NORDIC E-INFRASTRUCTURE COLLABORATION

PUHURI Deliverable D2

Project Title:	Puhuri
Public	This document is PUBLIC
Deliverable title:	Implementation of Puhuri core functionalities
Contractual delivery date:	30th of March 2021
Status	v. 1.0
Actual delivery date:	15th of April 2021
Updated	15th of April 2021
Partner(s) contributing to this deliverable:	FI, SE, EE, NO

Authors and Contributors:

Marina Adomeit, Kent Engström, Kalle Happonen, Jarno Laitinen, Ilja Livenson, Juha Nyholm, Ahti Saar and Anders Sjöström

Table of Contents

The scope of this deliverable

The first Puhuri deliverable [D1] discussed the requirements collection for the Puhuri Project. This deliverable of Puhuri project describes implementation of the Puhuri core functionalities for the authentication, authorisation and resource allocation.

This deliverable gives information especially for the service providers and for the national portal integration to Puhuri services:.

Chapter 1 summarises the use cases and the high level architecture of the technical components.

Chapter 2 describes the authentication and authorisation infrastructure services. User Registration belongs to the Common AAI services operated by GEANT eduTEAMS. Puhuri is an Infrastructure Service Domain (ISD) using the Common AAI called researcher-access.org.

Chapter 3 describes Puhuri Core storing the resource allocation information. It provides programmable interfaces (APIs) for the portals, which implement the user interfaces for the users and resource allocators.

Chapter 4 shows how CSC as a Resource Provider is integrating to Puhuri Core and Puhuri AAI.

LUMI supercomputer's [LUM] integration to Puhuri is the priority use case, but the same services can be used for other use cases. LUMI user accounts are created into LUMI before the Users will login.

Deliverable 3, to be written in Q3/2021, will report on the LUMI integration and national portal integration by SNIC in more detailed manner than in this document. Later also accounting information of LUMI usage will be send to Puhuri Core that the portals can make the reports for the projects. The deliverable 4, in December 2021, will report on the accounting implementation.

1 Puhuri high level architecture and use cases

1.1 High level architecture

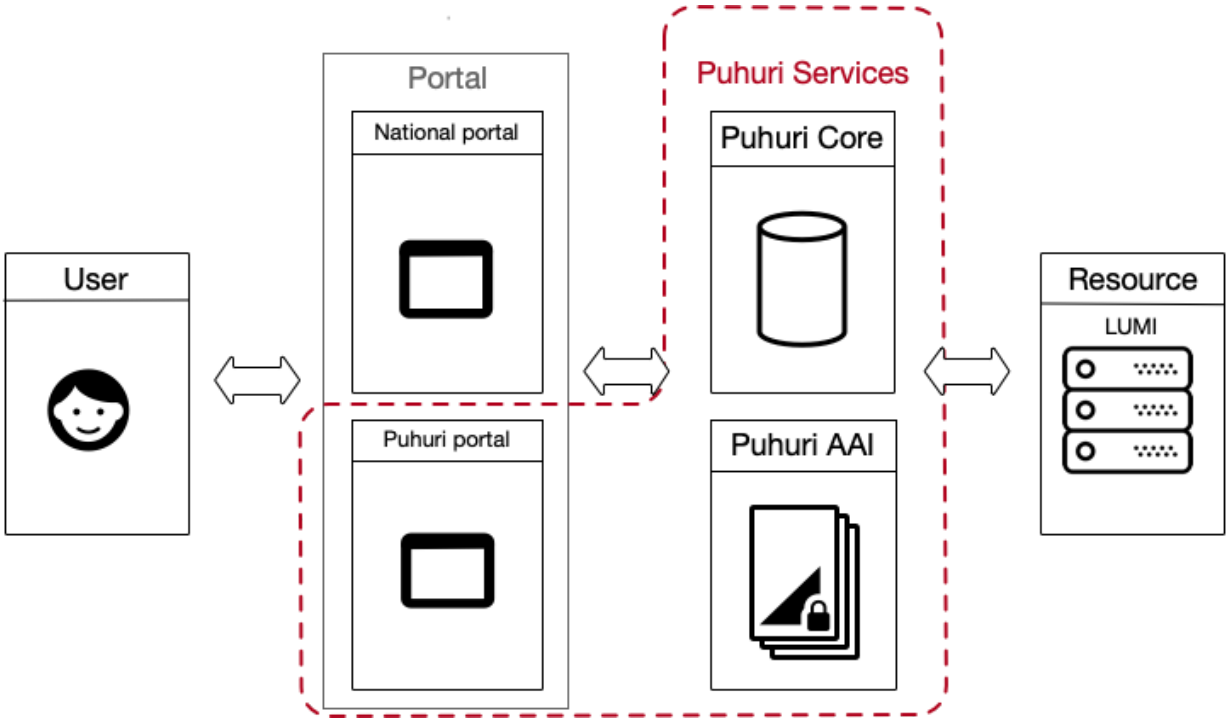

Figure 1. Puhuri's high level architecture.

Figure 1. High level architecture of Puhuri. User either uses a National portal or an instance of a Puhuri Portal for the Project Application and the Project membership management.

The end user is using Puhuri AAI to register Puhuri User Account and to login (authenticate) to the Resources. Via Puhuri Portal or a National Portal, depending which solution is available for User, they apply for a Resource allocation and then manage the Members of the Project (see Chapter 3).

Puhuri Core is the database storing Projects and their Members as well as Allocation, Accounting and related Resource information. A Resource provider (in Figure 1 LUMI) uses APIs to import the active allocation and project membership information from the Puhuri Core.

The Resources receive User related information via Puhuri AAI from User's Identity provider (IdP), when a User authenticates using home organisation's credentials. Chapter 2 covers the implementation.

1.2 User Registration workflow

All Project Members must register in Puhuri AAI to use Resources connected with Puhuri, in particular LUMI. The Registration creates a unique identifier (Community Unique Identifier, CUID) for the User, which is used for referencing and linking user identity across the different components.

The identity provider releases the attributes about User's identity and affiliation, which is important for the resource providers to know. Furthemore, Level of Assurance of such a data is provided by Puhuri AAI. User can also provide additional information such as phone number or SSH public keys. Those may be optional to give if the Resource Provider does not require those. User can define other email address than what was returned from the Idp. That email will then be verified.

User may also accept Terms of Use document (ToU) and the User also needs to accept the Acceptable Use Policy (AUP) document of the Puhuri AAI services.

The national portal may implement identity linking of local accounts with Puhuri AAI account. . The recommended way for that is to use OIDC registration flow, which can be triggered from the national portal and lead to redirection back with a created Puhuri User CUID

1.3 Project allocation

The Resource application review process happens outside of Puhuri. The national portals are expected to push approved projects and resources to Puhuri Core. The Principal Investigator (PI) creates project in the Puhuri Portal or National Portal. The submission is then accepted by the Resource Allocator, which is responsible for the particular application's resource granting. Then the project information is propagated to Puhuri Core and from there synchronised to the Resource Provider. In addition to Project description and members, also the Resource-specific resource allocation attributes are included. The detailed description is in Chapter 3.

The Principal Investigator can also delegate co-Principal Investigator (proxy-PI) roles for the Members of the Project. These PIs can invite or remove the Members of the Project. How that is done, depends on the implementation of the portal in question. Essential is that the Members must have registered the Puhuri AAI as well as all membership information relies on the Puhuri AAI CUIDs.

1.4 Other workflows

There are various other workflows related to the User and Project lifetime management. For example a Project Member may change organisation and thus there has to be a processes, how to react.

