

**EGI-Engage**

Design of the EGI Service Registry and Marketplace

D3.2

|  |  |
| --- | --- |
| **Date** | 16 February 2016 |
| **Activity** | JRA1 |
| **Lead Partner** | SWING |
| **Document Status** | DRAFT |
| **Document Link** | <https://documents.egi.eu/document/2658> |

Abstract

This document details the features of the “EGI Service Registry and Marketplace” and analyses existing open-source tools and extensions to implement the EGI marketplace demonstrator. It is based on the concept of EGI marketplace, defined in the D2.3, and in the envisaged requirements. The requirements were collected through an ample consultation process, involving large resource providers, projects and research communities and also other e-Infrastructures.

The EGI marketplace has the ambition of becoming the platform where an ecosystem of EGI-related services can be promoted, discovered and shared, including EGI offered services as well as discipline and community specific tools and services enabled by EGI and/or provided by third parties under defined agreements.

**COPYRIGHT NOTICE**



This work by Parties of the EGI-Engage Consortium is licensed under a Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/). The EGI-Engage project is co-funded by the European Union Horizon 2020 programme under grant number 654142.

**DELIVERY SLIP**

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Name*** | ***Partner/Activity*** | ***Date*** |
| **From:** | Dean Flanders | SwiNG | 27 Jan 2016 |
| **Moderated by:** | Diego Scardaci | EGI.eu | 27 Jan 2016 |
| **Reviewed by** | TBD  TBD  TBD | TBD  TBD  TBD |  |
| **Approved by:** | AMB and PMB |  |  |

**DOCUMENT LOG**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Issue*** | ***Date*** | ***Comment*** | ***Author/Partner*** |
| **V1** | 31 Dec 2015 | First version of Table of Contents | Dean Flanders (SwiNG) |
| **V2** | 27 Jan 2016 | Second version | Dean Flanders (SwiNG) |
| **V3** | 16 Feb 2016 | Third version | Dean Flanders (SwiNG) |

**TERMINOLOGY**

A complete project glossary is provided at the following page: <http://www.egi.eu/about/glossary/>

**Contents**

1 Executive summary 5

2 Introduction 6

2.1 Concept 7

3 Requirements 8

3.1 Service Management 8

3.1.1 Service Provider 8

3.1.2 User 12

3.1.3 Directory Manager 17

3.1.4 Main Administrator 19

3.2 Requirements: Access Management 19

3.2.1 Service Provider 19

3.2.2 Directory Manager 21

3.2.3 User 22

3.2.4 Directory Administrator 23

4 Assessment of the technologies to implement the EGI Marketplace demonstrator 24

4.1 Further considerations for the assessment of the technologies 24

4.2 Examples of existing platforms 24

4.3 Tools evaluated 25

4.4 TJRA1.2 High Level Evaluation 26

4.5 Comparison requirements 28

4.6 Consultation of experts on developing, designing or operating user-facing tools and/or a marketplace 28

4.6.1 Adequacy of the solution against requirements 28

4.6.2 Possible costs in terms of licenses and support 29

4.6.3 Is the solution supportable in terms of expertise within EGI 30

4.6.4 Final outcome of the consultation 30

4.7 Final consideration and choice of the solution to implement the EGI Marketplace demonstrator 31

5 Technical Architecture 32

5.1 High Level Architecture. 32

5.2 Key components 32

5.2.1 Directory 32

5.2.2 Portal 33

5.2.3 Service Catalogs 33

5.2.4 Modules 33

5.3 Technical Summary 34

5.4 Open IRIS integration in the EGI tools eco-system 34

5.4.1 AppDB technical integration 35

5.4.2 GOCDB technical integration 35

5.4.3 e-GRANT technical integration 35

6 Development Roadmap 36

6.1 Design and mockup of public facing marketplace (task 1) 36

6.2 Development of public facing marketplace (task 2) 36

6.3 Define technical integration of AppDB, GOCDB, and eGrant (task 3) 37

6.4 Integration work for of AppDB, GOCDB, and e-GRANT (task 4) 37

6.5 Populate service catalogue with EGI services (task 5) 37

6.6 Test release of first version of public facing marketplace for EGI (task 6) 37

6.7 First release of the EGI Service Registry and Prototype (task 7) 37

7 Summary 38

Appendix I. Essential features of a marketplace based on user surveys and interviews 39

Appendix II. Usage scenarios 40

Appendix III. Requirements from sections 2 and 3 mapped to tools evaluated 42

Appendix IV. Consultation of experts to assess the technologies to implement the EGI marketplace 47

# Executive summary

This document details the features of the “EGI Service Registry and Marketplace” and analyses existing open-source tools and extensions to implement the EGI marketplace demonstrator. It is based on the concept of EGI marketplace, defined in the D2.3, and in the envisaged requirements. The requirements were collected through an ample consultation process, involving large resource providers, projects and research communities and also other e-Infrastructures.

The EGI marketplace has the ambition of becoming the platform where an ecosystem of EGI-related services can be promoted, discovered and shared, including EGI offered services as well as discipline and community specific tools and services enabled by EGI and/or provided by third parties under defined agreements.

# Introduction

The primary goal of the EGI Service Registry and Marketplace is to ensure efficient resource usage at the institutional, national, and international level. This objective can be achieved offering features that:

* Allow cost sharing with accounting, billing, and enabling of fair usage of resources.
* Facilitate resource discovery at the institutional and inter-institutional level.
* Allow researchers and institutions to focus on value creation as opposed to maintaining redundant resources.
* Researchers can discover expertise that can be tapped into based on usage of resources registered.
* Remove administrative burdens from technology platforms allowing them to focus on technology delivery.
* Increase competitiveness by providing a low cost of entry to expensive technologies for small academic institutions and businesses.
* Facilitate inter-disciplinary research by providing access to technologies typically considered outside of a particular field.
* Avoid re-developing the same solution (tool duplication).
* Provide opportunities for collaborative improvements of services and resources.
* Possible reduce costs by facilitating complex application implementation and integration (e.g. issuing of persistent identifiers, providing links between resources and services).

The document is in the following structure: introduction, examination of requirements, evaluation of tools, technical architecture, roadmap and summary.

## Concept

Figure 1. The EGI marketplace - Actors

The EGI Marketplace should provide functionalities necessary for bringing together offering and demand for making research.

These functions include basic services for registering business entities, publishing and retrieving offerings and demands, searching and discovering offerings according to specific research communities requirements as well as lateral functions like reviewing, rating and recommending.

The Marketplace would provide researchers with a uniform interface to discover and match application and service offerings from providers and sources (e.g. published by different stores) with demand of consumers.

The best option envisaged in D3.2 is to go with an offering of a “marketplace as a service” to complement the EGI marketplace business functions where needed. This vision would allow resource providers (groups, facilities, organizations, etc.) have a self-service platform to register and manage their resources creating virtual pools of resources spanning groups, organizations, consortiums, collaborations, and the research community in general.

