PLG Portal - platform for managing distributed computing resources in a federated infrastructure

Andrzej Zemła, Marek Kasztelnik

ACK Cyfronet AGH
EGI 2024 conference
30.09-4.10 2024, Lecce

Partners:

PLGrid Consortium

Established in 2007 under an agreement between the largest polish HPC centers in order to build and manage distributed computing infrastructure for science - **PLGrid infrastructure**

Centre of Informatics Tricity Academic Supercomputer and network

CI TASK
Gdansk

Interdisciplinary Centre for Mathematical and Computational Modelling at University of Warsaw

PCSS
Poznan

ICM UW
Warsaw

NCBJ
Swierk

Poznan Supercomputing and Networking Center

National Centre For Nuclear Research

WCSS
Wroclaw

Wroclaw Centre forNetworking and Supercomputing

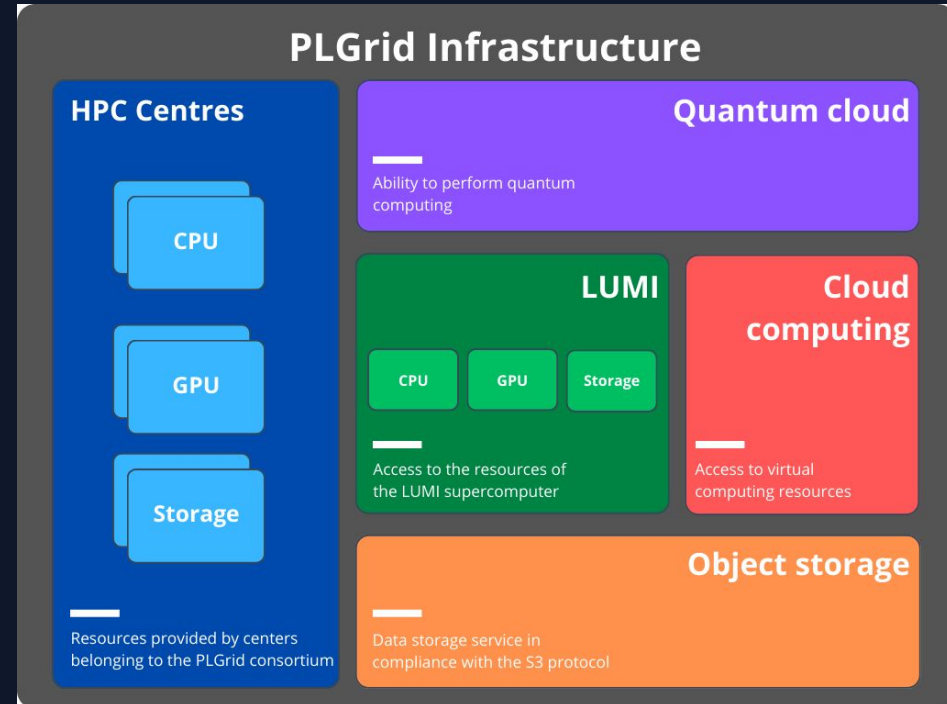Academic Computer Centre Cyfronet AGH

CYFRONET
Cracow

Project aims

Platform for integration of polish distributed computing resources with the infrastructures of European EuroHPC JU resources and other specialized platforms (single point of access)

- Authentication and authorization infrastructure (AAI)
- Integration platform https://portal.plgrid.pl
  - user management portal
  - grant (project) application process
  - resource allocation
  - integration and management of services
- Controllers

PLGrid Infrastructure

HPC Centres
CPU
GPU
Storage
Resources provided by centers belonging to the PLGrid consortium

Quantum cloud
Ability to perform quantum computing

LUMI
CPU
GPU
Storage
Access to the resources of the LUMI supercomputer

Cloud computing
Access to virtual computing resources

Object storage
Data storage service in compliance with the S3 protocol

Authentication and Authorisation Infrastructure

Authentication and Authorisation Infrastructure (AAI) https://sso.plgrid.pl

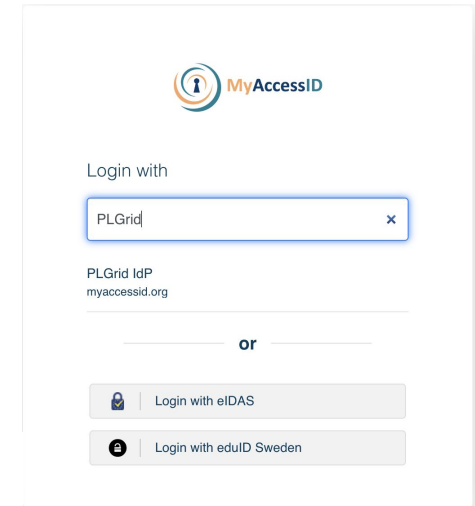Open source solution -
https://www.keycloak.org/

