

## PUHURI Deliverable D4

Project Title:	Puhuri
Public	This document is PUBLIC
Deliverable title:	Puhuri Accounting and Reporting
Contractual delivery date:	15th of December 2021
Status	V 1.0
Actual delivery date:	2nd of March 2022
Updated	2nd of March 2022
Partner(s) contributing to this deliverable:	FI, EE, SE

### Authors and Contributors:

Erik Edelmann, Kent Engström, Kalle Happonen, Jarno Laitinen, Ilja Livenson, Juha Nyholm, Ahti Saar, Anders Sjöström, Jefims Gasels and Sergei Zaiaev

## Table of Contents

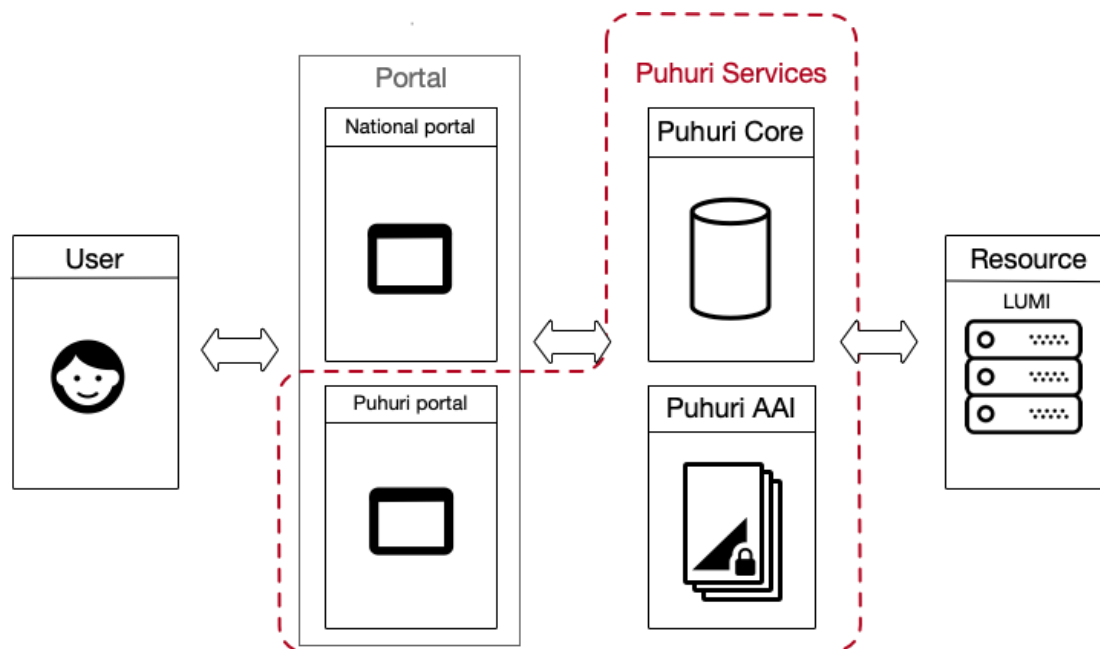
<b>Introduction</b>	<b>3</b>
<b>Accounting process on Service Provider</b>	<b>4</b>
LUMI Accounting data export process	4
Monitoring of the resource consumption	5
<b>Puhuri reporting</b>	<b>6</b>
Puhuri Core reporting	6
Puhuri statistics webpage	7
Puhuri Portal reporting	8
<b>Accounting Integration of the Swedish national portal to Puhuri</b>	<b>11</b>
Background	11
Fetching Accounting Data	11
Fetching Project Names	12
Fetching Usernames	12
Future Work	12
<b>References</b>	<b>14</b>

# Introduction

This deliverable is a report of Accounting and reporting for Puhuri Services. 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 is hosted at CSC in Kajaani data center. LUMI is the first use case for Puhuri and thus we focus on its accounting in this deliverable. The projects are awarded by LUMI consortium's resource allocators and also by EuroHPC Joint Undertaking (JU) via PRACE Peer Review portal.

LUMI consortium countries may integrate their national portal to Puhuri or use Puhuri provided Puhuri Portal to manage the LUMI projects and to see the accounting data (Figure 1). The LUMI integration is described in deliverable 3 [D3].

The technical integration documentation in <https://puhuri.neic.no> has been continuously extended. Please refer to that website for the latest technical details.