The Resource Provider may communicate directly with the User and have specific processes. Resource may define special resource allocation processes e.g. to apply for storage quotas.

User should be able to view own personal information and possible change the information collected at the Registration portal. Also Project related information such as accounting information will need to be shown. Accounting information flows from the Resource to Puhuri Core and then to the portals.

1.5 LUMI Use Case

EuroHPC Joint Undertaking (JU) is acquiring pre-exascale and petascale supercomputers (the EuroHPC supercomputers) which will be located at and operated by supercomputing centres in the Union. LUMI [LUMI] is one of those. It will be hosted at CSC in Kajaani data center. LUMI is the main use case for Puhuri and thus the schedule of the Puhuri project is steered according to that. The LUMI consortium involves in addition to Puhuri member countries also Belgium, Czech Republic, Poland and Switzerland. These countries may integrate their national portal to Puhuri or use Puhuri provided portal to manage the LUMI projects. The resource application review process of the call is out of Puhuri's scope. The LUMI integration is described in Chapter 4.

2 AAI implementation

2.1 AAI Architecture with Infrastructure Services Domains

Puhuri AAI is one of the essential Puhuri Service components as it enables Users to securely access to Resources such as LUMI. Draft of Puhuri AAI, which is compliant with AARC blueprint architecture, was presented in Deliverable 1 [D1]. Since then, the collaboration between Puhuri and Fenix in order to design an AAI solution for the benefit of both communities, led to further evolve this architecture.

HPC data centres are in the process of transforming to infrastructure service providers. The service portfolio will be diverse and based on different types of resources for computing, data processing, storage and networking, including HPC systems. Making these infrastructure services available to scientists and engineers at European scale, they need to be integrated into a shared AAI. In order to facilitate integration, we assume these infrastructure services to become available in different policy domains, which we call *Infrastructure Services Domains (ISD)*.

The motivation to pursue a common AAI solution for European ISDs (in subsequent text AAI solution) is two-fold:

- In their line of work, researchers may need to use services from different ISDs. There is a strong reasoning to make this experience as smooth as possible for the researchers and streamline their access to services. Ideally, the users should register once for use of these services and be able to request allocations in different ISDs according to the resource allocation mechanisms available within a given ISD.
- ISDs have very similar AAI requirements that are streamlined by adoption of AARC BPA which is widely adopted by ISDs. Therefore, it is reasonable to attempt to limit the proliferation of unique AAI solutions for achieving cost and resources efficiency. This as well reduces the risk of failing to find expertise for running AAI solutions as those are already sparse and difficult to find.

The concept of ISDs allows to simplify integration by allowing for ISD-specific policies and decentralised decision making, while keeping the option for maximising alignment such that the user is exposed to this substructure as little as possible.

Figure 2 shows the proposed AAI solution in the example of integrating the current ICEI (an implementation project of Fenix) and Puhuri/LUMI ISDs. The solution is based on the AARC architecture and follows its guiding principles.

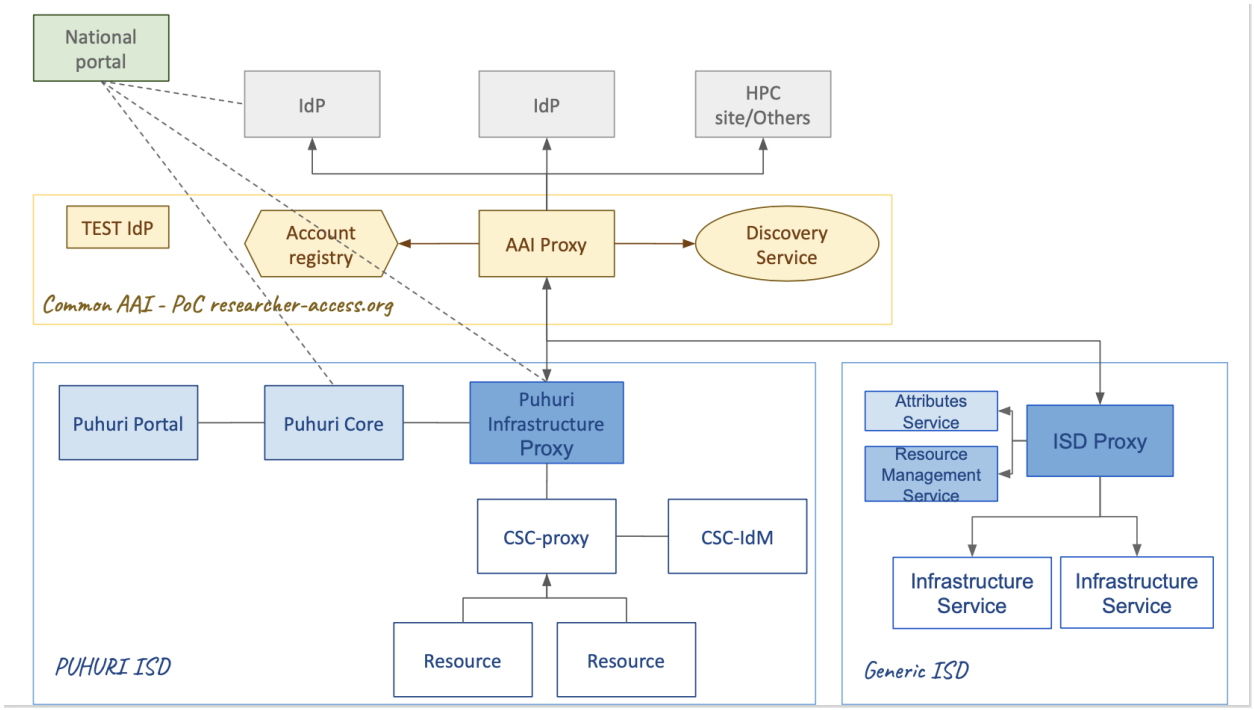

Figure 2: Proposed AAI solution

An Infrastructure Service can be part of several ISDs, but the Service needs to integrate to the ISD-specific resource management systems and also possibly implement ISD-specific support on the Infrastructure Service. Such domain specific features include attributes specific to a given service and the resource management policies governing roles e.g. user and project (resource) access policies.

The components of the proposed AAI solution and their brief description is given in the following:

- **IdPs**
  This layer contains all connected IdPs, such as IdPs published in the eduGAIN, HPC sites IdPs etc.
- **Common AAI Services**
  These represent a set of common identity services that would be used by all ISDs, as a multitenant service. This services would be delivered by GEANT using its eduTEAMS platform. Components of AAI Services is as follows:
    - **AAI** acts as SP-IdP proxy, integrating IdPs on the north side and ISD proxies on the south side. AAI also comprises token translation services which translate identity tokens between different technologies. The AAI may apply access policies defined by the connected ISDs (e.g. from which IdPs users are allowed to connect to a given ISDs)
    - **Discovery Service** enables users to select the IdP they wish to authenticate with. The Discovery service supports IdP filtering/hinting functionality so that ISDs can define which IdPs are applicable and should be visible for their context.

- **Account Registry** - enables user registration and creation of a central user profile with one unique identifier. It presents users with a privacy notice which allows users to be informed regarding processing of their personal data and the acceptable usage policy for Common AAI Services.

