FENIX
RESEARCH INFRASTRUCTURE

# FURMS

## Fenix User and Resource Management Service

Björn Hagemeier

2022-10-19

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

# Overview

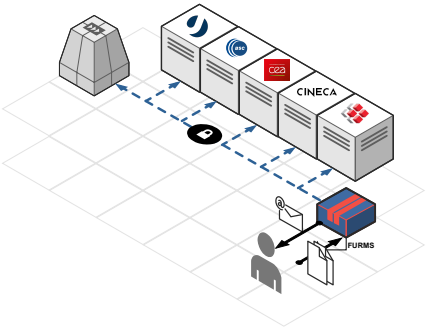
- Introduction
- Status of development
  - FURMS
  - site-local agent
- Site-local agent
- Deployment
- Roles
  - User
  - Fenix administrator
  - Project administrator
  - Site administrator
  - Site staff
- Commandline interface
- Example workflows

FENIX RI

# Introduction

**F**enix **U**ser and **R**esource **M**anagement **S**ervice

- One-stop-shop for users and administrators of the federated infrastructure
  - Manage community and project membership
  - Manage resource allocations for communities and projects
  - SSH key management
  - Policy documents incl. paper-based workflows
- Accounting
  - Ingest accounting information from sites
  - Visualize consumed budgets
- Logic
  - Budget notifications
  - Site integration




FENIX RI

# Stakeholders

**Community**

- A (virtual) organization of scientists that is entitled to use a given fraction of resource available within the Fenix infrastructure
- Initially HBP (25%) and PRACE (15%)

**Project**

- Communities create projects to which they allocate resources at their own discretion
- Peer-review process should be established

**User**

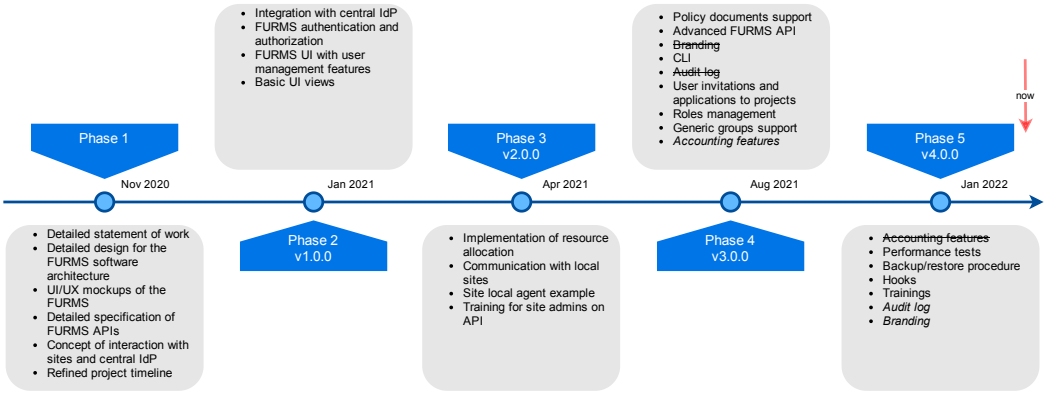
- Users are associated with communities and projects
- Privileged users will have special permissions to manage the Fenix infrastructure

**Site**

- 5 initial sites: BSC, CEA, CINECA, CSCS, JUELICH
- Sites contribute their resources to Fenix
- Accounting information for consumed Fenix credits sent to FURMS

FENIX RI

# Status and timeline




FENIX RI

# Site-local agent

- Disparate user and resource management at each site, grown over many years and very specific
  - Requires a mapping at each local site
  - Each site is responsible to implement their own agent
  - Template agent has been provided by developer of FURMS solution
  - New sites will have to implement their own agent as part of the onboarding process
- Status
  - BSC → *production*
  - CEA → development
  - CINECA → pre-production
  - CSCS → *production*
  - JSC → pre-production

FENIX RI

# Site-local agent

**Actions and communication**

Actions

1. Project provisioning and deprovisioning
2. User suspension
3. SSH key provisioning and deprovisioning
4. Adding and removing users to and from projects
5. Resource allocation to projects
6. Granting users access to allocations
7. Reporting resource consumption (accounting)
8. Policy acceptance
9. Ping agent