# Requirements

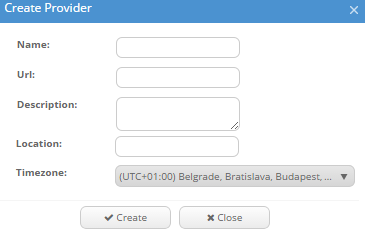
This section shows, via example screens, the marketplace requirements identified in D3.2, which were based on the original EGI use case examples before the project was initiated (Appendix II). This representation has been done to facilitate the work of the experts in portal development, which has been consulted to assess a set of technologies to implement the marketplace demonstrator.

Please, refer to section 3.5 of the D3.2 for a full list and description of the requirements.

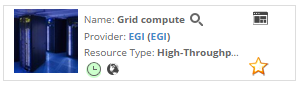
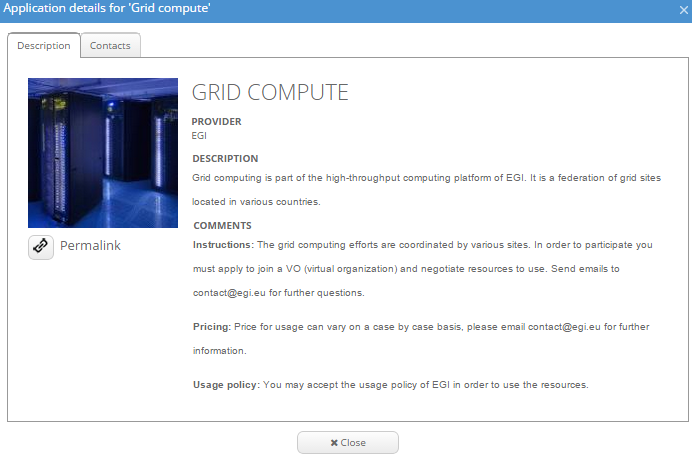
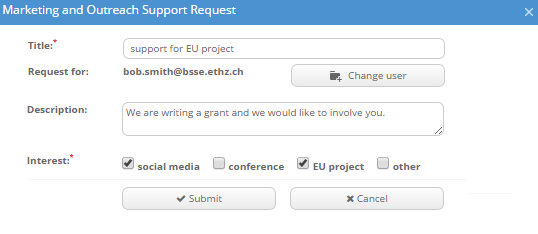
## Service Management

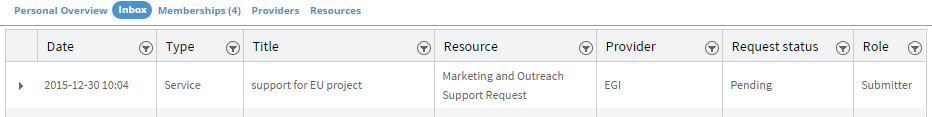
This section covers provisioning of services and service management requirements. The system should support the registration of any type of resource.

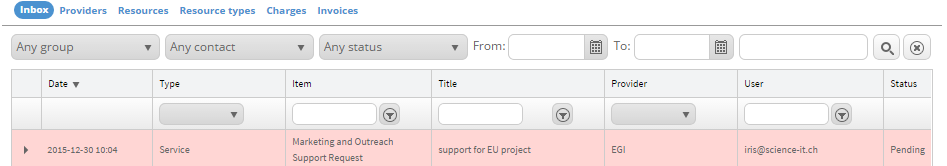
### Service Provider

Service provider perspective of service management.

#### Service Registration

* Users can register a resource provider.
  + Any user can register a resource provider.
  + There is no cost to register a service provider.
  + It is fully self-service and requires no approval.
* The service provider is able to register a service in the service catalogue and can specify detailed information and display options for his service including:
  + Service name
  + Service description
  + Service instructions
  + Service visibility
  + Assign pricing
  + Usage policy
  + Picture
  + Highlight service
  + Hide a service
  + Check order status
* Check order/request status:
  + Submission of order/request
  + User view
  + Admin view





#### Usage policies and SLAs

Resource providers may apply usage policies or service level agreements to users.

* The service provider is able to manage service level agreements and usage policies for his services:
  + Create service level agreements
  + Modify service level agreements
  + Assign service level agreements to services
  + Modify terms of usage and policies

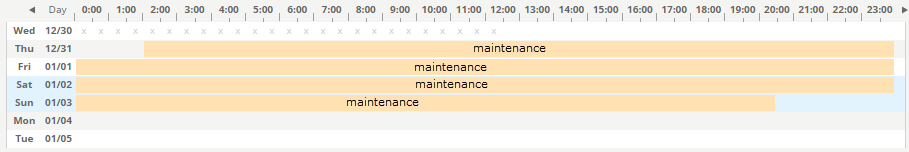


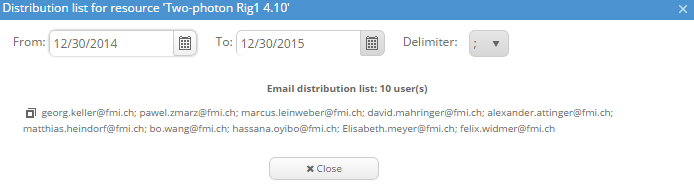
#### Availability

The service provider must

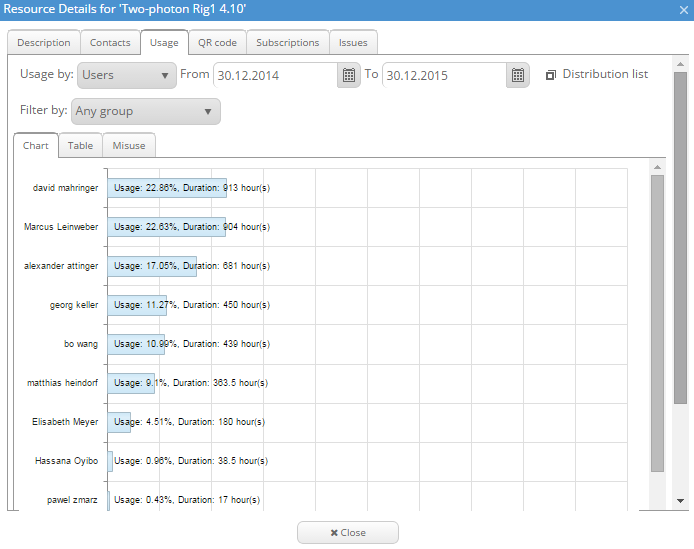
Availability: The service provider can define a time window for when his services are unavailable:

* Define unavailability (Time and Day, e.g. 2015-07-03  8:00 AM to 5:00 PM)
* Inform users