- **Infrastructure Services Domain**
  - **Infrastructure Services Domain Proxy**
    These are in the responsibility domain of an ISD. The ISD Proxy is a single integration point for all End Services in the context of the given ISD. It is currently planned to be an SATOSA Idp Proxy provided by GEANT. It also provides the authorisation layer services and possibility to present ISD specific acceptable usage policy if any. The ISD proxy is a place where the access policies for related ISD are implemented. While the Common AAI Service presents the identity layer used to identify users and collect user attributes, particular access decisions based on user attributes and other factors are made on the ISD proxy level.
  - **Attribute services (Puhuri Core)**
    Service that provides user attributes. Typically, this information includes community project/group memberships and roles and implements processes for onboarding users assigning roles to them. The semantic and syntax interoperability across the ISDs attributes will be addressed later.
  - **Resource Management Services (Puhuri Portal)**
    These services are needed for managing and providing resource allocation information within a given ISD.
  - **End Services (Resources)**
    These are the infrastructure services that a user can access. In the case of Puhuri, these are connected via CSC Proxy, and use CSC IdM as local registry for user accounts necessary to provision services.

2.2 Current status of AAI deployment Technical and contractual integration of Identity providers and Services

In February 2021, the proof of concept for the AAI architecture described in Section 2.1 was delivered. The Common AAI services under the name researcher-access.org and ISD proxy for Puhuri are delivered by GEANT. With this, in close collaboration with GEANT and other relevant parties, the integration of Swedish National Portal, Puhuri Portal, Puhuri Core, CSC proxy and resources commenced. The integration plan is roughly as follows:

- Integrate Swedish National Portal, Puhuri Portal, Puhuri Core and CSC proxy as OIDC clients to Puhuri ISD Proxy. Implement test IdP that shall be used for initial testing.

- Implement, test, validate and adapt (if needed) the registration flows described in section 1.3
- Implement, test, validate and adapt (if needed) the project allocation flows described in section 1.4
- Implement, test, validate and adapt (if needed) access to resources flows.
- Start connecting and testing production IdPs from eduGAIN.
- After the main flows are tested, identify and define ways to deal with corner cases.

The AAI requirements that were described in Deliverable D1 will be validated and implemented during the integration with researcher-access.org proof of concept as well. Currently there are no foreseen deviations from the requirements originally described.

In parallel with technical integration, discussion with Fenix about governance of Common AAI services is ongoing. Also the Level of Assurance requirements discussed in the next section are part of that discussion with Fenix.

Puhuri started exploring solutions to cater for the users from R&E sector that don't have federated identity and for industry users. Current thinking is that national eIDs solutions will be connected to researcher-access platform to complement federated identity access, and that a final fall back solution to register and manually validate identity of users will need to be put in place. These would be the features supported by the researcher-access platform, and conversation with GEANT about ways to support these in ongoing.

2.3 Level of Assurance Requirements

For access to LUMI and other resources connected to Puhuri there are certain requirements that users identity needs to fulfill when it comes to assuring its authenticity. This is necessary to manage risks related to the access control of services connected to Puhuri. This section presents current thinking about Level of Assurance that should be defined for access to LUMI resources, but these are still up for discussion and agreement with LUMI countries.

There are several Assurance frameworks in existence today. Access to LUMI resources will be dominantly supported by identites coming from the IdPs from the R&E sector and eduGAIN. Best-fit and natural is to use the Assurance Framework that originated as collaborative work of R&E federations - the REFEDS Assurance suite[1].

According to REFEDS Assurance suite, there are two components, or vectors, when it comes to Assurance:

- **Identity Assurance**, that talks about quality and trustworthiness of the identity, and

[1] https://wiki.refeds.org/display/ASS

- **Authentication Assurance**, that talks about quality and trustworthiness of the authentication method.

These two aspects are elaborated in the following.

2.3.1 Identity Assurance Requirements

Framework that we will use to express identity assurance is REFEDS Assurance Framework https://refeds.org/assurance [RAF]

For access to LUMI resources, thinking is that following eduPersonAssurance values as defined by RAF are required:

- To insure **identifier uniqueness**:
  - https://refeds.org/assurance/ID/UNIQUE; or
  - https://refeds.org/assurance/ID/eppn-unique-no-reassign

- To insure sufficient **identity proofing and credential issuance, renewal, and replacement:**
  - https://refeds.org/assurance/IAP/medium; or
  - https://refeds.org/assurance/IAP/high

In the beginning, Puhuri will support all IdPs that LUMI countries wish to connect, but in discussion and agreement with LUMI countries, there would be a date defined when Identity Assurance requirements will become mandatory. For IdPs from countries that are not part of LUMI, Puhuri would mandate defined Identity Assurance requirements from the beginning.

An option for users whose IdP will not be able to meet Identity Assurance requirements would need to be established. This will require additional work for users, and in general it is not advisable to resort to this option unless IdP can not comply. Researcher-access platform will need to implement a feature for users to elevate their identity assurance by using:

- eID to elevate the trust
- Manual identity proofing process

2.3.2 Authentication Assurance

Framework that we will use to express authentication assurance is REFEDS Authentication Profile https://refeds.org/profile/mfa [MFA].

Current thinking is that adherence to the MFA profile will not be necessary for all use cases when user needs to logins, but for certain cases such as:

- User managing his/hers SSH key
- Project PI managing project membership

3 Resource allocation technical setup

In the Puhuri project, Waldur software[2] is being used for resource allocation and accounting. Waldur is an open-source cloud marketplace with a self-service environment for users to request and get access to resources. Waldur was used in NeIC's Dellingr project for testing cross-border resource sharing between Nordic countries and outcome was positive. Same software is used as a basis for self-service of Estonian National Research Infrastructure (ETAIS) and is co-developed and operated by the Puhuri project member organization, ETAIS/University of Tartu.

In the Puhuri project, Waldur platform was enhanced for better support of custom deployments and business logic. The deployment was automated by using GitLab repository's Continuous Integration (CI)[3], Ansible[4] configuration and deployment scripts and Helm Charts Packaging for Kubernetes[5]. Effort is made to assure that operations of the Puhuri Core as well as Puhuri Portal services are time and effort efficient while also reliable.

Puhuri Core is the heart of the resource allocation as it holds necessary information about users, group management, roles, resources, accounting etc. Puhuri Core is accessible using REST API with target users of API being:

- National allocation portal;
- Service Providers that want to use allocation information via Puhuri Core

Thus, in the LUMI Use Case Puhuri Core users are the resource allocation organizations, which allocate the resource of their share. Representatives of the allocator (review team) can make the decisions on providing allocations to each of the projects. Puhuri Core users hence have the following roles:

- "Allocating body member" - able to create projects and allocate resources according to the Country's review board.
- Project admins - PIs
- Project members - all users, non-PI

3.1. Portal as Puhuri Core client

The above mentioned REST API is likely used via a portal. If the country does not have a national allocational portal or is not willing to integrate at once, Puhuri partner ETAIS plans to provide Puhuri Portal for each allocator. This Portal allows to manage projects and allocations, get reports and statistics as well. Puhuri Portal and its integration will be described more deeply

[2] https://github.com/waldur
[3] https://docs.gitlab.com/ee/ci/
[4] https://docs.ansible.com/ansible/latest/index.html
[5] https://github.com/helm/helm

in Deliverable 3 (Integration of portals and LUMI services). Business model for adopting Puhuri Portal for non-LUMI usage will be decided and agreed at a later stage.