Communication

- Message broker (RabbitMQ)
- Dedicated queues per site
- Broker permissions set to prevent interference with other sites

FENIX RI

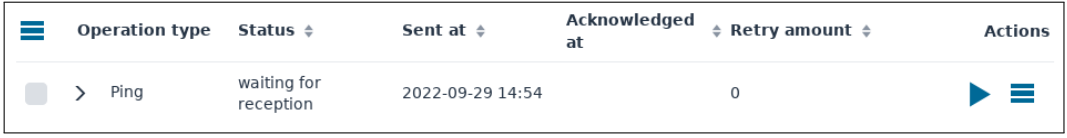
# Asynchronous communication

- message exchanges trigger potentially long-running operations
- communication can be triggered by either party
- polling may be inappropriate
  - causing unnecessary load
  - no timely propagation of information
- unfit for synchronous request/response patterns
- messaging based approach
  - supported by `messageCorrelationId`
  - immediate responses or acknowledgements
  - protocol versioning supported

FENIX RI

# Asynchronous communication

- message exchanges trigger potentially long-running operations
- communication can be triggered by either party
- polling may be inappropriate
  - causing unnecessary load
  - no timely propagation of information
- unfit for synchronous request/response patterns
- messaging based approach
  - supported by `messageCorrelationId`
  - immediate responses or acknowledgements
  - protocol versioning supported



| | Operation type | Status | Sent at | Acknowledged at | Retry amount | Actions |
|---|---|---|---|---|---|---|
| ☐ | Ping | waiting for reception | 2022-09-29 14:54 | | 0 | |

FENIX RI

# Deployment

Deployed in two separate environments

Staging environment

- development and integration testing
- 1 central instance for all sites and site-local agent developers
- +1 instance per site and site-local agent developers
- easy to revert baseline state per site (useful for development)
- connected to Fenix AAI acceptance environment

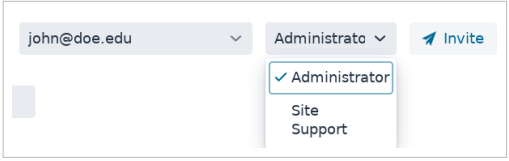
Production environment

- RabbitMQ: secure connection and only accessible from participating sites' IP ranges
- Hosted in highly-available VMWare cluster at JSC
- connected to Fenix AAI
- 4 daily backups
- `https://furms.fz-juelich.de/`

FENIX RI

# User onboarding

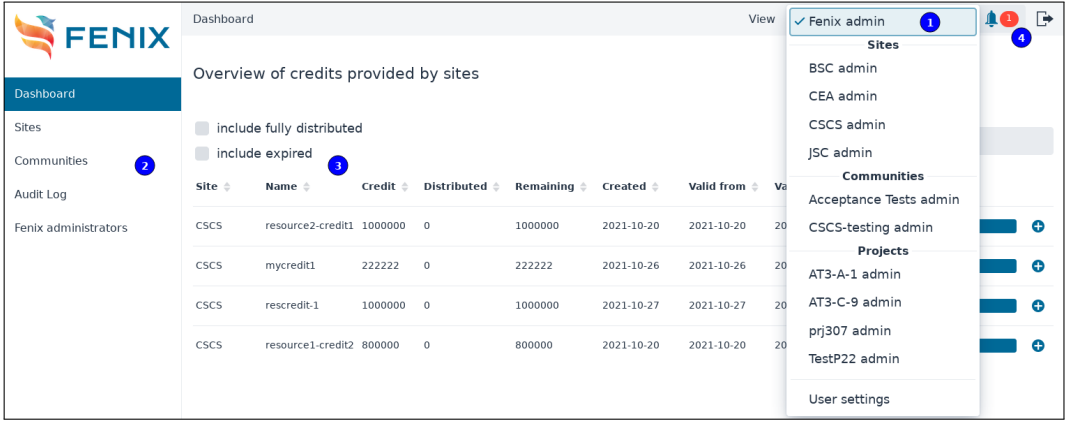
- Invitations to users can be sent by email
- Users can login using an account from any of the participating sites
- Users already known to the system can typically be found within the same context