**Figure 1. Puhuri's high level architecture.**

Chapter 1 is accounting collection implementation at CSC for LUMI supercomputer

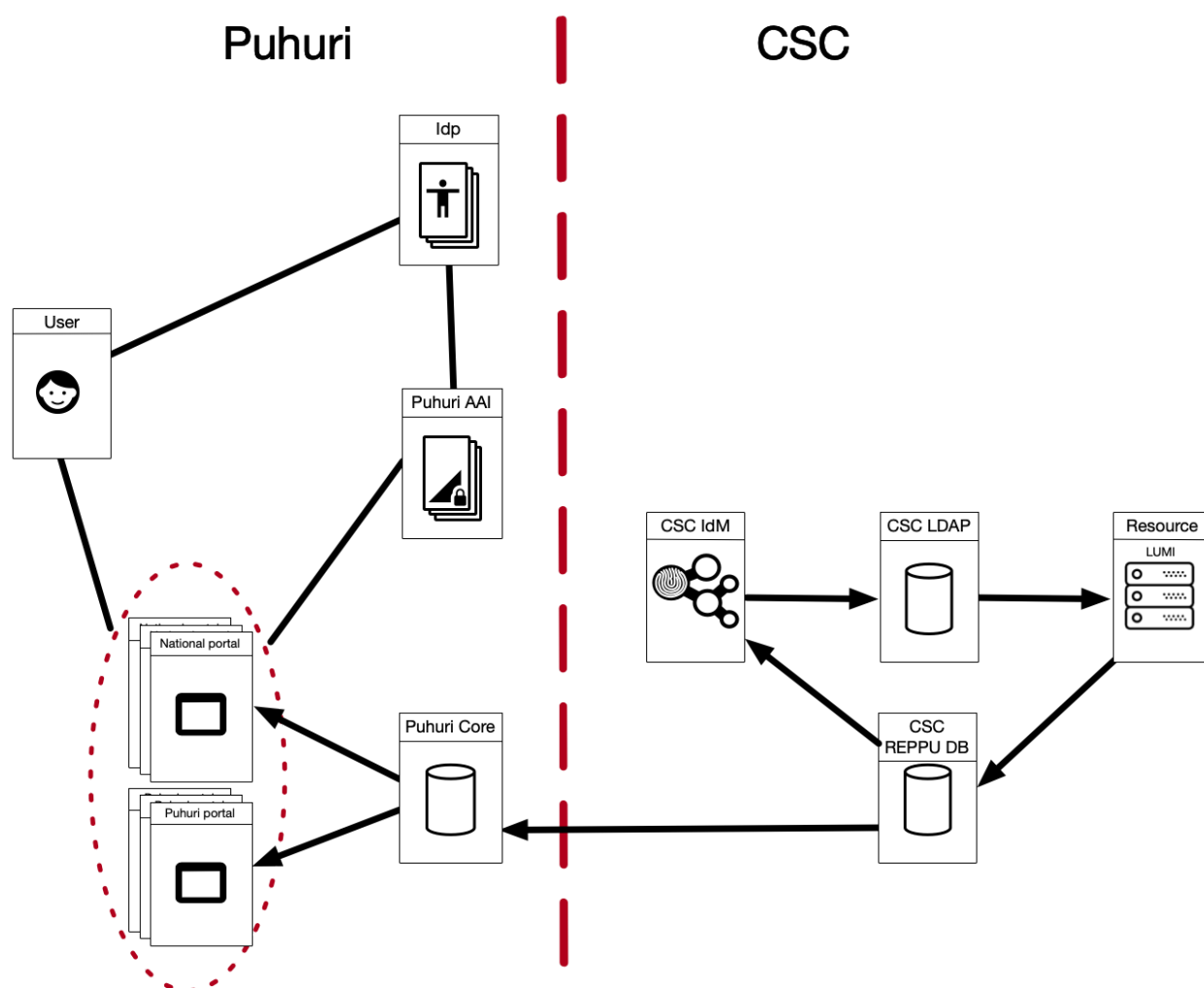
Chapter 2 describes the accounting collection in Puhuri Core and reporting in Puhuri Portal and Puhuri statistics web page

Chapter 3 reports how SNIC is integrated SUPR portal for accounting

# 1. Accounting process on Service Provider

At the moment, Puhuri collects accounting on the allocation level granularity, i.e. not individual jobs or users. Accounting data is collected for each month. The values for the current month can be updated frequently, e.g. in case of LUMI it is done daily. The sections below provide a more detailed description of integration done for LUMI, which is the first service provider in Puhuri.