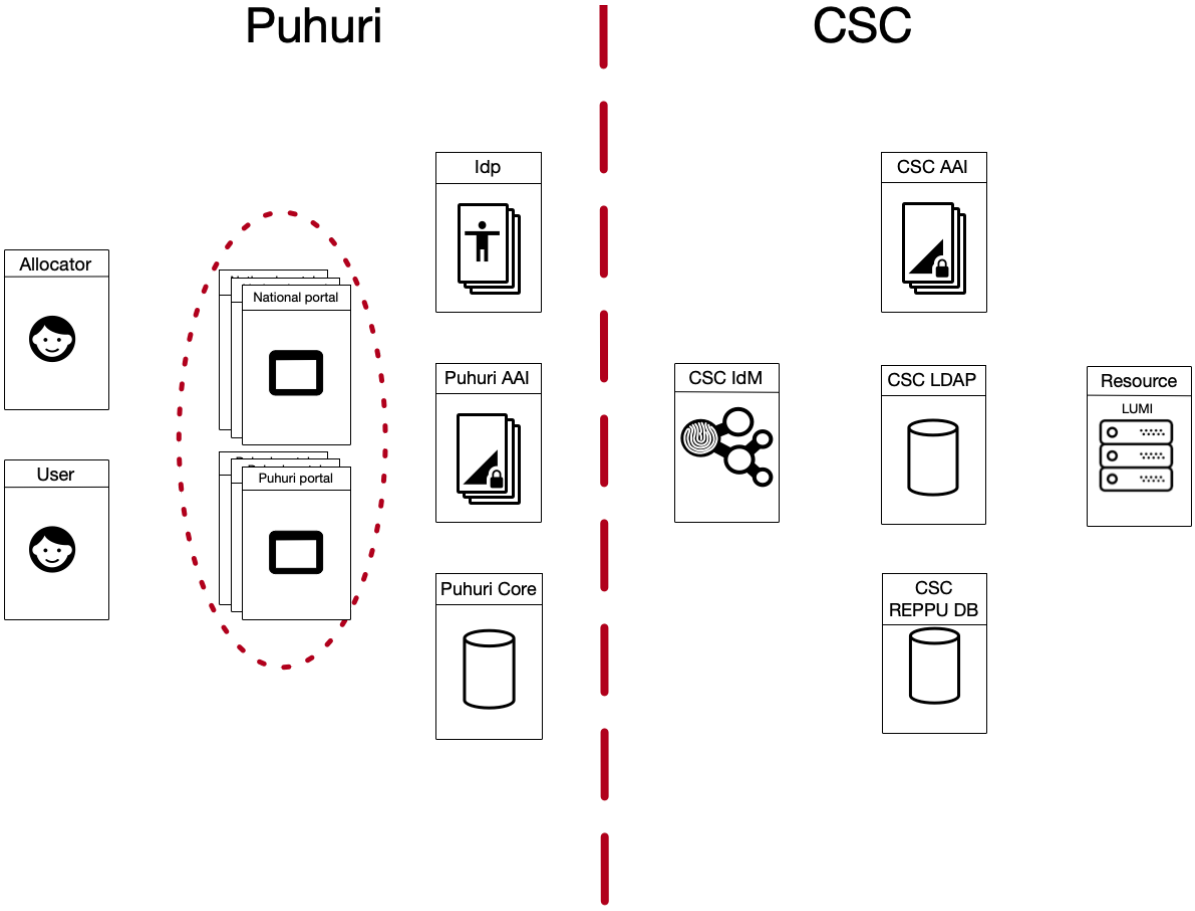

Figure 3. Puhuri portal and national portal are clients for Puhuri Core APIs. The portals are used both by the User and Resource Allocator. Into the Puhuri Portal they must authenticate via Puhuri AAI.

3.2 Environments

In total, we run three different k8s-based deployments for Puhuri Core. Each environment[6] has its own intention and policy.

- Dev (https://puhuri-core-dev.neic.no/) runs the latest code and is updated frequently.
- Demo (https://puhuri-core-demo.neic.no/) is deployed with the versioned code, either the same or newer than on production and contains mocked data for users and consumption.
- Production (https://puhuri-core.neic.no/) is intended to be used for production data exchange.

All deployments are integrated with eduTEAMS-based AAI aka Researcher Access AAI - and implement the business processes we have decided so far in the Puhuri project. The whole deployment process is designed to allow for dynamic changes into the processes during the stabilization phase. In addition to OIDC flow, Puhuri Core implements a custom integration with Researcher Access for looking up user data based on provided unique references of the user, aka CUID.

3.3 Information stored in Puhuri Core

Information stored in Puhuri Core depends on the information objects, which are needed to manage users, projects and resource allocations. There are 7 main information objects in Puhuri Core:

- Project - a project is an object with one or more allocations on one or more resources. The project has a number of consumable units assigned for each of the allocations. It is assigned to a PI, has one or more optional proxy-PI and one or more members. The project is time limited by the lifespan
- Puhuri Account - a representation of a unique physical person with a unique identifier within Puhuri Core. Containing the contact information and institutional details for the person. A Puhuri Account can have multiple Resource Accounts
- Role - role information to be be used for example when making authorization decisions in Puhuri Core or Resources
- Allocation - the right to access and use a resource from a start date to an end date and to an extent of a number of consumable units as defined in a project
- Resource account - an unique identifier for a Puhuri Account on a resource. (Username or similar)
- Resource component - an allocatable part of a resource with a type and a unit
- Resource - a Resource object represents a compute cluster, storage system, or cloud resource.

[6] https://puhuri.neic.no/environments/

In Section 4.4. there is an image of the Puhuri Information Model. For integrating with Puhuri Core, it must be noted that Waldur and API uses little bit different terminology:

Puhuri	**API**
Resource	Offering
Resource component	Offering component
Allocation	Resource

Project attributes:

- `uuid` - unique identifier which is autogenerated by the Puhuri Core
- `name` - name of the project, chosen by the user
- `description` - short description of the project, chosen by the user
- `type` - chosen by the Service Provider from predefined list (LUMI options: general access, benchmark, extreme scale, development, community)
- `end_date` - date when project ends, chosen by the user
- `url` - project url
- `created` - date/time when project is created in Puhuri Core
- `customer` - organization's url under which project is created
- `customer_abbreviation` - organization's abbreviation
- `customer_name` - organization under which project is created
- `customer_native_name`
- `customer_uuid` - organization's unique identifier

Puhuri account attributes are following and data is mostly coming from Puhuri AAI:

- CUID - community user identifier coming from Puhuri AAI
- Given name - user's given name
- Surname - user's surname
- E-mail - user's e-mail
- Home Organization - user's home organization
- Affiliation
- Identity assurance
- Phone number - user's phone number, managed by user
- SSH key - if user has ssh key stored in Puhuri AAI profile, it will be pulled to Puhuri Core, managed by user

Allocation is called Resource in Waldur/API and its attributes are:

- `offering` - respectful offering's URL

- `attributes` - specific attributes for the offering. These are dynamic, but for the bootstrapping we took the metadata as was used in the NeIC's Dellingr project.
    - `name` - allocation name
    - `Oecd_science_domain_configuration` - OECD science domain
- `plan` - plan's URL (if offering is billable)
- `limits` - a set of resource limits for an allocation (CPU, GPU and storage)
    - `gb_k_hours`
    - `gpu_k_hours`
    - `cpu_k_hours`

Resource is called offering in Waldur and its attributes are following:

- `name` - offering's name
- `name_exact` - offering's exact name
- `customer` - organization's URL
- `customer_uuid` - organization's UUID
- `allowed_customer_uuid` - allowed organization's UUID
- `service_manager_uuid` - service manager's UUID
- `attributes` - a set of attributes (key-value pairs)
- `state` - offering's state (Active, Draft, Paused, Archived), should be Active
- `category_uuid` - category's UUID
- `billable` - signalizing if an offering is billable or not, should be true
- `shared` - signalizing if an offering is public or not, should be true
- `type` - offering's type

Some of the attributes are Service Provider specific and these existence depends on the requirements from the Service Provider. For example, OECD science domain is required by LUMI.