#### Reporting

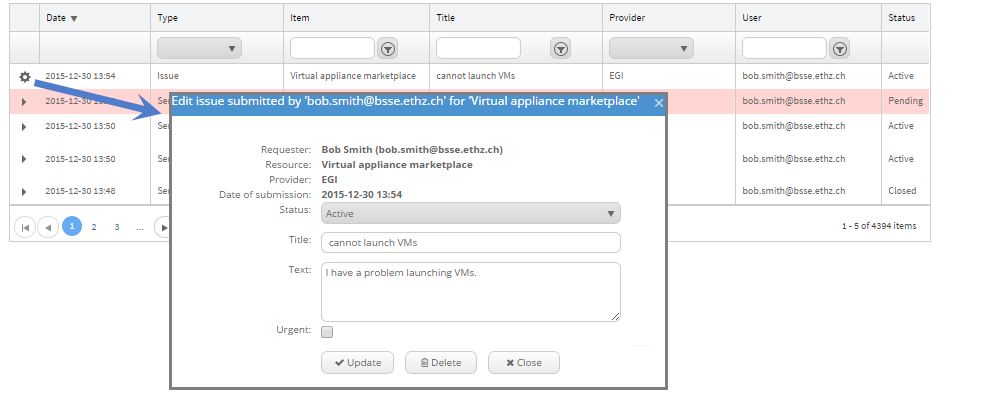
Reporting: The service provider is able to view usage reports for users, groups and communities that are using his service. The usage report shows data about:

* Number of users
* Services used
* Service consumption

#### Tickets

Tickets: The service provider is able to manage his tickets:

* reply to tickets
* close tickets
* delete tickets

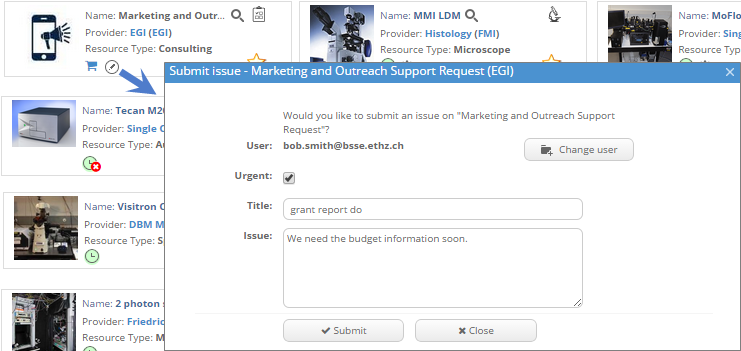


### User

User perspective of service management.

#### Tickets

Tickets: The user is able to open support tickets.



#### C:\Users\deanf\AppData\Local\Temp\SNAGHTML524f9d0a.PNGPublications

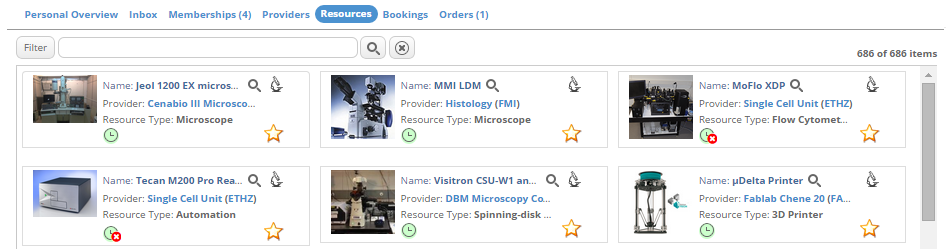
Submission of publications:

* Ability for users to submit publications and link them to a provider(s) and resource(s) used.
* Integration with external databases (e.g. OpenAire)
* Optional features:
  + Show publications linked to resources.
  + Show publications linked to a provider.
  + Search publications list to find providers of interest for a method.

#### List

List: The user can list all services they have access to see details (pricing, how to access, SLAs, associated information, creation date, last update, etc.):

* Services
* Applications
* Appliances



#### Search

Search: The user is able to search and filter for resources on the basis of:

* characteristics
* search term
* virtual organization

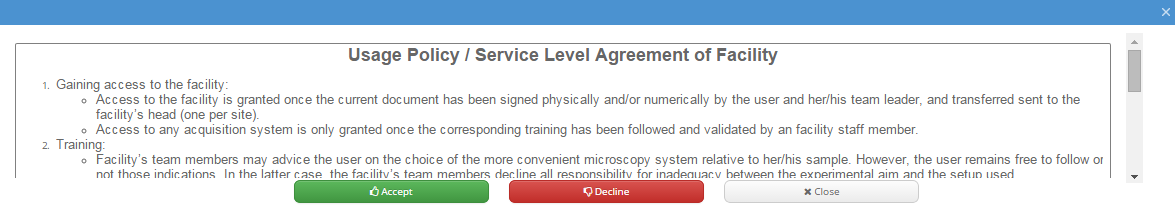
#### Status

Availability (available, unavailable, service outage, maintenance, etc.)

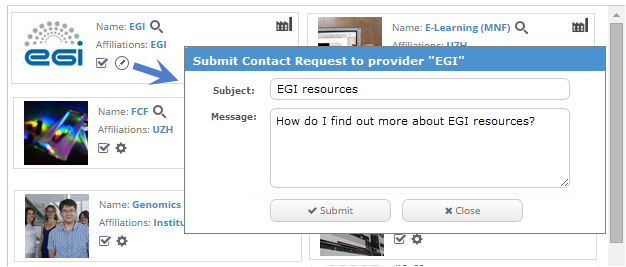
#### C:\Users\deanf\AppData\Local\Temp\SNAGHTML53137977.PNGNegotiation

Policies: Negotiate the SLA through a broker or accept the pre-defined SLA associated to the service

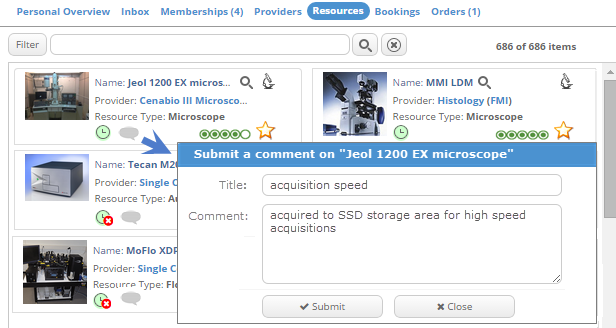
#### Policies

Policies: The user can accept or decline usage policies.

#### Contact

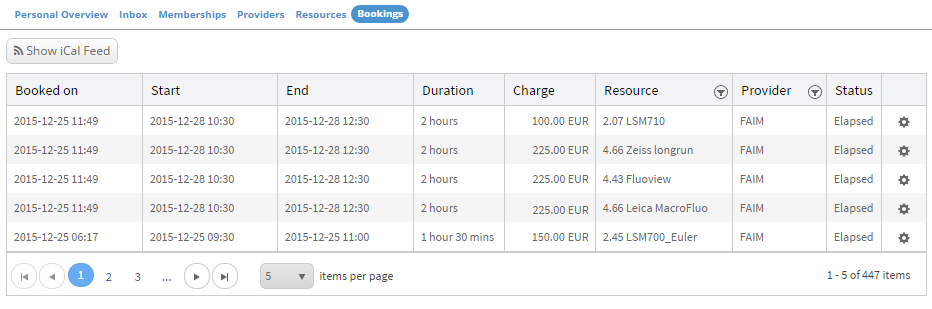
The user is able to contact the service provider via messages.