## 1.1. LUMI Accounting data export process



**Figure 2. Accounting data export procedure from LUMI Resource to Puhuri Core. From there the National Portals and Puhuri Portals can fetch the accounting information and report that in a way they wish to implement. Chapter 2 tells about Puhuri Core database**

**and Puhuri Portal, which are developed by University of Tartu. Puhuri Portal can be used for those, who do not have a national portal integrated to Puhuri Core. [D3]**

LUMI accounting collection is:

1. Pull in Resource usage data from LUMI services to CSC's internal reporting database (REPPU). For example there is a script, which reads SLURM log file. Other scripts are reading storage usage per project.
2. Identity management system also exports Users and their Projects to REPPU database.
3. Push Resource usage data from REPPU database to Puhuri Core (matches Projects in Puhuri Core using Puhuri Project UUID). Database schema is CSC specific. This is scheduled to run several times per day.

Same resource allocation units are used for accounting as what was received from Puhuri Core. CSC has implemented scripts for data synchronization with Puhuri Core and its Identity Management (IdM) system. One script is used to pull in User, Project and Allocation data from Puhuri Core to CSC IdM and one is pushing the accounting data back to Puhuri Core. CSC is also sending CSC projects to Puhuri Core for Resource Allocation reporting since CSC projects are not managed in Puhuri Core.

During local instance creation, the imported objects are extended with auxiliary attributes, which are a part of CSC's IdM systems schema for Users and Projects. These auxiliary attributes include, among other things, the imported Puhuri Users local CSC username and the imported Puhuri Projects local CSC identifier, which can be used in accounting reporting. However, at the moment user specific accounting is not sent to Puhuri Core.

## 1.2. Monitoring of the resource consumption

LUMI services can read the resource consumption situation from the LDAP database, where Idm is also pushing the usage information fetched from the accounting database. In that manner, LUMI Services such as SLURM job scheduler or storage services can allow or prohibit sending new jobs or data. As the accounting is done based on the end date, it is possible that on-going SLURM jobs may consume more resources than what was allowed. Service Provider could implement various logic to control the usage.

## 2. Puhuri reporting

### 2.1. Puhuri Core reporting

Puhuri Core provides APIs and SDK<sup>1</sup>, which can be used by Service Providers to push resource allocation usage data to Puhuri Core. There is a symmetrical API endpoint for pulling the accounting data from the Puhuri Core aimed at Resource Allocators. In particular, this endpoint is used by Puhuri Portals which pull data and make it visible to end-users, Resource Allocation committee members as well as to Puhuri Portal operators (see Chapter 2.3). Puhuri Core SDK can be used by other clients (e.g. national portals, see Chapter 3) to easier integrate collected accounting data with custom logic.

Puhuri Core contains mostly operational accounting data, i.e. the last update of the data within each month. For historical charts and trends additional service - <http://puhuri-stats.neic.no/> also called as Puhuri Stats - has been developed, which is collecting and aggregating regularly exported accounting data (see the next section 2.2).

Since project attributes (and analysis by these) can vary between the Service Providers, then different Service Providers can have different dashboards and graphs. For example, industry vs academic projects division is one of the key performance indicators of LUMI, then we will implement this kind of dashboard.

Permission logic is as follows:

- Service Provider (e.g. LUMI) can see all projects related to this Service Provider in Puhuri with allocation in creating or active state.
- Service Provider is able to report usage for allocation in active state.
- Resource allocators are able to read reported usage data.

Puhuri Stats provide visibility according to the following logic:

- Service Provider can see division specific data - multiple Resource Allocators can share a division, e.g. be from one country.
- Resource Allocators from the same division can see each other's data, e.g. different organizations in a country sharing a common LUMI national share.
- Common statistics - counts of users, project, organizations - can be seen by users authenticated with MyAccessID.