3.4 Puhuri Core integration with Puhuri AAI

Puhuri Core is integrated in with Researcher Access AAI over two methods:

1. Is a standard OIDC protocol, where Puhuri Core acts as a client. This allows it to do permission grants as well as accept logins of Puhuri users into Core. While this is technologically possible, direct end user login to Puhuri Core is not foreseen.
2. Access to user registry over a custom EduTeams userinfo endpoint for retrieving user data from the registry based on the lookup. Such integration is needed to be able to present National Portals API-endpoint for adding users to the resource allocation using the unique identity reference of the Puhuri AAI. At the moment of writing the exact final approach to protecting the endpoint from the Researcher Access AAI side is being discussed. Temporary implementation relies on the concept of protected resource of Oauth2 service.

3.5 Puhuri Core functionality for resource allocators

Allocators are external parties that are entitled to manage projects and resource allocations in Puhuri Core. In the scope of Puhuri, resource allocators are typically organizations operating national portals or research communities.

Resource allocator is expected to:

- Be eligible to share specific resources, e.g. LUMI share.
- Be aware of the Researcher Access identifiers of the users, aka CUIDs.

The up-to-date documentation for the allocators is kept at https://puhuri.neic.no/resource-allocators/ . Below we provide a high level guide reflecting the situation at the moment of writing of the deliverable. The finalized version of the Puhuri Core will expose documentation in OpenAPI format.

3.5.1 Common operations

Almost all operations require authentication. Authentication process is a two-step: 1. Generation of authentication token using Authentication API. 2. Passing that token in the Authorization header along with all other REST API calls.

Please note that all of the responses to the listing are paginated, by default up to 10 elements are returned. You can request more by passing page_size=<number> argument, number up to 200 will be respected. Information about the whole set is contained in the response headers.

3.5.2 Project management

Customer lookup

Puhuri Core implements a multi-tenant model to allow different organizations to allocate shared resources simultaneously and independently from each other. Each such organizaton is a customer of Puhuri Core and is able to create its own projects. Project allows us to create new allocations as well as connect users with the project.

Hence, to create a project, one needs first to have a reference to the customer. The reference is a stable one and can be cached by a REST API client.

Customers are created by Puhuri Core support team. Please reach out to support@hpc.ut.ee if you think you should have one.

Project creation

In order to create a new project in an organization, user needs to provide the following fields:

- `customer` - URL of the project's organization
- `name` - project's name
- `description` - description of a project description
- `end_date` - optional date when the project is
- `backend_id` - optional identifier, which is intended to be unique in the resource allocator's project list. Can be used for connecting Puhuri Core projects with the client's project registry.

Please note that the project becomes active at the moment of creation!

Project update

It is possible to update an existing project using its URL link. Name, description and backend_id can be updated.

Project lookup

User can list projects and filter them using the following query parameters:

- `name` - project's name (uses 'contains' logic for lookup)
- `name_exact` - project's exact name
- `description` - project's description (uses 'contains' logic for lookup)
- `backend_id` - project's exact backend ID

In case API user has access to more than one customer, extra filter by customer properties can be added:

- `customer` - exact filter by customer UUID
- `customer_name` - filter by partial match of the full name of a customer
- `abbreviation` - filter by partial match of the abbreviation of a customer

3.5.3 Project membership management

Puhuri AAI user mapping lookup

Puhuri Core maintains its own set of user records. Project membership is essentially a link between a project and user, carrying also information about the role name.

A mapping from the Puhuri AAI CUID is implemented as a separate call. Resource allocator is able to send the CUID, which would return a link to Puhuri Core user identity or error message, if this was not possible (e.g. CUID is incorrect or connection with Puhuri AAI user registry has failed). Note that endpoint returns UUID, full URL of the user is constructed as Puhuri Core URL + /api/users/USER_UUID.

Membership management

Creating a membership for a user means creating a permission link. While multiple roles of a user per project are allowed, we recommend for clarity to have one active project role per user in a project.

The list of fields for creation are:

- `user` - a user's URL, looked up from a previous step
- `project` - a URL of a project where the permission needs to be created.
- `role` - a role of the user. 'member', 'admin' (aka co-PI) and 'manager' (aka PI) are supported.

Each permission has a unique URL. To remove the permission, REST API client needs to send a DELETE HTTP request to that URL.

It is also possible to list available project permissions along with a various filters. Users can list permissions for all projects in their visibility range. To limit the permission set to a specific project or user, the following filters are supported:

Possible query params for filtering:

- `project` - a projects's UUID
- `project_url` - a projects's URL
- `user` - a user's UUID
- `user_url` - a user's URL
- `username` - a user's username (aka CUID)
- `full_name` - a user's full name

- `customer` - an organization's UUID
- `role` - a role's name

3.5.4 Resource allocation management

Creating and managing resource allocations in Puhuri Core follows ordering logic.

All operations on resources, which lead to changes in allocations - e.g. creation, modification of allocated limits or termination - are wrapped in an order. It is possible to have multiple actions of the same type in one order. Such actions are called Order items.

To create a new Allocation, one must first choose a specific Offering from available. An Offering corresponds to a specific part of a shared resource that Resource Allocator can allocate. For example, it can be a national share of LUMI resources.

User can fetch the offerings and filter them by the following fields:

- `name` - offering's name
- `name_exact` - offering's exact name
- `customer` - organization's URL
- `customer_uuid` - organization's UUID

Generally an Offering has a stable UUID, which can be used in Puhuri Core client configuration. An Offering defines inputs that are required to provision an instance of the Offering, available accounting plans (at least one should be present) as well as attributes that can or should be provided with each request.

3.6 Puhuri Core functionality for service providers

The main service provider for Puhuri is CSC with its service LUMI. As such development of the functionality is largely driven by use cases that arise from LUMI, however we believe that they are generic enough to integrate a wide range of similar service providers.

From the service provider side, Puhuri Core provides several groups of APIs. Functionality related to the setup of environment on the Service Provider side:

1. Getting a list of active projects where service provider’s resources are available.
2. Getting a list of active Allocations of Resources.
3. Getting a list of members and their roles in each project.

Functionality related to reporting of accounting of resources

1. Updating usage of each Resource Allocation.

luu
Functionality in progress includes:

1. Ability to report on the status of the environment provisioning to improve feedback to resource allocators.
2. Ability to report locally generated usernames for each Puhuri AAI identity.

Service provider documentation is in development and will be available at https://puhuri.neic.no/service-providers/.

3.7 Towards a reference client for Puhuri Core API

In order to validate Puhuri Core APIs as well as to provide a graphical tool for the resource allocators that do not have or do not want to integrate their own allocation portal, we are working on a reference solution codenamed Puhuri Portal. The solution is planned to be described in more details in Deliverable 3.

Based on the same technology as Puhuri Core, Puhuri Portal users will have additional roles to reflect established processes of resource allocation. The roles will be clarified after the stabilization of Puhuri Core, so far envisioned ones are:

- Organization owner: represents a scientific organization or community, able to create new projects, manage project teams and manage requests for resources.
- Project manager (PI): able to request and use provisioned resources (with optional approval by Organization owner), can add project members from the same organization (i.e. pre-approved by the owner).
- Project member: can only participate in particular project work, able to use provisioned resources.
- Portal support: can provide support to users of the platform. Could potentially be linked with a national helpdesk.