- Centralized Management
- Single-Sign On - login once to multiple applications and services
- Standard Protocols - OpenID Connect, OAuth 2.0 and SAML 2.0
- Easy integration with services
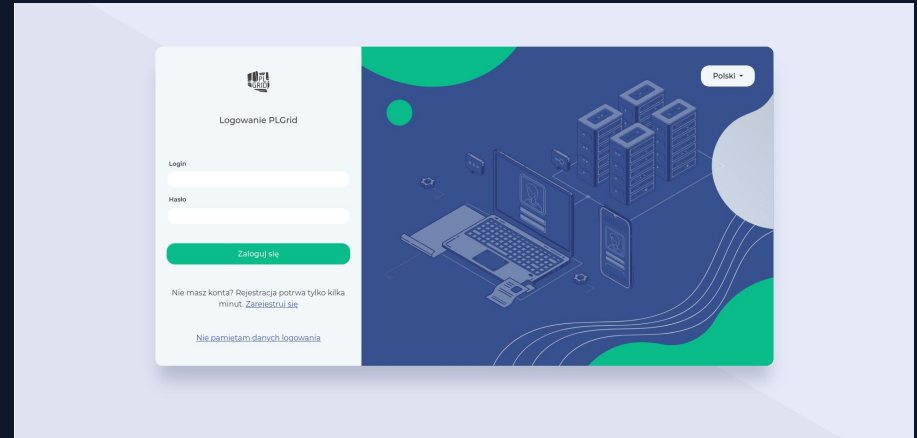
Integration with MyAccessID

Integration with MyAccessID is necessary to be able to create allocations on the LUMI supercomputer

- **added required attributes by MyAccessID,**

- **integrated with the PLGrid IdP,**

- **ability to assign allocations to specific users by controller,**

- **the user have the ability to add SSH keys, which allow access to the login nodes.**

Kerberos & SSH keys

Save

| Public ssh Key: | ➕ | User's public SSH key. |

Integration platform
https://portal.plgrid.pl

Profile - comprehensive user data

Providing a verified email in the domain of the scientific unit as the essential step to gain full access to the resources.

Details about user affiliation is an integral and crucial component of the user profile.

The possibility of providing one of the most popular scientific database identifiers like ORCID number or ResearchID.

Affiliation - user association

Flexible and intuitive affiliation type selection system.

Different permissions depending on the type of affiliation.

Verification of affiliation based on the RAD-on (system of information on science and higher education).

Access for the world of science and business.

Research teams

- easy management of your team

- possibility to invite members via email and by sending an invitation link

- archiving teams from completed projects

Grant lifecycle

The user obtains all the necessary permissions to apply for a grant.

Start

The user creates a draft, fills out the application,
specifies which resources and how many are needed.
The user submits the application for review.

Start   1   Draft ——————

The operator reviews the application. If necessary, it is sent back to the user for corrections. If everything is fine, the application is forwarded to the resource administrator.

Start   **1** Draft ——— **2** Veryfication ———

The resource administrator reviews the application, may propose resource negotiations, and send it back to the user. Once approved, the grant and its resources become active.

The grant lasts up to 12 months. The user can request resource renegotiation, subject to admin approval.

Start   **1**   Draft ——— **2**   Veryfication ——— **3** R   Active ———

The grant ends,
and the user has
30 days to settle it.

Start   **1** Draft ———— **2** Veryfication ———— **3** **R** Active ———— **4** Grant settlement ————

The operator reviews and approves the settlement, closing the grant.

Start   **1** Draft ——— **2** Veryfication ——— **3** (R) Active ——— **4** Grant settlement ——— **5** Ended

Resource allocation

Customization of the resource allocation process

Integration with different types of resources, especially from other infrastructures, must ensure consistency and interoperability.

We cannot assume that the procedural and technical requirements will be the same.