---

<sup>1</sup>

<https://puhuri.neic.no/SDK%20guide/allocation-management-sp/#reporting-usage-for-a-resource-allocation>

## 2.2. Puhuri statistics webpage

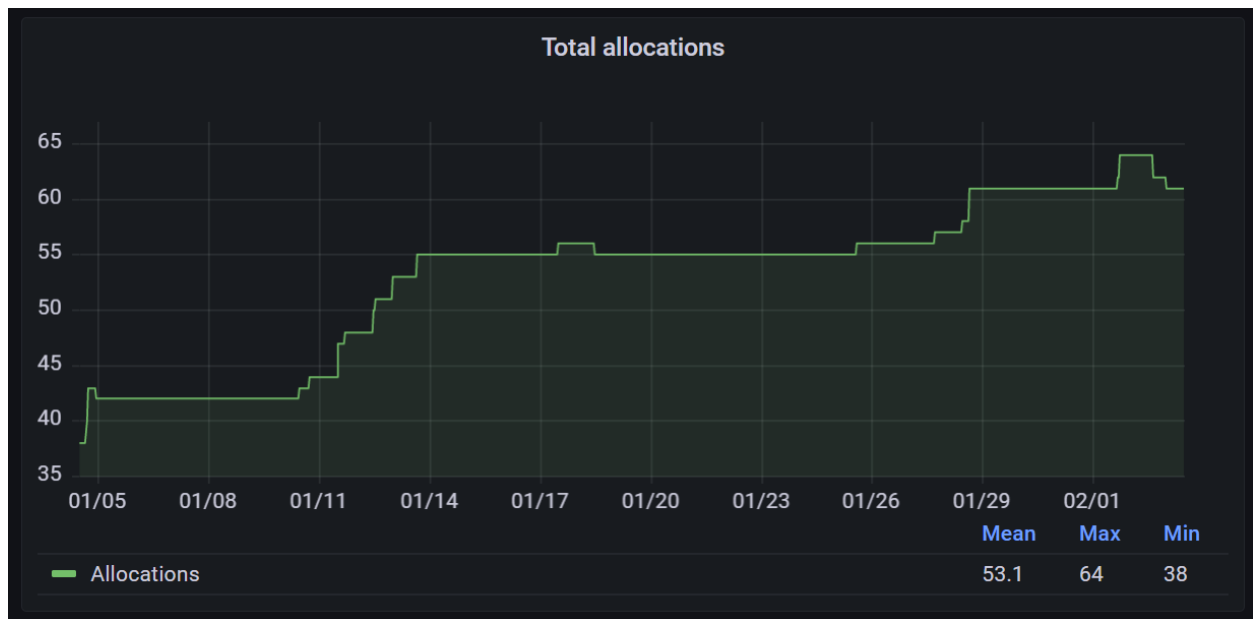
Puhuri statistics webpage<sup>2</sup> offers an opportunity to view accounting (usage and allocation) and general reporting (number of end-user and projects) information using different dashboards.

The dashboards have different access policies: some of them are semi-public (only login needed) while some are restricted (only specific users can see those). Authentication method is the same as for portals and Puhuri Core - MyAccessID. The goal is to develop integration with Puhuri Core so that the user role on the statistics page is set based on the user role in Puhuri Core. Currently role management is done manually based on request to Puhuri Core operator.

Currently used dashboard types based on restrictions are:

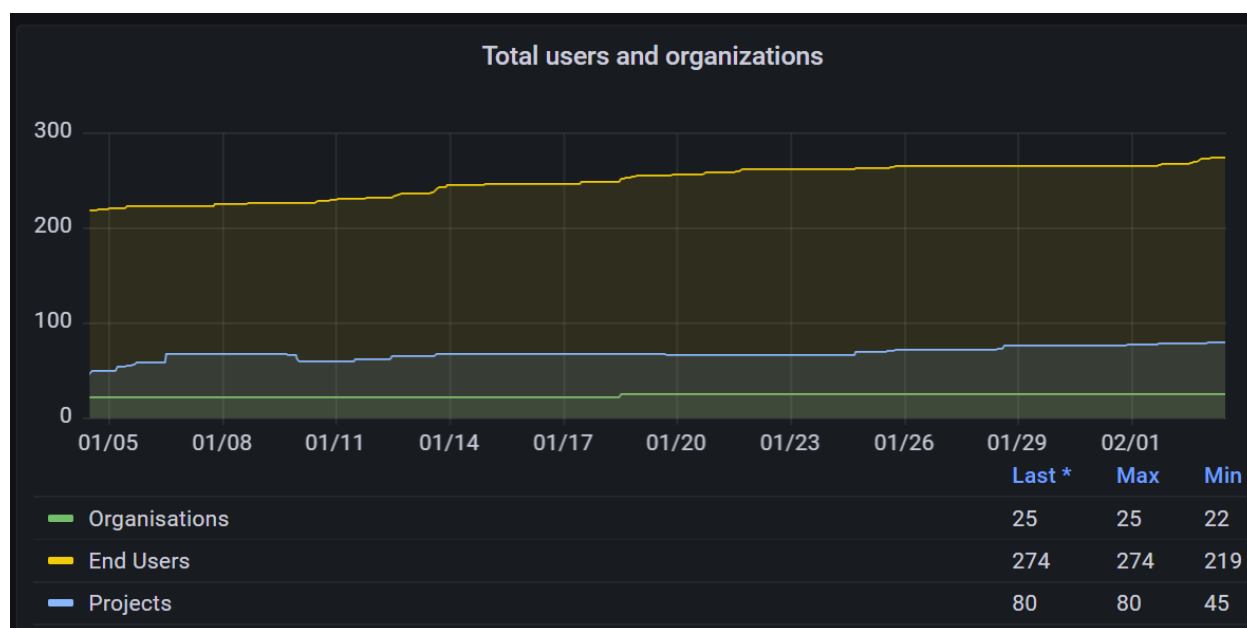
- Semi-public - all logged in users can see Puhuri general graphs
- Restricted to Resource Allocators - users with Resource Allocator's permission can see reporting data which corresponds to their allocating organization
- Restricted to Service Provider - Service provider representative can get access to whole accounting data

Current semi-public dashboards contain graphs like "Total allocations" and "Totals users and organizations".



**Figure 3: Number of Total Allocations mean resource allocation decisions of the Resource Allocators. One Project can have multiple Allocations.**

<sup>2</sup> <https://puhuri-stats.neic.no/>



**Figure 4: Reporting the LUMI Service specific amount of active users, organisations and projects.**

In the future we aim to add OECD science codes and industry usage flags as an attribute set for a project usage.

## 2.3. Puhuri Portal reporting

Resource Allocators are able to see accounting data in Puhuri Portal. This accounting information is pushed from a Service Provider's (ie LUMI) database to Puhuri Core once per hour and then pulled from Core to Portals.

To view the accounting data, Resource Allocators should login and "Support" menu on the upper right corner, then navigate to Reporting -> Usage reports on the left side menu.

Then the following window opens and it is possible to do some filtering if needed. "Client organization" here means the organization or grouping under which project and allocation was submitted. Typically this is PI's working group. "Project" enables to filter the results by project name. "Offering" enables filtering by project type. In LUMI case for example LUMI Regular.



## Usage reports

Support dashboard / Reporting / Usage reports

Accounting period

January, 2022

Client organization

Select organization...

Project

Select project...

Offering

Select offering...

Show

10

entries

Export as
Refresh

**Figure 5: Resource Allocators can filter the results by accounting period, client organization, project (project name) and offering (project type).**

Puhuri does not do any conversion of the reported values, so it is very important to agree on the used semantics and not to change these. The reported values are the same as they are in the Service Provider’s accounting database.

Below you can see how the reporting view looks like for the dedicated Puhuri Portal user who has the Resource Allocator role. All projects are listed with the monthly usage per allocation component (CPU k hours, GPU hours and for storage it is TB hours).

## Usage reports

Support dashboard / Reporting / Usage reports

Accounting period

January, 2022

Client organization

Select organization...

Project

Select project...

Offering

Select offering...