FENIX RI

# User interface

## Generic layout



# Roles and responsibilities

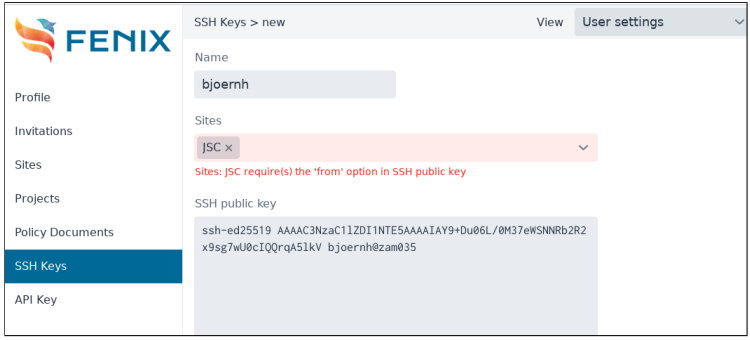
# Roles and responsibilities

**Overview**

- Users (all)
  - Register
  - Apply for community or project roles
  - Confirm compliance with infrastructures and site rules
  - SSH key management
- Fenix administrator
  - Setup communities
  - Grant permissions to community managers
- Community administrator
  - Setup projects
  - Grant permissions to project managers
- Project administrator
  - Grant permissions to project users
- Site administrators
  - Policy documents
  - Services
  - Resources
  - Other site administrators
  - Settings
- Site support
  - Record policy acceptance in paper-based policy workflow

FENIX RI

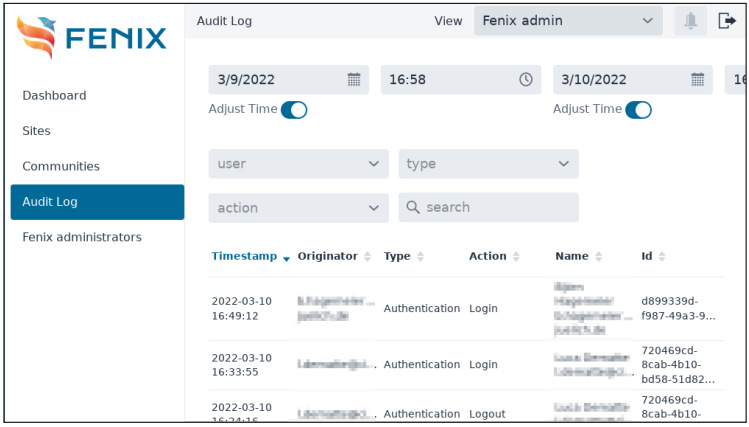
# User




- Invitations
- Site connection info
- Projects
- Policies
- SSH keys
- API key (really useful only for administrators)

FENIX RI

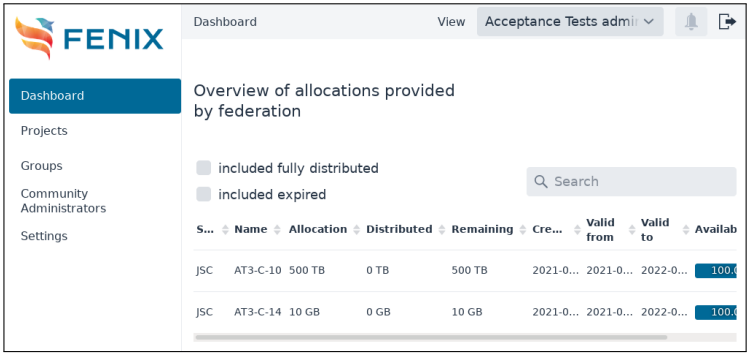
# Fenix administrator




- allocate resources to communities
- manage sites, communities, and federation
- access to audit log