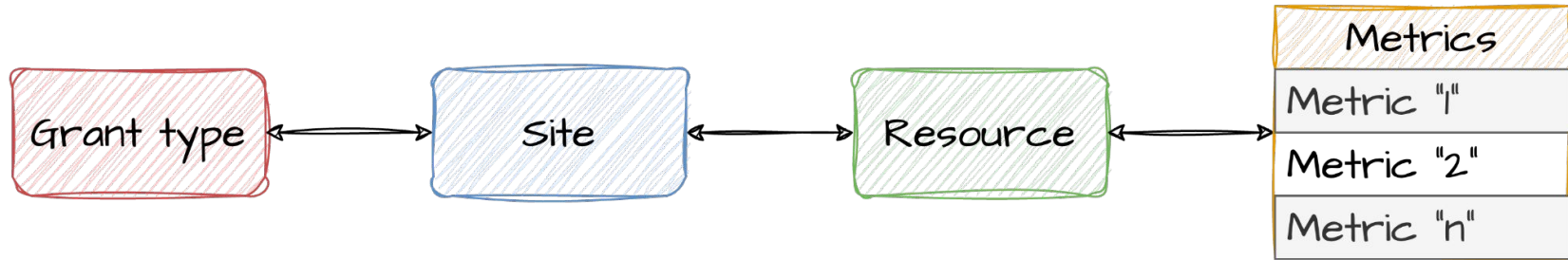Main points of customization:

application process/form

resource metrics

granting policy

Glossary

```
┌──────────────┐       ┌──────────────┐       ┌──────────────┐       ┌──────────────┐
│  Grant type  │ ◄───► │     Site     │ ◄───► │   Resource   │ ◄───► │   Metrics    │
└──────────────┘       └──────────────┘       └──────────────┘       ├──────────────┤
                                                                      │  Metric "1"  │
                                                                      ├──────────────┤
                                                                      │  Metric "2"  │
                                                                      ├──────────────┤
                                                                      │  Metric "n"  │
                                                                      └──────────────┘
```

The class of grants with common set of policies, mostly related to infrastructure (ex. PLGrid, LUMI)

The exact HPC system, tier Cloud, etc... (ex. Athena, BEM, LUMI, Object Storage)

Specific type of computing or storage elements (ex. CPU, GPU, capacity, VM)

Requested resource parameters

Grant type

Access to specific types of resources may have different application policies:

- application process
    - continues recruitment
    - regular calls
    - fast/test access (pilots)
- time frame
- review process (if required)
- settlements/reports
- evaluation survey

i	Basic information
i	Bundles
i	Bundle categories
i	Calls
i	Grant form extensions
i	Review types
i	Grant evaluation

Grant type

Basic template created on the basis of plgrid grants

Splitted into sections (basic informations, resources, grant details, etc.)

Can be extended with various types of fields, ex:
- rich text
- string
- select
- rating

Grant details

Grant start date *

07.12.2023

The application must be reviewed and evaluated before the grant is allocated. The above date will be considered as the actual start date of the grant. In case of prolonged negotiations, the grant will be launched as soon as the application is accepted.

Duration in months *

12

Field of Science *

Please select field o science

Research purpose *

Substantive description of the research, public abstract - information about planned works under the grant and what the user wants to achieve. In this field you can add graphic files, links, pdf files. (min. 250, max. 1000 characters).

Computational readiness *

Justify the requested computational and storage resources. Provide basis on which the requested resources were estimated. Describe the computational techniques and platforms that you will use, including application code names (together with exact version), programming languages, libraries, and other software used. Describe also parallelization and scalability aspects, including parallel programming system used (e.g., MPI, PGAS, "embarrassingly parallel"). If possible, provide data for your application parallel performance, speedup and scalability.

Attachments

Drag and drop to upload or click here to browse

10 MB file size maximum. Allowed file type: pdf.

Previous publications of the user on the subject being the basis of the grant.

Save changes

Site

Site has a relationship to the type of grant, provides various resources and has dedicated administrators. They can:
- **edit basic information like name, logo,**
- **accept the resources under grants,**
- **add public API authentication key (JWT),**
- **define resources:**
  - **postfix,**
  - **operator acceptance criteria.**

Resources:

ACK Cyfronet Object Storage
DC-Nawojki

Key *

dc-nawojki

NOTE: All white chars will be converted to - (dash) and capital letters to lowercase due to normalization.