A demo instance of the Puhuri Portal has been deployed in the same hosting environment as Puhuri Core. It is described in https://puhuri.neic.no/environments/ .

Resource allocation process is designed to be flexible and generic, meaning that we standardize the main flows and actions, but allow for customisations, where we foresee changes. For example, we expect LUMI to provide a common schema for allocating resources for the initial LUMI services (cpu-hours, gpu-hours and storage-hours), but we allow to provide own allocation components to other service providers or not yet clearly defined resources under LUMI umbrella. As such, Puhuri Core serves as a generic solution for sharing of resources by multiple allocators.

4 Resource integration with the LUMI use case

LUMI will be the first Resource that will integrate with Puhuri. Later on, Puhuri will also cater other Resources in addition to LUMI. Puhuri will have to define the rules on how to integrate such Resources. These Puhuri rules must align with the GEANT Code of Conduct, and also take into account the GDPR with regard to the information minimization principle.

CSC has implemented a mechanism for scheduled data synchronization from Puhuri Core REST API to its Identity Management System (IdM). This scheduled data synchronization will be used to pull in User, Project and Allocation data from Puhuri Core to CSC IdM.

The relevant data is pulled in from Puhuri Core REST APIs to CSC IdM using a Python script[7], which pre-processes and transforms the imported data into a format that is more easily consumed by the IdM system.

The IdM system regards Puhuri Core as the authoritative source of information with regard to Puhuri imported Users, Projects and Allocation data, and updates the imported objects local state in CSC IdM accordingly, when the imported objects state has changed in Puhuri Core. Please see sections 4.4 and 3.3 in this document for more information regarding the Puhuri information model.

The LUMI services will be able to consume the imported User, Project and Allocation data by querying the relevant information from CSC's LDAP cluster.

LUMI specific Terms of Use will be accepted by the User in Researcher Access AAI when registering for an Puhuri account.

Closed Projects and expired User accounts are de-provisioned from CSC's IdM system and related systems according to CSC's Project and User account life-cycle policy. Project and User data removal will be implemented using automated scripts that CSC uses for its other HPC systems. The data is removed at the end of the imported entities life-cycle process.

CSC will also implement a service for exporting LUMI Accounting information from CSC's reporting database (REPPU) to Puhuri Core. This will be done later in 2021.

[7] https://github.com/nyholmju/import-puhuri

4.1 User, Project and Allocation data import process

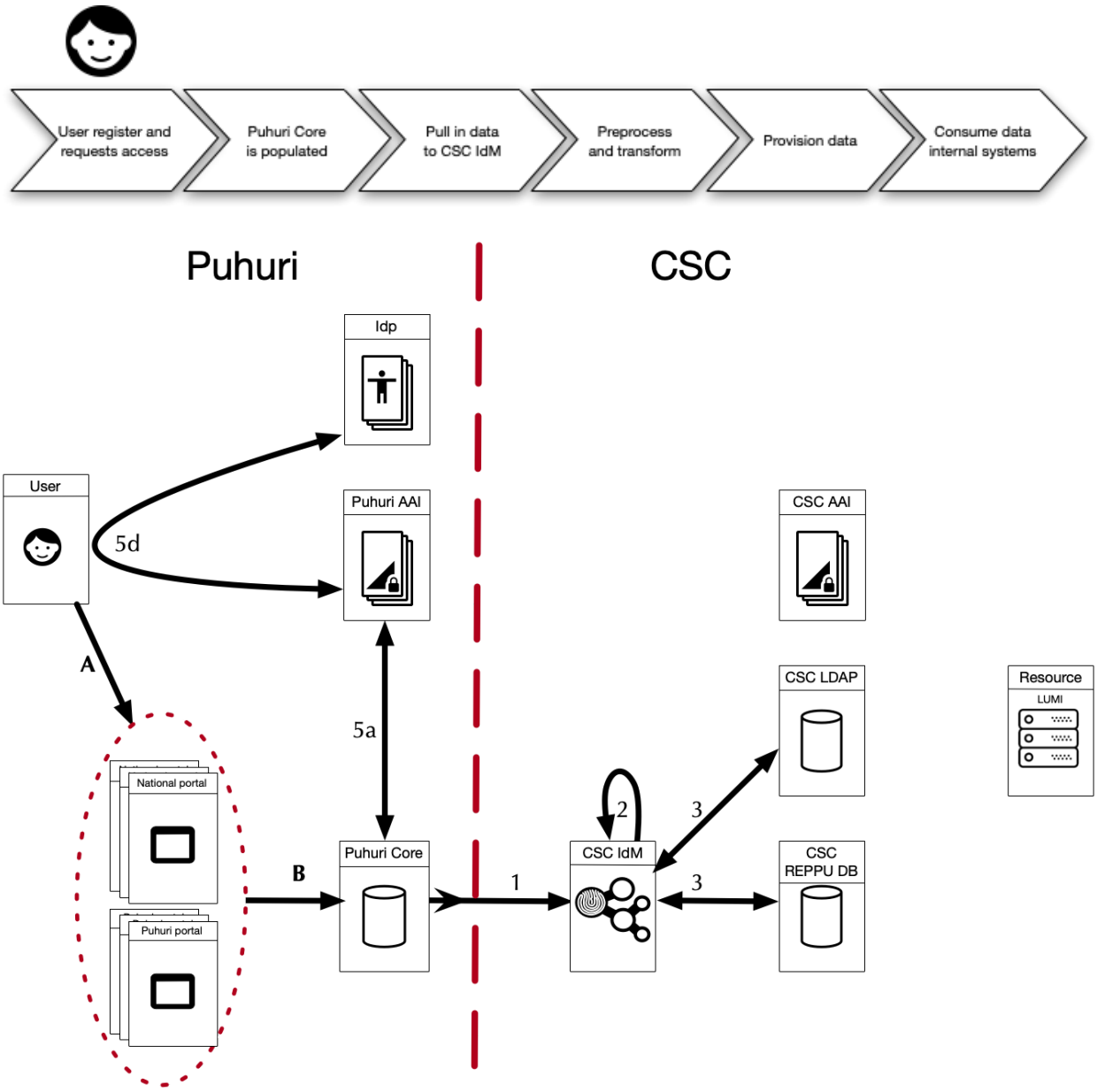

Figure 4. User registration process for LUMI. User, Project and Allocation data import procedure. When User has registered, Puhuri Core will get information from Puhuri AAI Registration Service (step 5a). This data, and possible additional data, such as User provided ssh public keys, are then propagated to CSC IdM (step 1).

The data import procedure outlined in Figure 4 consists of the following steps:

A) User registers, requests, and is granted resources
5)
5a) Authenticate User (initiation of the authentication workflow)
5d) User uses his/her home institution's IdP to authenticate (User attributes are returned to Puhuri Core)
In addition to the IdP provided attributes, User can give additional attributes such as ssh public keys
During the login process, the User accepts LUMI specific Terms of Use
B) Information on User, Project and Allocation is propagated to Puhuri Core
1) Users, Projects and Allocations are pulled in from Puhuri Core REST API to CSC IdM (script, cron job)
2) The pull script pre-process and transform imported data to comply with CSC's internal data model
Local instances of User and Project are created in CSC IdM during CSC IdM import
CSC IdM sends welcome email to User which contains information regarding Users local CSC user id
3) CSC IdM provisions imported local instances of User and Project to target systems

When interfacing with Puhuri Core REST APIs, the Waldur projects waldur_client.py[8] can be used as a good starting point, when developing programs or scripts, which publish or consume data from Puhuri Core.