#### Rating

The user is able to review and rate services and service providers:

* Rating and commenting system

#### Usage

The user is able to view his own usage information including services used.

#### Documentation

Access documentation (knowledge base).

#### Finance

Finance: The user is able to review incurred expenses and pay for a service.

#### Resource access

Request authorization to access one or more services.

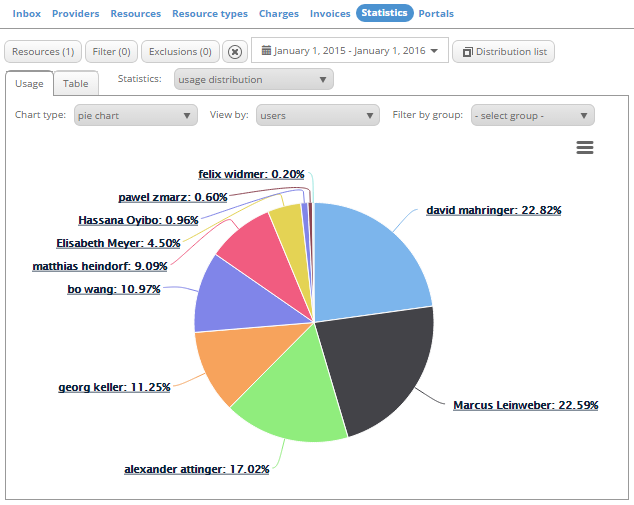
### Directory Manager

Service management from the directory manager perspective.

#### Reporting

##### Resources available

Resources available to his virtual organization / users



##### Resources used

Resources used by his virtual organization / users

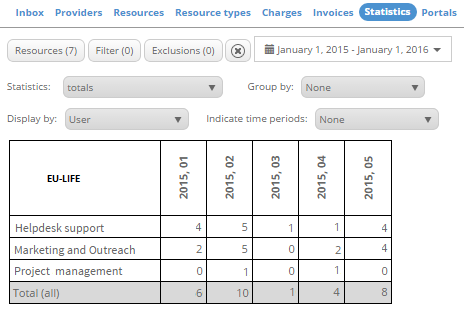
##### Resources booked / used

* Resources available to his virtual organization / users
* Resources used by his virtual organization / users

##### Number of users

Number of users.

##### Services used

Services used.

##### Consumption / usage

Consult consumption / usage.

### Main Administrator

Service management from the perspective of the main systems administrator.

#### Manage provider requests

* Accept resource provider request
* Decline resource provider request (with reason)
* Disable a provider (with reason)

## Requirements: Access Management

This section covers access management requirements. The system should support access management of resources in terms of visibility and actual access to the resource where possible. The access management requirements are based on the those specified in the “Concept of EGI Marketplace” (EGI Engage D2.3), which were based on the original EGI use case examples before the project was initiated (Appendix II). In order to demonstrate the requirements example screens were created with example service providers and services that could be within the system.

### Service Provider

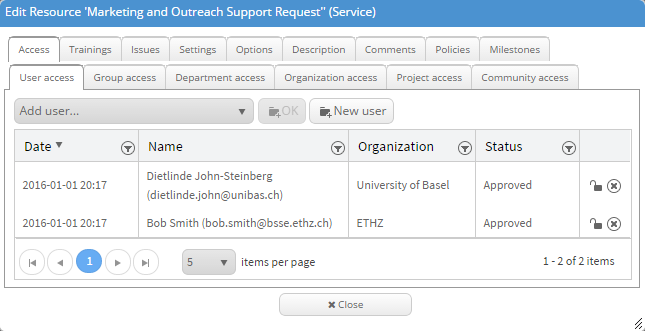
Service provider access management overview.

#### Service access request

The service provider can manage service access requests for an individual user, group or community:

* Allow access
* Decline access (with reason)

#### Directory Management

The service provider can manage all users, groups and communities that have access to his service:

* List of all members
* Remove members
* Invite members

#### C:\Users\deanf\AppData\Local\Temp\SNAGHTML59a087bc.PNGUsage limits

The service provider can manage usage quotas for his resources and assign them to users, groups and communities:

Define quota (e.g. time)

#### Service access

The service provider is able to publish his services in the service catalogue and can manage to whom his services are visible as well as hide services:

* Users
* Groups
* Virtual organizations
* Everyone

### Directory Manager

Manage access to organizations, groups, departments, projects, and communities

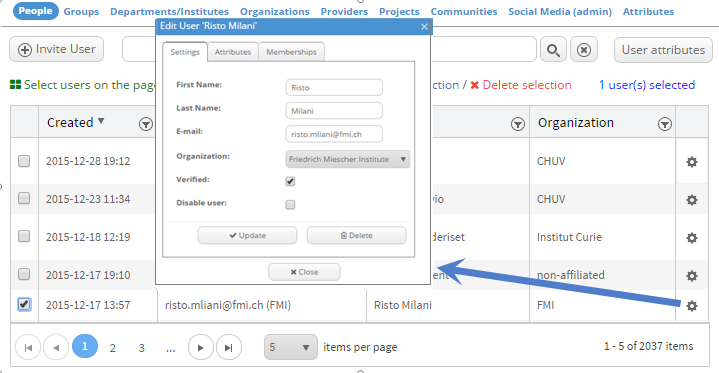
#### Manage directory requests

A directory manager can manage membership requests for users:

* Accept membership request
* Decline membership request (with reason)

#### Manage memberships

The directory manager is able to manage the memberships:

* List of all members
* Remove members
* Invite members
* Promote member to administrator

### User

User access management overview.

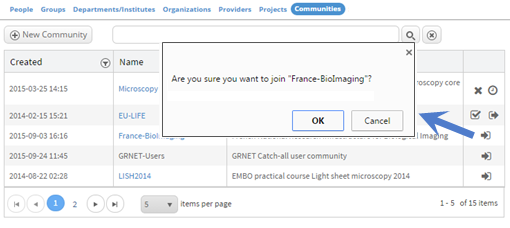
#### Create a provider

Users can request to become a resource provider.

#### Directory search

Search for suitable community

#### Directory requests

The user is able to request access to:

* Services
* Communities / Virtual Organizations
* Groups

### Directory Administrator

Directory administrator access management overview.

#### Manage Directory

Able to manage all aspects of the directory to support directory managers and users.

# Assessment of the technologies to implement the EGI Marketplace demonstrator

There is a variety of possible options for tools to use for this activity, so several were selected for closer examination.