Postfix

Operator acceptance criteria

CPU plgplgwolframprom-cpu

GPU plgplgwolframprom-gpu

Full site name in Polish *

ACK Cyfronet Object Storage

Full site name in English *

ACK Cyfronet Object Storage

Resource

Metrics
Metric "i"
Metric "2"
Metric "n"

Different types of resources can have different metrics and/or units
- CPU - kilo-hours
- GPU - hours
- QPU - seconds!!
- storage capacity- GB (static pool) or TBh
- cloud resources …. (# VM, VM spec, network spec… the whole bunch of metrics)

Additional metrics:
- memory (per slot or working node)
- specific cpu/gpu type in heterogeneous machines
- max walltime

And any combinations, names whatever we can think of.

The platform has to ensure flexibility!

ACK Cyfronet Cloud

Cloud Cyfronet

CLOUD-3.0 - Szkic

Liczba publicznych adresów IP

10

Liczba rdzeni

4

Maksymalna pamięć RAM

4   GB

Maksymalna pamięć dyskowa wolumenów

4   GB

Maksymalna liczba wirtualnych maszyn

10

Anuluj   Stwórz alokację

Metrics

Metrics
Metric "1"
Metric "2"
Metric "n"
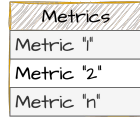
Resource

- **resource metrics scheme is fully customizable**
- **metric attributes:**
  - **name**
  - **value**
    - **types:**
      - **int (with min and max bounds)**
      - **string**
      - **select box**
  - **unit value**
- **It allows to define virtually any type of resources**

Based on the metrics schema, an appropriate API is generated

```
[
  {
    name: String,
    grantName: String,
    resource: String,
    site: String,
    pilot: Boolean,
    start: Date,
    end: Date,
    lastChange: Time,
    status: Enum(accepted, blocked),
    parameterValues: [
      {
        key: String,
        value: String/Number
      },
      ...
    ]
  },
  ...
]
```

Integration scheme
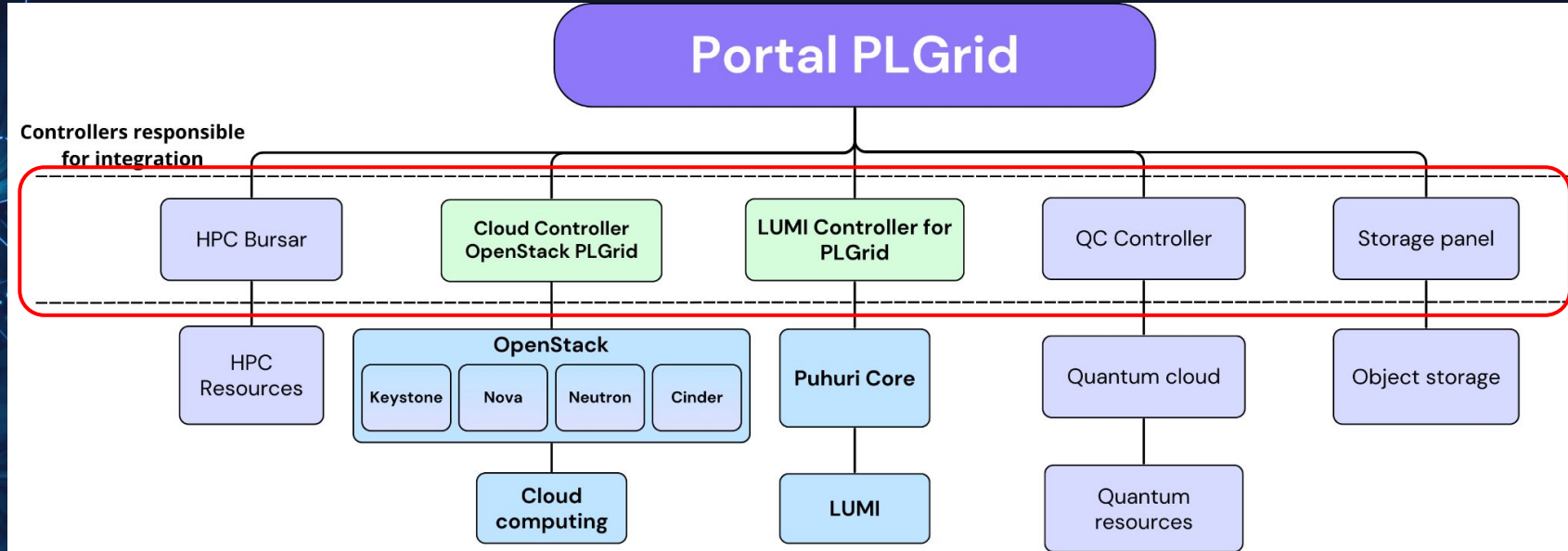
Why controllers layer approach?