[8] https://github.com/opennode/ansible-waldur-module/blob/6679b6b8f9ca21099eb3a6cb97e846d3e8dd1249/waldur_client.py

4.2 LUMI User authentication process

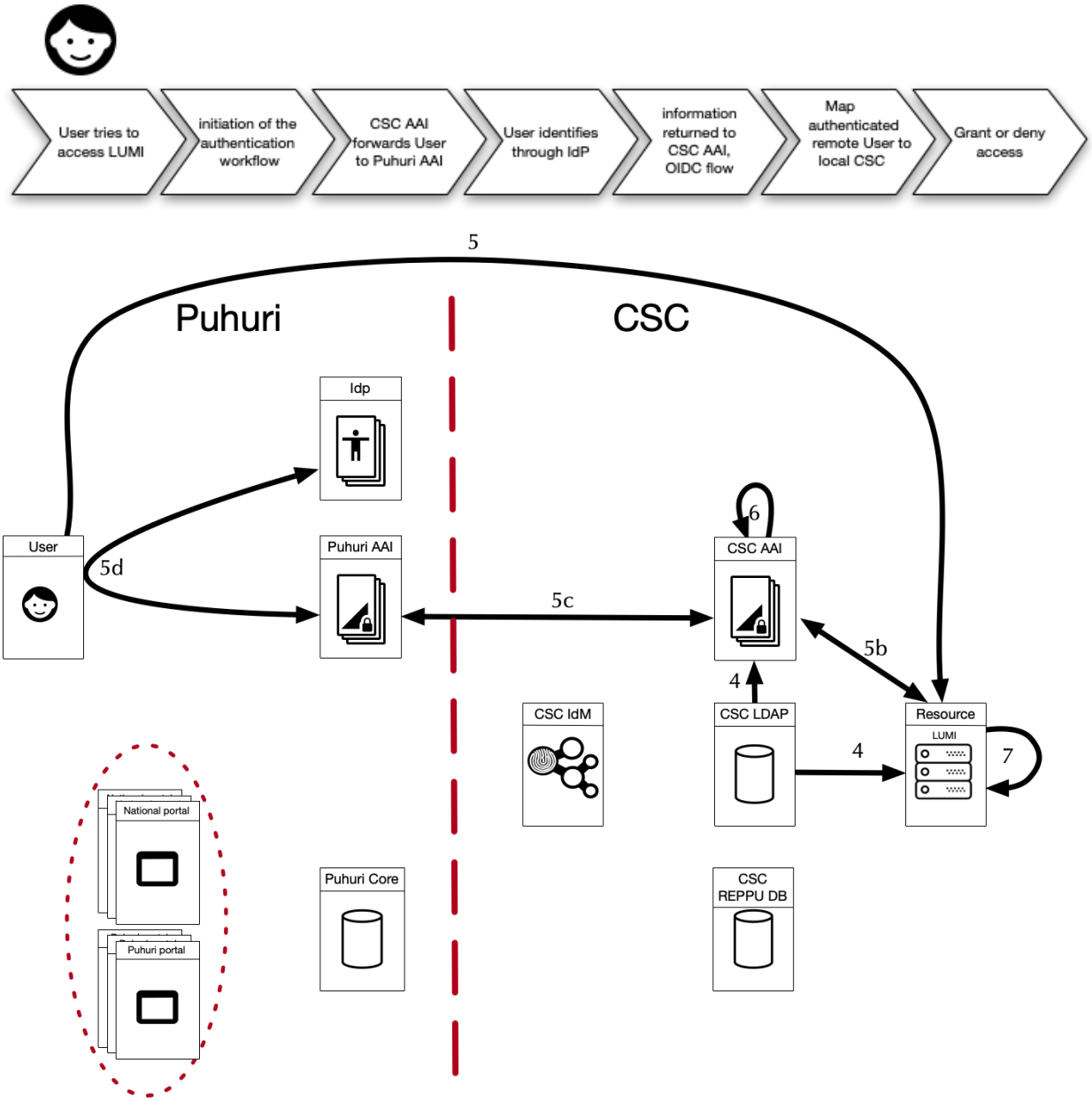

Figure 5. User authentication flow for the LUMI services. The Users can authenticate to the LUMI services using the OIDC device flow or an ssh public key based authentication.

4.2.1 Authentication flow when using OIDC device flow

The authentication flows outlined in Figure 5 consists of the following steps:

4) CSC AAI and LUMI Services consume the imported local instances of User and Project from CSC LDAP

5) User tries to establish ssh session with LUMI using his/her CSC provided local user id

5b) User is instructed by LUMI ssh service to authenticate using web browser
User opens provided hyperlink in web browser or scans provided QR code
Users web browser is redirected to CSC AAI

5c) User selects Puhuri AAI login method in CSC AAI

5d) User uses his/her home institution's IdP to authenticate
Users attributes are returned to CSC AAI

6) CSC AAI attempts to map authenticated remote User to local CSC identity
CSC AAI queries CSC LDAP using Puhuri AAI provided unique User identifier (CUID)
If a match is found in CSC LDAP, and the matched local user id is not disabled in CSC LDAP, the web browser prompts the User to Approve or Deny the ssh access request

7) LUMI grants or denies access based on CSC AAI authentication result and imported Users and Projects state in CSC LDAP

4.2.2 Authentication flow when using ssh public keys

The authentication flows outlined in Figure 5 consists of the following steps:

4) CSC AAI and LUMI Services consume the imported local instances of User and Project from CSC LDAP.

5) User tries to establish an ssh session with LUMI using his/her CSC provided local user id while requesting an ssh key based authentication.
Steps **5b), 5c), 5d)** and **6)** are not performed when using the ssh key based authentication with LUMI.

7) LUMI reads Users ssh public keys from LDAP and/or LUMI file system and performs ssh key based authentication.
LUMI grants or denies access based on ssh public key authentication result and imported Users and Projects state in CSC LDAP.

The Resource provider may have additional processes to authorise the first login, which are not feasible or wise to implement in earlier processes. For example, those can be very service specific processes, such as collecting complementary User information, that is required by the Resource provider, but is not directly available via the Puhuri integration. A Resource may also implement a multi factor authentication in addition to the authentication flow described above.