A first analysis has been performed to identify the most promising solutions that can be used by third parties. These has been assessed against the following requirements:

* adequacy of the solution against requirements;
* possible costs in terms of licenses and support;
* solution supportable in terms of expertise within the EGI collaboration

The evaluation has been done both within the TJRA1.2 and through the consultation of experts on developing, designing or operating user-facing tools and/or a marketplace

## Further considerations for the assessment of the technologies

We must review all potential solutions for the establishment of the EGI Engage marketplace solution proof of concept (POC) to be delivered in 2016 (D 3.7, August 31 2016). The requirements defined based on the original EGI use cases within the EGI Engage deliverable D2.3 and the amount of effort available (~4 person months) must be considered in the selection. The cost to operate the tool, effort to develop or extend an existing tool is very limited within the EGI Engage work program, and there should be zero cost of entry to any users of the system. The solution should leverage and work well with existing EGI solutions (AppDB, GOCDB, e-GRANT, etc.) and ideally have knowledge of how to support the tool within the EGI community.

## Examples of existing platforms

Below is a list of existing other existing platforms that are being used for similar purposes.

|  |  |
| --- | --- |
| **Comparable Services** | **Description** |
| GEANT Cloud Catalogue[[1]](#footnote-1) | Catalog of cloud services being developed by GEANT |
| UberCloud Marketplace[[2]](#footnote-2) | HPC cloud marketplace |
| Science Exchange[[3]](#footnote-3) | Various research services |
| Internet2 Net+[[4]](#footnote-4) | Internet2 list of cloud service providers |
| Helix Nebula Marketplace[[5]](#footnote-5) | EU research cloud marketplace |
| UK Gov[[6]](#footnote-6) | UK government list of cloud services |
| Microsoft Azure Marketplace[[7]](#footnote-7) | Microsoft cloud solution |
| Fortissimo marketplace[[8]](#footnote-8) | It is a list of "Experiments" that give examples of services can be used with links to these services. |
| Strategic Service Store [[9]](#footnote-9) | The main goal of the STRATEGIC project is to facilitate organizations and notably public bodies to leverage the benefits of public cloud services. |

However, in most of these cases, the solutions are not available for use by other marketplaces, with the exception of Ubercloud which is using a WordPress based plugin.

## Tools evaluated

Based on the above considerations and on a study of the solutions currently available in the IT scenario, the following products had a high level evaluation done:

* AppDB
* GOCDB
* FIWARE Marketplace Generic Enabler
* Open IRIS
* PrestaShop
* WooCommerce

The EGI Marketplace requirements are quite peculiars and, then, finding a very good match is complex. The tools selected for the evaluation represent well-known solutions for three different categories:

* EGI tools that could be adapted/extended to become a marketplace: GOCDB, AppDB
* Technologies to implement marketplace from other research activities: OpenIRIS, FIWARE
* Generic web tools with feature to implement marketplaces: WooCommerce, WordPress

Identify on which of these categories rely on has been one the main outcome of the assessment. Indeed, the kind of technical work to implement the marketplace could deeply change according to this choice.

## TJRA1.2 High Level Evaluation

This section summarise the outcome of the evaluation of the identified technologies done internally by the TJRA1.2.

* **AppDB**
  + **Pros:**
    - It is an established tool for VMs (69 registered) and software (512 registered).
    - It is well integrated into the EGI ecosystem.
  + **Cons:**
    - It would require effort to full-fill the requirements specified within the EGI Engage marketplace requirements.
    - It has a focus on software and VM registration, and would require extensions in other areas.
  + **Notes:**
    - AppDB is funded by grants and in part by EGI. Development is coordinated by IASA. Usage is free of charge.
* **GOCDB**
  + **Pros:**
    - It is an already established tool for registration of resource providers and the resources they manage.
    - It is well integrated into the EGI ecosystem
  + **Cons:**
    - It would require modification in order to meet the requirements specified in the EGI Engage marketplace requirements.
    - Currently has a focus on grid and cloud resources, limited front facing capabilities.
  + **Notes:**
    - GOCDB is funded primarily by grants and in part by EGI. Development is coordinated by STFC. Usage is free of charge.
* **Open IRIS**
  + **Pros:**
    - Current statistics: 2100 users, ~50 registered resource providers, ~750 registered resources, ~70 unique logins per day, ~200 resource requests per day, 7 organizations actively using the platform, expression of interest and testing from several other organizations including a commercial companies.
  + **Cons:**
    - Currently the front facing portion does not give the impression of a marketplace as a user has to login first to see resources, but this is scheduled for release.
    - It will require some enhancements in order to better support software and VM use cases.
  + **Notes:**
    - Open IRIS is funded by grants and Open IRIS consortium members. Development efforts are led by the Swiss eScience Coordination team, the Swiss NGI, and by an EGI work group. Usage is free.
* **PrestaShop**
  + **Pros:**
    - It has a wide user base and is well established.
    - Many features of the store concept and pricing exist.
  + **Cons:**
    - It is unlikely that the required level of integration into AAI for each individual marketplace participant can be met.
    - The use cases and types of items envisioned in the marketplace are generally outside the current scope of the solution.
  + **Notes:**
    - PrestaShop is an open source solution. Usage is free and extensions can be programmed or bought.
* **WooCommerce**
  + **Pros:** 
    - It is developed with a multi-vendor marketplace focus, though is targeted towards commodity services of small business.
    - The primary advantage of using a solution such is that is based on WordPress and is a commonly used platform with many options available.
  + **Cons:**
    - It is unclear if a large variety of service providers would be willing to manage their items in the WordPress store and would need substantial benefits to do so.
    - It is unlikely that the required level of integration into AAI for each individual marketplace participant can be met.
    - The use cases and types of items envisioned in the marketplace are generally outside the current scope of the solution.
  + **Notes:**
    - WooCommerce is an open source solution based on WordPress. Usage is free and extensions can be programmed or bought.

## Comparison requirements

The developers for AppDB, GOCDB, and Open IRIS were asked to evaluate their tool and indicate what features their tool have in common with the marketplace or would not require substantial effort to modify to add their capabilities (Appendix III). The table below summarizes the matches in each category.

This analysis has been performed only against a sub-set of the solutions. For this reason, it will be used only for a direct comparison between two of the tools analysed.

Table 1.How GOCDB, AppDB and Open IRIS matches EGI marketplace requirements.

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Area** | **Application Match** | | |
| **GOCDB** | **AppDB** | **Open IRIS** |
| Service Provider | 20 | 29 | 50 |
| Directory | 20 | 12 | 42 |
| **Total** | 40 | 41 | 92 |

## Consultation of experts on developing, designing or operating user-facing tools and/or a marketplace

An external evaluation was performed of the above tools by eight people, experts on developing, designing or operating user-facing tools and/or a marketplace, representing different academic organizations: Cyfronet (1), EGI (2), GRNET (1), INFN (1), Institut Curie (1), MTA SZTAKI (1), VIB (1). They were asked to rate the solutions in the following areas:

* Adequacy of the solution against requirements: to understand how each technology under evaluation fits with the EGI marketplace requirements;
* Possible costs in terms of licenses and support: the cost of the solution is another parameter that has to be taken into account in the final choice;
* Is the solution supportable in terms of expertise within EGI: the availability of expertise related to a technology within the EGI collaboration would allow to develop a marketplace that could be easier maintained.