FENIX RI

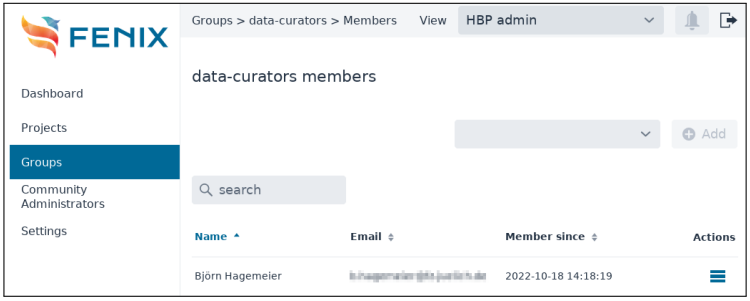
# Community administrator




- manage projects and groups
- resource allocation

FENIX RI

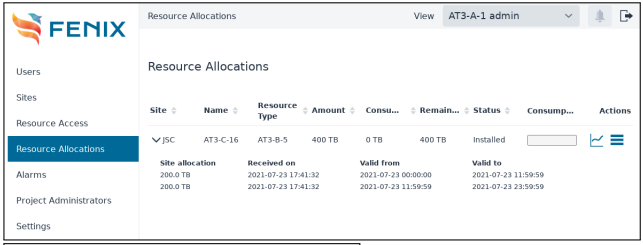
# Groups




- intended for cross-project groups
- no allocation of resources, but potential access to services
- currently no use-case realized

FENIX RI

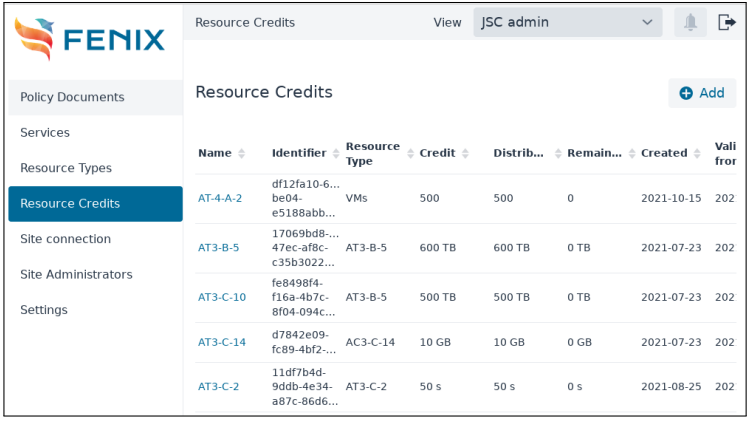
# Project administrator



- view allocations and consumption
- alarm management
- further administrators
- user and access management

FENIX RI

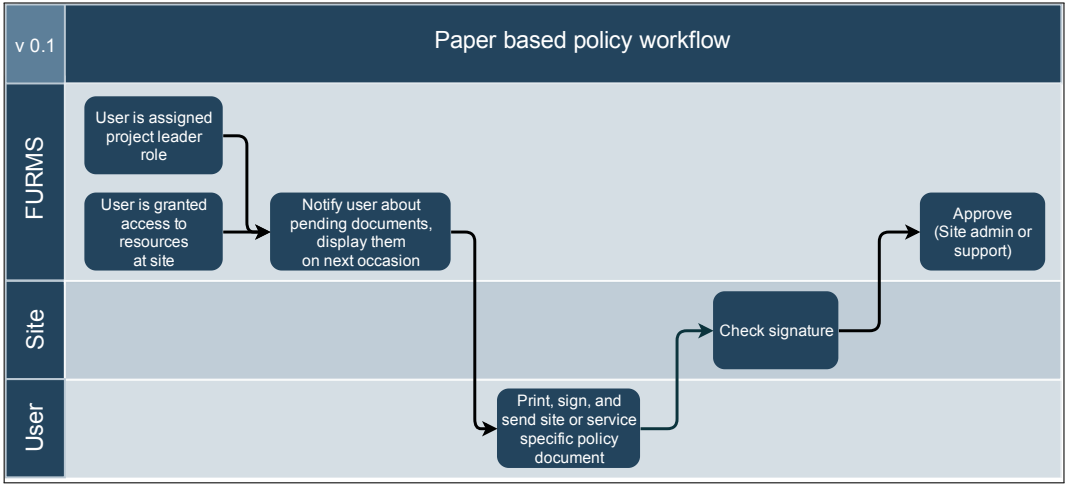
# Site administrator




- define and register available resources and services
- manage policy documents linked to site and services
- test agent connection
- general settings