4.3 LUMI Accounting data export process

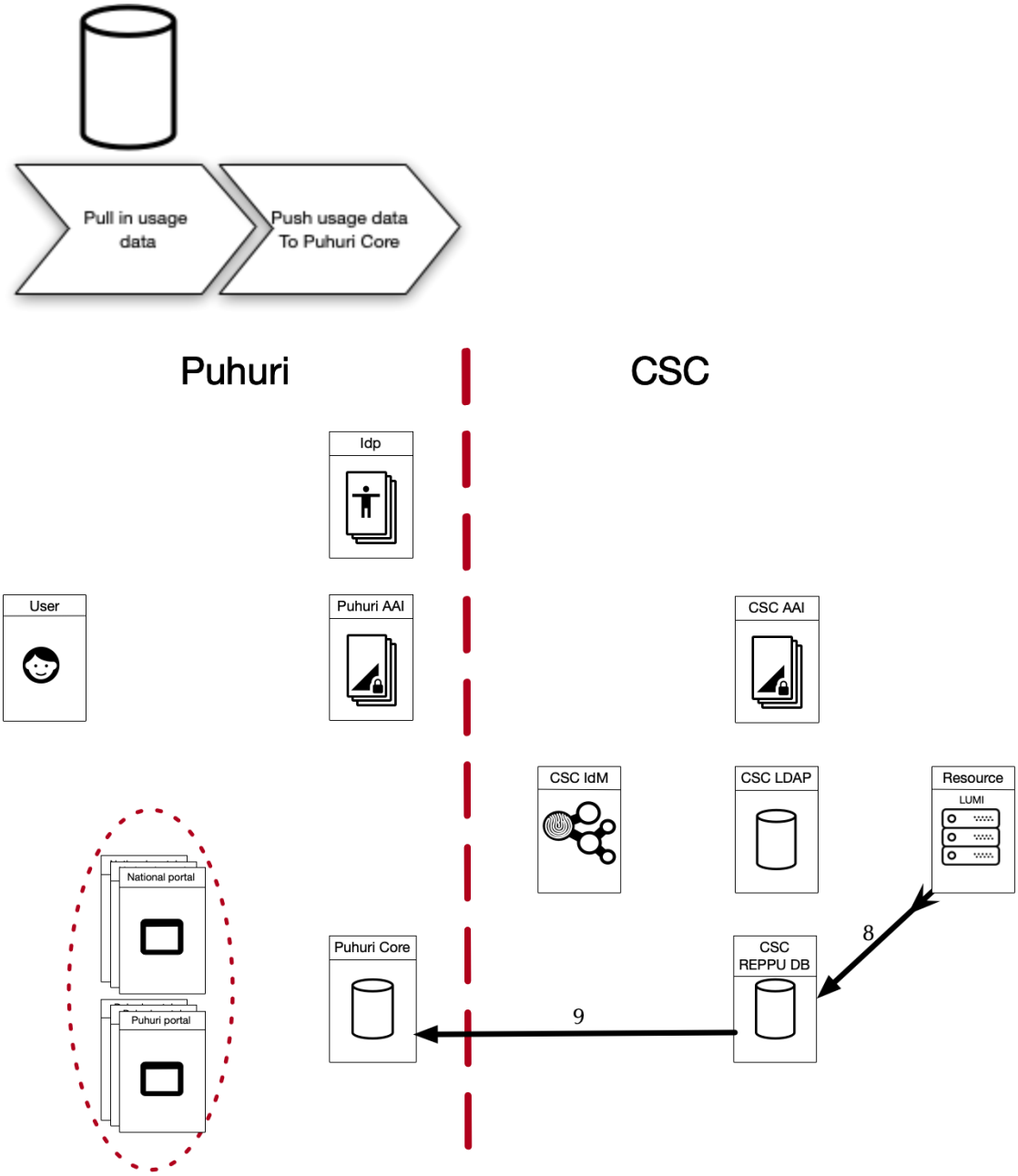

Figure 6. Accounting data export procedure from LUMI Resource to Puhuri Core.

The accounting reporting steps are:

8) Pull in Resource usage data from LUMI services to CSC's reporting database (REPPU)

9) Push Resource usage data from REPPU database to Puhuri Core (matches Projects in Puhuri Core using Puhuri Project UUID)

4.4 Puhuri information model

Figure 7 shows the information model of Puhuri. The terminology used in Figure 7 is documented in section 3.3 of this document. Resource Provider's local data model needs to map to the Puhuri Model. The application review process drawn on top of the image is not yet supported by Puhuri Portal.

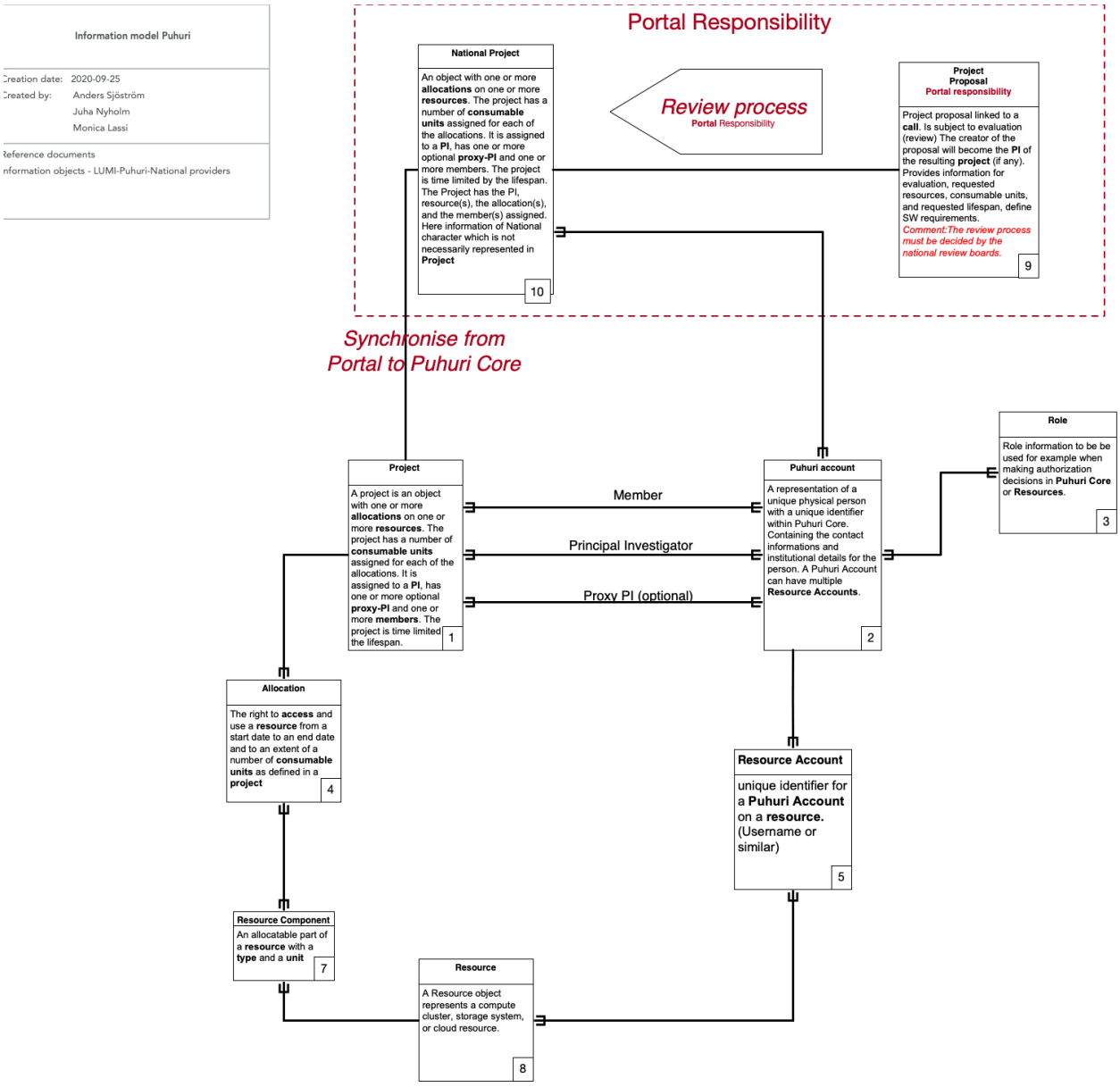

Figure 7. Graphical representation of the Puhuri information model.

References

[D1] Requirements and architecture plan for Puhuri. Puhuri Deliverable 1
https://zenodo.org/record/4288776#.YFytZ9yxVhF

[LUM] LUMI EuroHPC supercomputer web page. https://www.lumi-supercomputer.eu/

[PUH] Puhuri Documentation Portal https://puhuri.neic.no