Security

An external app **run periodically based on data from the API** is more secure due to less access to it from the outside world.

Moreover, the web application does not need to have information enabling authentication to external applications with the administrator role.

Consistency

The use of controllers independent of the main web application allows you to **maintain order** in the code and **consistency in the operation** of the application.

Thanks to this, the PLGrid Portal does not have to contain unnecessary and non-generic functionalities needed for integration with other e-infrastructures.

Maintenance

Maintaining a small controller independent of the main web application is very easy.

The local configuration of such a controller is independent of the PLGrid Portal and requires the installation of only those dependencies that are used in the controller.

Flexibility

Controllers provide a **high level of flexibility** in mapping information returned by the PLGrid Portal API.

This makes it **easier to transform** the received information to meet the requirements of external e-infrastructures.

Possibility to choose the best-suited technology

In the case of some controllers, dependencies available only for specific technologies are used for integration purposes.

One example is a CCOP controller, designed to integrate the PLGrid Portal with cloud computing resources. For this purpose, it uses, among others, the OpenStack SDK3 library, the only official version of which was created for Python.

Creating separate controllers allows you to select technology and tools in such a way that their implementation is as simple as possible.

Team

Marek **Kasztelnik**

Konrad **Perłowski**

Bartłomiej **Bańda**

Mieszko **Cholewa**

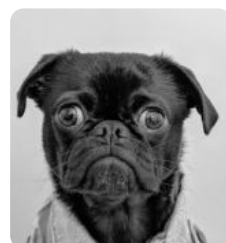Arkadiusz **Zięba**

Radosław **Duma**

Jakub **Perlak**

Magdalena **Paluchowska**

Łukasz **Kitowski**

Jacek **Chudzik**

Partners: CYFRONET · ci task · PCSS · icm · WCSS · NARODOWE CENTRUM BADAŃ JĄDROWYCH ŚWIERK

Thank you for your attention

Outline

PLGrid is a nationwide computing infrastructure designed to support scientific research and experimental development across a wide range of scientific and economic fields. PLGrid provides access to supercomputers, quantum computers, specialized accelerators for artificial intelligence, cloud computing, disk storage, optimized computing software and assistance from experts from the entire Poland. The Polish PLGrid infrastructure is managed by the PLGrid Consortium, established in January 2007, which includes the following computing centers: Cyfronet, ICM, PSNC, CI TASK, WCSS, NCBJ.

In order to make it easier for users to use the available distributed resources, it was necessary to create a centralized platform that includes many applications, tools, and solutions, with the PLGrid Portal as its main component. The platform has been developed since the beginning of the PLGrid Consortium, going through successive new versions. Thanks to the experience gained over 10 years, it has become a mature and flexible solution. This allows us to easily adapt to changing requirements, which makes us able to effectively respond to new challenges in terms of both user and operational convenience in federated infrastructure.

As the main application in PLGrid Infrastructure, the Portal PLGrid consists of many elements from creating an account to requesting distributed resources through PLGrid grants. All the necessary functionalities for the User like creating and managing an account, affiliations, subordinates, teams, services (access to resources), applications, and ssh keys. However, it is the process of requesting resources that is specific. The user fills out a grant application, which is negotiated with resource administrators - and after the grant is completed, it must be settled. From our perspective, the most important thing was to realize such Portal that would be flexible. We have a variety of types: accounts, teams, services, and grants. In an easy way, new types of resources can be defined, that will have other limitations. For example, a specific account type cannot create a specific grant type or a given type of grant has other fields that are required to be provided by the user. On the operational side, we have many roles, where the main like Resource/Service Administrator have dedicated web views, the ability to replicate data to LDAP, and API access which allows you to synchronize all data among various HPC centers and different types of resources (computing, storage, object storage, cloud, etc).