Show

10

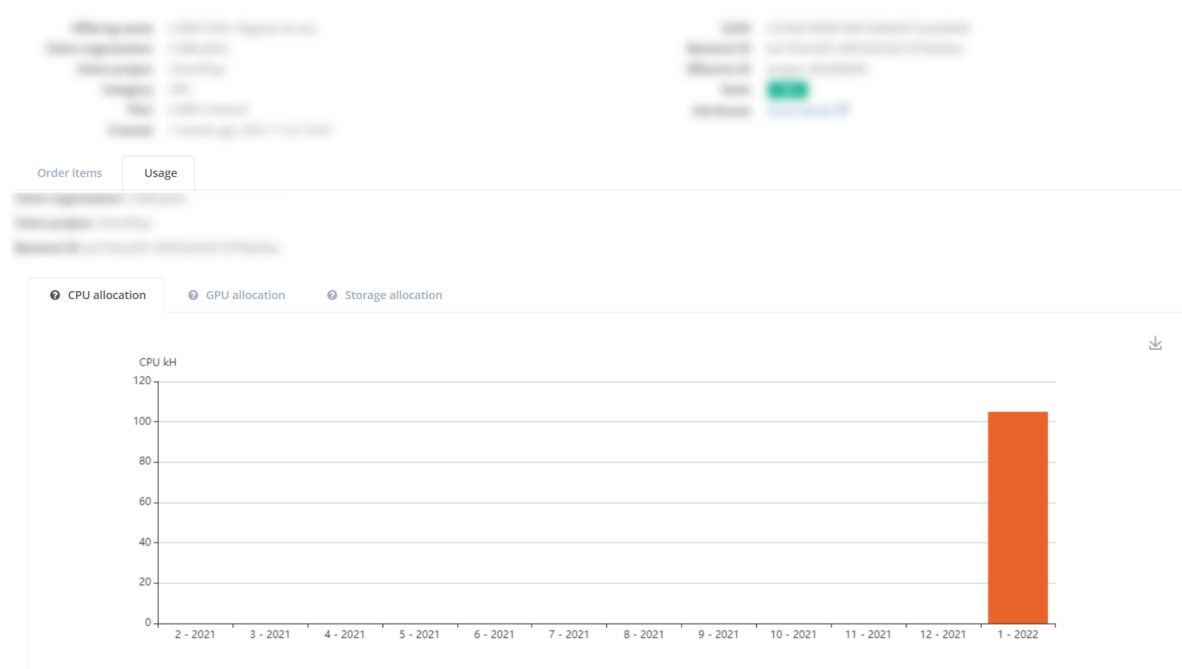
entries

Export as
Refresh

**Figure 6: Consumption status in January 2022 for the projects, which are visible for the viewer based on their organisation and role.**

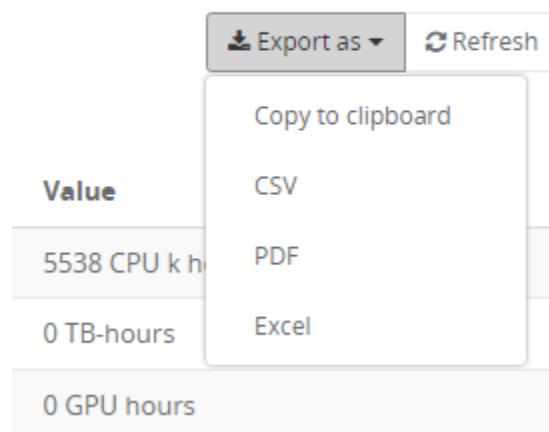
Another option is to view the reporting and used resources under the allocations. This is allowed also for PIs as well as other members of the team. To see the reporting, PIs should select their

project and then open the “usage” tab. Then it is possible to see the usage of the different components.



**Figure 7: CPU consumption in January 2022 for a project (project related information is blurred due confidentiality).**

Puhuri Portal also provides a way to export usage reports. When opening Usage Reports, then in the upper right corner, there is a button “Export as” and then it is possible to choose between CSV, PDF and XLSX file formats.



**Figure 8: Export file types for accounting information in Puhuri Portal.**

## 3. Accounting Integration of the Swedish national portal to Puhuri

### 3.1. Background

The SUPR project began in 2011 to provide SNIC (the Swedish National Infrastructure for Computing) and the multiple computing centres providing SNIC resources with a unified database of users and projects, as well as support for handling project proposals within different application rounds. Users with different roles (PIs, project members, reviewers, staff etc.) use the portal at <https://supr.snic.se/> to perform their various tasks.

We use SUPR to manage SNIC allocations on LUMI, using a script run every 10 minutes to update Puhuri Core based on the information in SUPR about projects and their members. For a discussion of this, see deliverable D3.

In the remaining sections of this part, we describe what has been implemented since deliverable D3 and future work that remains to be done.