In the following, an analysis of the experts’ assessment for each of the identified areas is reported. The section is completed by an overall summary of this consultation.

### Adequacy of the solution against requirements

In this first area, Open IRIS and AppDB have been the solutions that obtained a consensus from all the evaluators. Open IRIS has been recognized how the tool that best fits the EGI marketplace requirements being already organised as a service catalogue. Although the AppDB was not designed as a marketplace, its existing features could be extended with some effort to satisfy the identified requirements.

The FIWARE MP Generic Enabler has been also well evaluated by a certain number of experts but the changes it requires to implement the requirements seem greater.

Evaluators assessed in very different ways PrestaShop and WooCommerce. For someone, they are excellent solutions to implement the marketplace, for others, they are structured to serve commercial companies with objectives very far from the EGI ones.

Finally, there is a common agreement on considering major the extensions needed by the GOCDB to satisfy the requirements.

Table 2. Adequacy of the solution against requirements, 1 (poor) – 5 (complete)

|  |  |
| --- | --- |
| **Solution** | **Score** |
| Open IRIS | 33 |
| AppDB | 30 |
| FIWARE MP GE | 27 |
| PrestaShop | 24 |
| WooCommerce | 24 |
| GOCDB | 13 |

### Possible costs in terms of licenses and support

All the solutions under evaluation offer free use and access, then the assessment in this categories mainly focussed on understanding the cost EGI could incur adopting one of these tools.

EGI-based software, AppDB and GOCDB, are already co-funded by EGI and additional cost to support their extension to act as a marketplace should be low.

Another important factor that has been identified to assess the costs in terms of licenses and support is the size of the user base of each tool. Tools with large use base could be supported easier than technologies adopted by few projects. This is also relevant for the long-term sustainability of the solution. In relation to this, OpenIRIS, with its increasing user base and the Swiss Federation support, PrestaShop and WooCommerce, with the large communities behind them, seems the most promising solution. FIWARE MP Generic Enabler seems to not give enough guarantees about its sustainability and the support that could be needed.

For some evaluators, adopting commercial tools (like PrestaShop and WooCommerce) could increase the cost of support since EGI should rely on external organizations to achieve its aims.

Finally, one experts raised the point that no one of the not commercial solutions (AppDB, GOCDB, FIWARE MP Generator and OpenIRIS) guarantees the support of thousands or hundreds of thousands of users via high-availability (HA) configurations.

Summarising, with regard this metric, there is an agreement to give a slight preference to EGI based tool. Commercial tools (PrestaShop and WooCommerce) received the lower evaluations since consultancy could be needed to extend some of their features and the licenses cost could be changed by external organizations.

Table 3. Possible costs in terms of licenses and support, 1 (high cost for EGI) – 5 (low cost for EGI)

|  |  |
| --- | --- |
| **Solution** | **Score** |
| AppDB | 32 |
| GOCDB | 29 |
| Open IRIS | 26 |
| FIWARE MP GE | 23 |
| WooCommerce | 22 |
| PrestaShop | 21 |

### Is the solution supportable in terms of expertise within EGI

In this category, the EGI tools (AppDB and GOCDB) received the best score as expected. However, Open IRIS and FIWARE MP General Enabler has been considered supportable in term of expertise within EGI by the evaluators. For the commercial tools, also in this case, there is a great difference between the different assessments, with experts that believe they can be easily adopted as well-known solutions with a large user base and the others that consider them far from the current EGI expertise.

Table 4. Is the solution supportable in terms of expertise within EGI, 1 (poor) – 5 (good)

|  |  |
| --- | --- |
| **Solution** | **Score** |
| AppDB | 37 |
| GOCDB | 31 |
| Open IRIS | 25 |
| FIWARE MP GE | 24 |
| WooCommerce | 20 |
| PrestaShop | 19 |

### Final outcome of the consultation

Considering the three metrics defined and the evaluations of the experts, the solutions that better fits the EGI needs to implement a marketplace are Open IRIS and AppDB.

Open IRIS has a better match with the EGI marketplace requirements and already offers many marketplace features as it is now. About the cost, evaluators believe that this should be affordable considering that Open IRIS is open source and that its long-term sustainability should be guaranteed by the Swiss NGI through national fund.



AppDB needs extensions to become a marketplace but the cost to maintain it should be low considering it is already funded by EGI, then the additional cost would be only related to the marketplace features. Furthermore, it will be surely supportable within the EGI collaboration.

FIWARE MP Generic Enabler has to be extended too to meet the requirements but the knowledge of this solution in EGI is minor with respect AppDB.

There is a consensus between evaluators to consider major the work to extend GOCDB to become the EGI marketplace although it is a well-known solution for EGI.

Finally, experts has very different opinions on adopting PrestaShop or WooCommerce as tools to implement the marketplace. For someone these solutions are very far from the EGI needs and expansive, others believes that their large base would assure a low support cost and a long-term sustainability.

## Final consideration and choice of the solution to implement the EGI Marketplace demonstrator

In looking at the original requirements specified by EGI before the project started, Open IRIS comes the closest to meeting those requirements as ranked by the external reviewers. AppDB and GOCDB were rated favourably by the reviewers in terms of possible costs and knowledge within EGI as they both are existing EGI solutions. However, they would require modification to reach the originally defined requirements; this modification would be extensive in the case of GOCDB.

FiWARE Marketplace Generic Enabler is not well known within the EGI collaboration and, for this reason, the cost to support it could be high. In addition, its long-term sustainability seems not clear.

The commercial tools PrestaShop and WooCommerce do not have a consensus between all the evaluators. Indeed, some of those believe that this commercial solutions do not fit very well with the peculiar requirements the EGI marketplace should satisfy.

Taking into considerations all the points of the above analysis Open IRIS and AppDB seem the most promising solutions to implement the EGI marketplace demonstrator. The first for its best match of the requirements, the second has well-known EGI tool (low cost for EGI) that could be extended to become a marketplace. Anyway, the direct comparison with respect the requirements between these two solutions, done by the same tool developers, shows that Open IRIS, as it is now, matches a greater number of requirements with respect AppDB (92 vs 41, see section 4.5). Furthermore, the limited budget available to develop the demonstrator advise us to point towards the solution that already satisfies the major number of requirements and could become the EGI marketplace with a minimal effort.

For these reasons, we have decided to adopt Open IRIS as solution to implement the EGI marketplace. Open IRIS is actively developed with the Swiss EGI partner as well as part of a working group of other partners (ETHZ, FMI, Institut Curie, Unibas) and is free for general use.

Then, the EGI marketplace technical architecture depicted in the next section will be based on Open IRIS.

# Technical Architecture

This section describes the high-level architecture of the EGI marketplace and how it maps into Open IRIS and its key components. It also provides a summary of the technical architecture.

