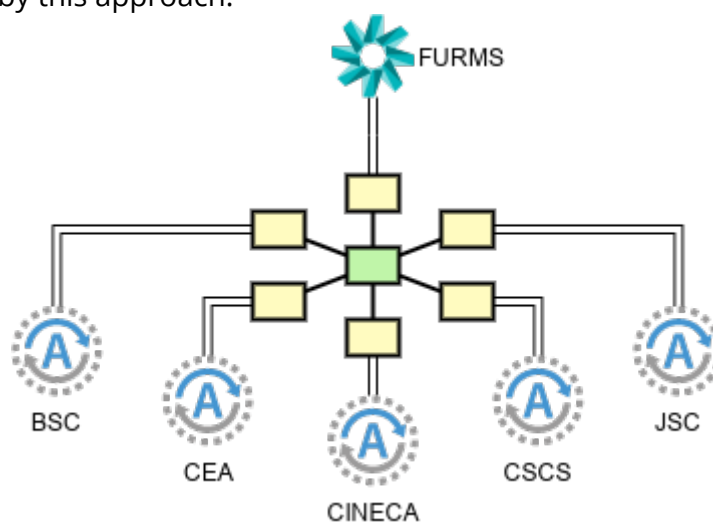


Figure 9: The central FURMS instance and site-local agents communicate via message-oriented middleware.

Due to common requests by developers of the site-local agents, FURMS additionally offers a RESTful API. The RESTful API is also used by the central IdP when querying additional attribute from FURMS as part of an ongoing authentication.

## 3. Development

The development of the FURMS system has largely been procured as a research and development service (cf. Section 4). The contract comprises the development of the central FURMS instance and APIs for communication with site-local agents. Upon finalizing the design and implementation of these, the contract transitions into a maintenance phase until the end of the ICEI project in September 2023.

Site-local agents are very specific for each site participating in the federation, such that the development contract only comprises an example agent, from which developers of the site-local agents can start. It is the responsibility of each participating site to develop their own site-local agent.

## 4. Procurement procedure and status

The project is currently in the status of executing the contract awarded to the successful bidder in the procedure. The process thus far comprises the following steps, including a comparison to the timeline communicated in the first version of this deliverable.

All contractually agreed deadlines have so far been met by the contractor. We have captured this additional information in this document as part of the update requested after the interim review in November 2021.

Planned	Actual	
September 2019	October 2019	Final version of tender documents
October 2019	December 2019	Publication of tender
December 2019	January 2020	Pre-selection of bidders (cancellation of procedure)
October 2019	February 2020	2 <sup>nd</sup> publication of tender
December 2019	April 2020	Pre-selection of bidders
March 2020	May 2020	Negotiations
May 2020	June 2020	Final bids
June 2020	July 2020	Contract negotiations and awarding
September 2020	February 2021	First components available (release 1.0.0)
May 2021	May 2021	FURMS release 2.0.0
September 2021	October 2021	FURMS release 3.0.0 and 3.0.1
January 2022	January 2022	FURMS release 4.0.0

The availability of the first components has been a matter of negotiation between the public procurer and the supplier, which took place prior to the awarding of the contract. Our expectation was that the supplier would provide initial components within up to six months after awarding the contract. This was postponed to February 2021 in favour of more intense discussions about details of the software design. This design phase was finished in December 2020 and the development commenced accordingly.

Detailed information about the envisioned timeline for the developments and execution of the contract can be found in the annex 3.

## 5. Deployment

The lead partner for this procurement is JUELICH. Therefore, the initial components of FURMS (release 1.0.0) have been deployed as a test and integration instance there<sup>2</sup>. This setup will remain at least for the duration of the ICEI project. A production environment<sup>3</sup> has been deployed starting with release 2.0.0.

The site-local agents have since been deployed at each of the sites and connected to the integration instance for development. As a matter of fact, due to easier handling of

<sup>2</sup> <https://furms-integration.fz-juelich.de/>

<sup>3</sup> <https://furms.fz-juelich.de/>



resetting the distributed common states among the central instance and site-local agents during development, central instances have been deployed per site. A central instance for production has been deployed and can be used by each of the sites once their site-local agents have been finally developed.

## 6. Comparison to existing systems

While there are known previous attempts to achieve functionality similar to FURMS, none of these solutions fully matched the requirements of the Fenix infrastructure

### 6.1 EOSC

Whereas at first glance EUDAT's Data Project Management Tool (DPMT) seems to aim in a similar direction as FURMS, it is more of a service catalogue matching user demand and service offerings. No direct resource distribution at local resource providers is supported, which is an important need covered by FURMS.

### 6.2 PRACE

While PRACE relies partially on a common authentication system based on X.509 certificates, it does not have a central system in place to facilitate the distribution of available resources to projects at the sites. To this end, it uses the established systems at each local site independently.

### 6.3 Other activities

We are aware of comparable activities that were in development in parallel to the FURMS development. One example is the project management tool of the German Network for Bioinformatics Infrastructure (de.NBI) targeted at provisioning of cloud projects at sites. Information about the details of the tool is scarce. In comparison to FURMS, this tool seems to be limited in scope, too.

Finally, the Helmholtz Infrastructure for IT Services (HIFIS) is developing a tool to distribute resources to projects at participating sites. This activity started only after the FURMS development. Discussions with stakeholders showed that there is some overlap and at least ideas from FURMS may be used there as well. It is too early to say whether a full adoption would be advised.

### 6.4 Local project management systems

Each of the participating sites has a long history of providing HPC and storage resources. They all have their own approaches of managing system access in projects, often depending on the surrounding IT infrastructure. There is no single solution that fits all of these purposes and could easily be transferred among sites. Most of the systems do not support federated AAI out of the box. Also, we have found most of the local systems to have evolved over time and providing separate interfaces for separate purposes. At some sites, the full functionality of the FURMS system is even distributed across several divisions within the organization. For this reason, we have opted to work with an integration layer, the FURMS site-local agent, to map the functionality at each of the sites.

## 7. Planned updates

No updates of this deliverable have been planned.

## 8. Annexes

The annexes comprise documents that have been provided by the contractor after the design phase. They are attached for informational purposes to support the illustration of the current state of developments and envisioned final product. Specifically, they are:

ID	Annex	Description
1	Concept of the interaction of FURMS with sites	An overview document describing the interaction between the central FURMS instance and each of the site-local agents.
2	Concept of the interaction with the central IdP	A document describing the interaction of the central FURMS instance with the central identity provider.
3	Detailed plan for the timeline and deliverables	A more detailed plan of which features will be available when and which requirements will be covered by them.
4	Resource planning for the development	Information about the contractor's staffing.
5	FURMS software architecture	A general overview of the architecture and relations among components.
6	FURMS wireframes	Initial wireframes to support discussions about what FURMS should look like in the end.
7	FURMS REST API reference	The reference for the REST API, which can be used by developers at the sites to query information from FURMS.