### 3.2. Fetching Accounting Data

As described earlier in this document, usage data from LUMI is pushed to Puhuri Core and stored there per month, project and type (CPU, GPU and Storage for the moment).

This data is available from Puhuri Core via the `/api/marketplace-component-usages/` API as described at <sup>3</sup>.

Based on that we have created a script that runs every hour, and iterates over the LUMI projects in SUPR. For each project, we ask Puhuri Core for usage data, filtering on `resource_uuid` and `date_after` to get the data for the project specific number of days into the past. By default, we ask for data 45 days into the past, to get the current month and any adjustments to the preceding month.

SUPR already has a database object `ResourceProjectMonth` that stores usage on a certain Resource (in this case LUMI-C, LUMI-G or LUMI Storage) for a certain Project during a calendar month, so we use this to store the data from LUMI by creating or updating the appropriate object.

---

<sup>3</sup> <https://puhuri.neic.no/resource-allocators/#reporting>.

### 3.3. Fetching Project Names

The LUMI project names (used as the SLURM account name, etc) are pushed from LUMI to Puhuri Core and are available as `backend_id` via the `/api/marketplace-resources/` API that we already use to compare allocation limits between SUPR and Puhuri Core.

We have added a new field in the SUPR Project object to store the LUMI project name, and added a feature to the sync script that makes sure that we try to fetch this on every run as long as the field is empty in SUPR.

### 3.4. Fetching Usernames

The LUMI usernames are pushed from LUMI to Puhuri core and are available in the `username` field in the data returned by the `/api/marketplace-offering-users/` API as described at <sup>4</sup>. The same record also contains a `user_uuid` field that can be used to map to the Person in SUPR and create a SUPR Account object, which connects a Person to a Resource and records the username. We have added a new script to fetch the LUMI usernames and compare them to existing SUPR Accounts and do the necessary changes.

There are two complications to handle:

- The SUPR Account object contains a status field that should be "enabled" or "disabled", but there is no corresponding flag in the data from Puhuri Core. For now, we handle this by setting the status to "enabled" when we see the user in the data from Puhuri Core, and by setting it to "disabled" on existing LUMI Accounts in SUPR that we no longer get from Puhuri Core.
- For now, we can get away with considering all usernames we get via this API to be LUMI user names. If SUPR and Puhuri Core will be used together to handle other service provides than LUMI, we will need to add code to look at the returned offering names to know what SUPR Resource to update Accounts for.

### 3.5. Future Work

We still need to work on presenting project usage and the corresponding allocation information in the SUPR portal for users and staff. We will have to sort out complications related to this:

---

<sup>4</sup>

<https://puhuri.neic.no/SDK%20guide/allocation-management-ra/#listing-associations-between-users-and-resources>

LUMI uses different allocation units as compared to the SNIC centre resources that SUPR has handled up till now (LUMI uses CPU/GPU hours for a whole project while SNIC resources use CPU/GPU hours per month; LUMI uses TB hours while SNIC resources use TB or GB without hours)

We also need to figure out how to handle requests for project extensions for LUMI projects, when the PI wants to continue the same research on LUMI for another year. The normal way in SUPR for SNIC projects is for the PI to request a new project (via a new proposal), marked as a continuation of the old project. If the new project is approved, there will be a new project name in SUPR (and therefore also a new Slurm account name) to be used for running jobs, but any storage will be kept in the same place as before, as that location is inherited by continuation projects via the SUPR "directory name" feature.

LUMI does not have a similar "directory name" feature, so if a new project is created in Puhuri Core, the storage will have to be moved by the users from one place to another on LUMI.

In SUPR, we may work around this by letting the new SUPR project be associated with the same Puhuri Core/LUMI project as the old SUPR project. This works, but we then run into problems with the semantics of the requested allocation in the new proposal: Is it an update of the total limit for the one and only LUMI project? Is it an addition to the allocation in the earlier SUPR project that should be added to the one and only LUMI project? In that case, should it be on top of the earlier allocation, or on top of what has actually been run during the old SUPR project?

For SUPR, we need to figure out how to cope with the semantics above; decide not to support continuation projects that keep the same storage location or introduce a feature in Puhuri Core/LUMI to inherit storage locations between earlier and later projects.

# References

[D3] LUMI integration to Puhuri. Puhuri Deliverable 3 <https://zenodo.org/record/4727686>

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

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