## High Level Architecture.

Below is the proposed high-level architecture of the EGI marketplace.

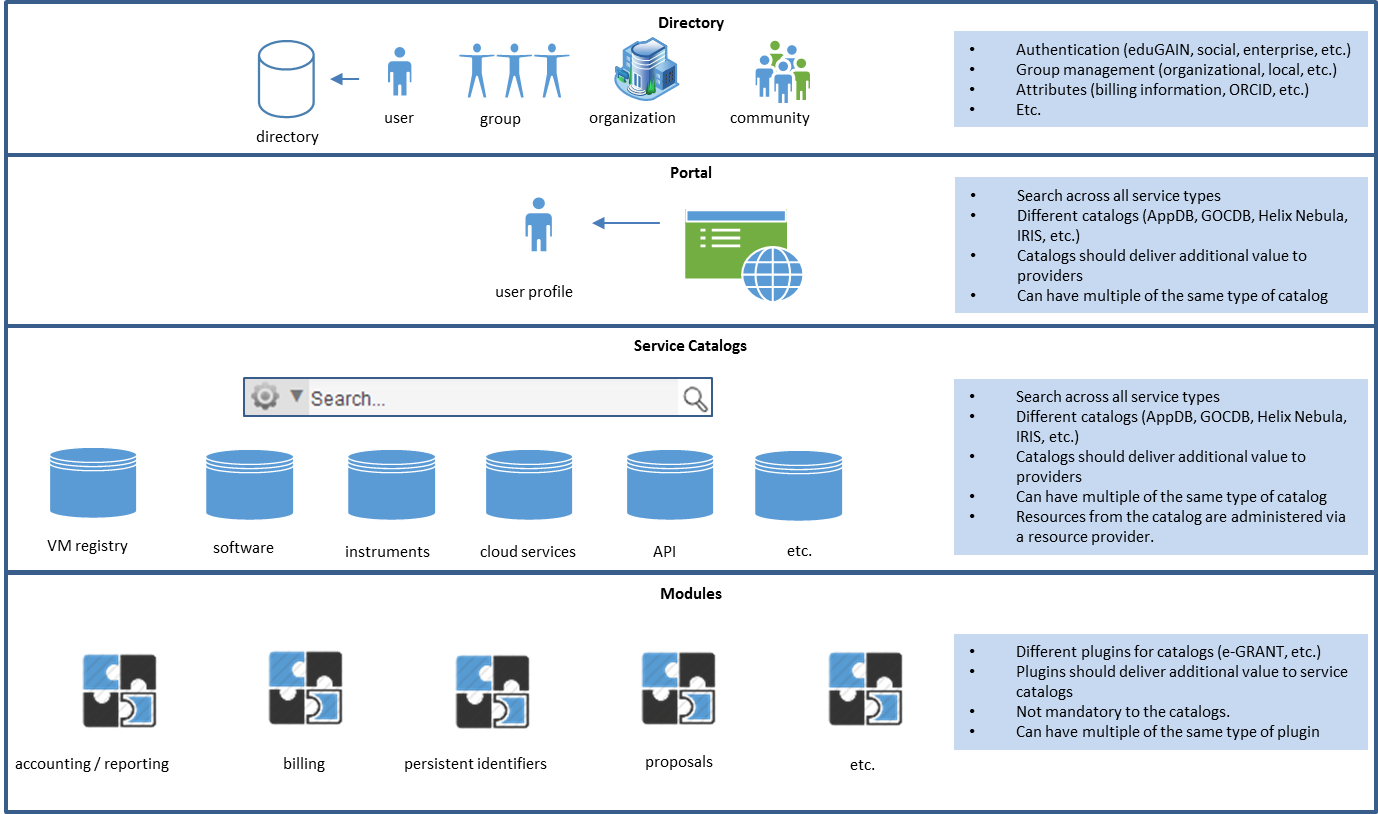


Figure 2. The EGI marketplace - High-level architecture

## Key components

* **Directory**: the directory contains information in terms of user authentication (local, federated, social), groups, departments, organizations, projects, or communities.
* **Portal**: the portal is the environment in which the user logs into the system and presents a view according to their access, and allows for personalization.
* **Service Catalogs**: it is a list of resources (instruments, services, applications, etc.). These are managed by service providers that define the visibility and access levels based on user, groups, departments, organizations, projects, or communities.
* **Modules**: The modules are either internal or external to the system and allows for various functions (accounting/reporting, billing, request management, etc.).

### Directory

The directory contains information specifically on users and groupings of users within an organization in terms or group or department, or across organizations via projects or communities. Each of these groupings have administrators assigned to them. The directory is largely self-service allowing for the creation and management of each type without the intervention of the main administrator, with the exception of the organization, which requires the main administrator of the system to nominate a person to be responsible for organization and manage future administrators for the organizations.

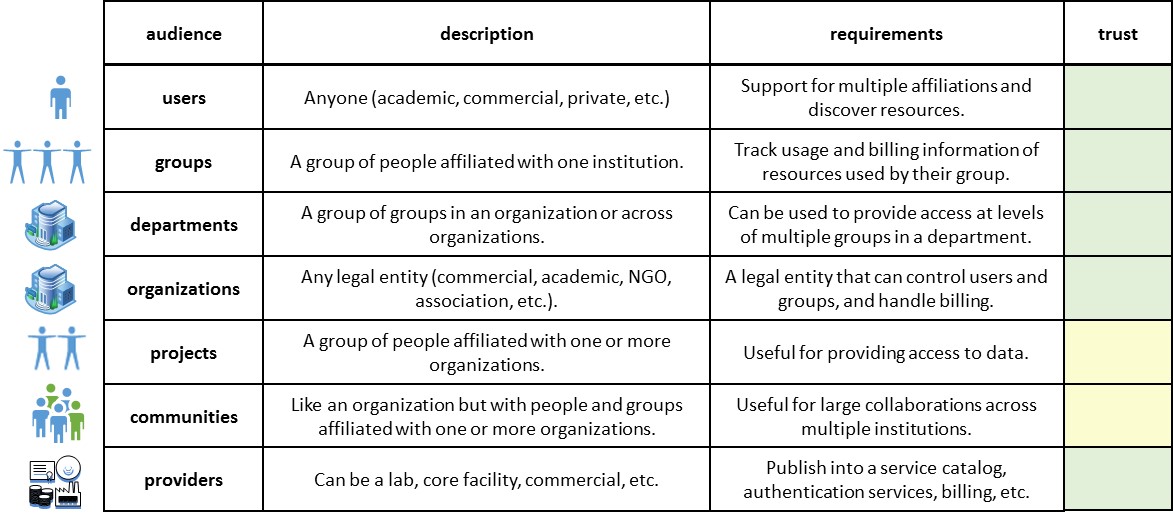


Figure 3. How users are grouped in the EGI marketplace

### Portal

There may be one or more portals defined by resource providers with independent URLs. Resource providers can control their visibility within individual portals. These are aggregations of resource providers as defined by the portal administrator. Within each of these portals, the users can define their preferences and browse resources.