FENIX RI

# Site support

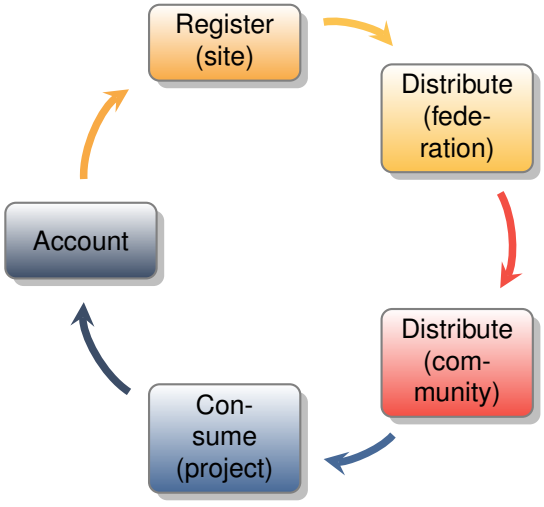



FENIX RI

# FLOW OF EVENTS (EXAMPLES)

# Resource allocation

1. sites register resources
2. Fenix distributes resources to communities
3. communities split resources among their projects after peer review
4. users consume resources of their respective projects at sites
5. sites send accounting information to Fenix




FENIX RI

# User invitation

1. administrator invites a user (via email)
2. user responds by following temporary link
   - optional registration
   - registered users also receive notification in web interface
   - accept invitation
3. administrator is notified about successful invitation

FENIX RI

# Project provisioning

**Definition**

- **FURMS project:** project at federation level, may or may not have an expression at sites
- **local project:** project at site instantiated from FURMS project

Site resources added to project:

- notification of site
- if existing local project
  1. add resources to local project
- if new local project
  1. create local project
  2. add resources to project
  3. add users of FURMS project to local project according to resource access

FENIX RI

# CLI

- a CLI is available to access the REST API
- authentication through user id and API key
- predominantly for admin access

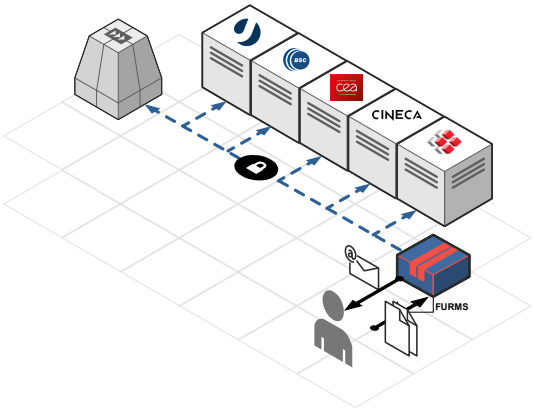
```
$ furms site show
↪  cee47020-d6d5-4b25-ab68-78642f1bc428
{
  "id" :
  ↪  "cee47020-d6d5-4b25-ab68-78642f1bc428",
  "name" : "JSC",
  "sitePolicyId" : null,
  "resourceCredits" : [ ],
  "resourceTypes" : [ ],
  "services" : [ ],
  "policies" : [ ]
}
```

FENIX RI

# SUMMARY

# Summary

- Introduction
- Status of development
- Site-local agent
- Deployment
- Roles and responsibilities
- Example workflows



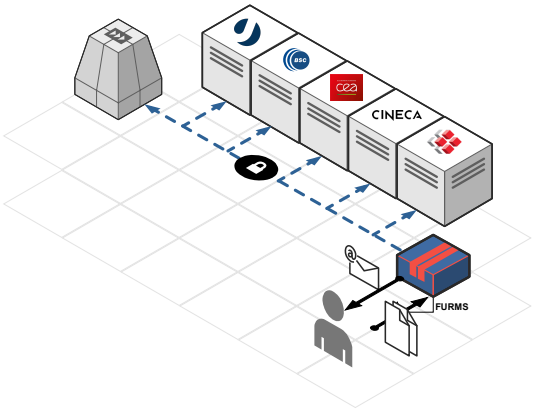

Pointers

- FURMS: `https://furms.fz-juelich.de/`
- Documentation: `https://unity-idm.github.io/furms/`

FENIX RI

# Summary

- Introduction
- Status of development
- Site-local agent
- Deployment
- Roles and responsibilities
- Example workflows




Pointers

- FURMS: `https://furms.fz-juelich.de/`
- Documentation: `https://unity-idm.github.io/furms/`

THANK YOU

FENIX RI