### Service Catalogs

This a collection of resources imported and managed via resource providers. Providers can be affiliated to one or more organizations as well as one or more communities. Creation and management of a resource provider is fully self-service and can manage access to the resource provider and its individual resources on the level of user, group, department, organization, project, or community.

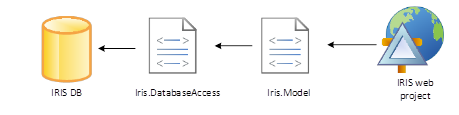
### Modules

These are modules available to resource providers to manage their resources. The use of the modules is optional to resource providers. Resource providers may choose only to use the base functionality to manage their provider and resources. However, they have also available to them additional modules that can be used to help with accounting/reporting, issues, request management, restrictions, billing, scheduling, project management, etc.

## Technical Summary

Dependency injection is widely used in Open IRIS. It is based on Unity data injection container. Unity is used as a controller factory to enable data access command\query design and for application services like logging or sending email.

The data access implementation is based on commands and queries, both of these uses an Entity Framework to access and manipulate data stored in the database. All commands use ICommand and all queries use IQuery. Commands and queries are stored in a model class library, their implemented in a database access class library as shown in the reference schema below:



The web project is a client of data access layer services and does not have knowledge of the database and entity framework.

Require JS is used for dependency injection of view models. This library also enables loading of only the required set of JavaScript files on a web page, and can load asynchronously when needed. Knockout JS is used for building the Model-View-ViewModel (MVVM) like pattern based user interface. This allows for having an independent user interface layout and JavaScript view model.

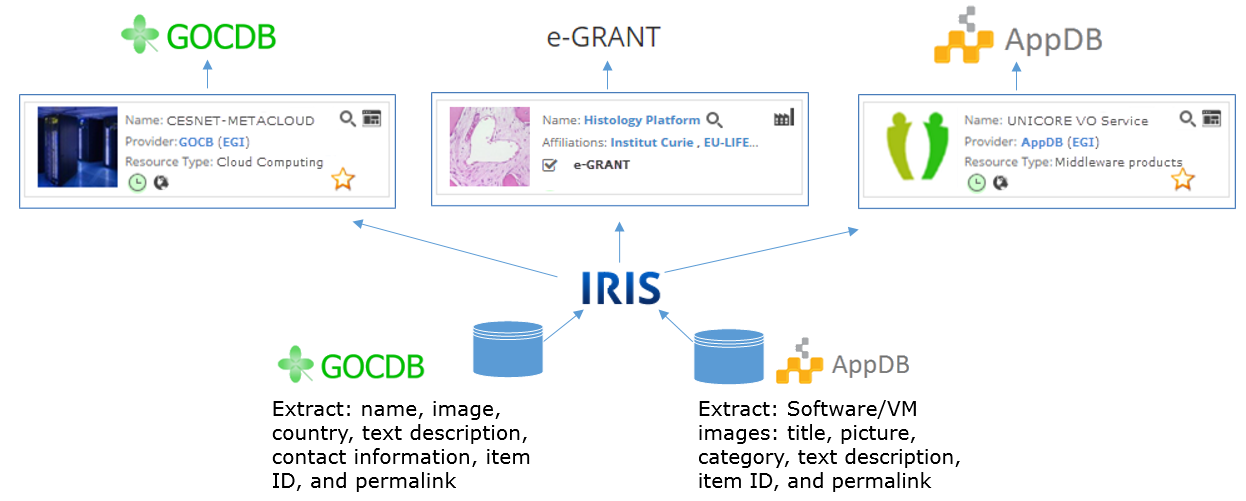
Kendo web controls is the main part of the user interface. It used widely in all parts of the system. Sometimes custom bindings for KnockoutJS are required.

Postal JS library is used to enable communication between independent JS modules. It is designed as a classic service bus library with publishers and subscribers.

All elements are issued a GUID and this is used by the API to get information on the object. Access is controlled via secret key, which can be regenerated and security is handled via Oauth2.

## Open IRIS integration in the EGI tools eco-system

Open IRIS is designed to be integrated with various other services (e.g. instrument databases, application catalogues). The first such use cases for this will be the integration with the EGI tools AppDB, GOCDB, and e-GRANT to create one location to facilitate the discovery of a wide variety of services. As Open IRIS contains resources (e.g. instrument booking) that are used daily by many users this will increase exposure of the contents of AppDB and GOCDB to a wider audience of users. In addition, integration of e-GRANT would allow other resource providers to integrate their offerings into e-GRANT.



### AppDB technical integration

AppDB is a registry for software and virtual machines. In order for the integration into Open IRIS a resource provider can be defined for AppDB which is affiliated with EGI. This resource provider can then define a service to automatically pull information from AppDB via defined APIs.

It is planned for software and virtual machine resources registered in AppDB to have their title, image, category, text description, item ID, and permalink extracted and loaded into Open IRIS as resources. These resources will be tagged as “public” and primarily be a link to a URL to forward users to the AppDB portal. Users of Open IRIS will be able to mark these items as favourites so they can define a virtual pool of resources relevant for their research.

### GOCDB technical integration

GOCDB is a registry of resource providers for computational and storage resources. In order for the integration into Open IRIS a resource provider can be defined for GODCB which is affiliated with EGI. This resource provider can then define a service to automatically pull information from GOCDB via defined APIs.

It is planned for resource providers registered in GOCB to have their name, image, country, text description, contact information, item ID, and permalink extracted and loaded into Open IRIS as resources. These resources will be tagged as “public” and primarily be a link to a URL to forward users to the GOCDB portal. Users of Open IRIS will be able to mark these items as favourites so they can define a virtual pool of resources relevant for their research.

### e-GRANT technical integration

e-GRANT is a tool for submitted requests for research resources and establish SLAs from EGI affiliated organizations. Within Open IRIS this will be primarily offered as a service from the EGI resource provider. As e-GRANT itself can be used to submit resource requests to computational providers it will be offered as a service within Open IRIS to other resource providers interested in using the service. Then a link to e-GRANT can be displayed in the information on their resource provider where interested users can submit requests or establish an SLA with a resource provider.

# Development Roadmap

The service registry and marketplace is a new activity in EGI, therefore there is not an established tool in place. During 2015 requirements were gathered from the EGI Engage proposal input, from EGI.eu staff, from informal discussions, and those expressing interest in using such a tool (either as a consumer or provider).

Table 5. Roadmap to release the first version of the EGI marketplace demonstrator

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task Number** | ***Task Name*** | ***Start Date (MM/YY)*** | ***Completion Date (MM/YY)*** | ***Status*** | ***Dependencies***  ***From other***  ***tasks*** |
| **1** | Design and mockup of public facing marketplace. | 03/2016 | 04/2016 | Started |  |
| **2** | Development of public facing marketplace. | 04/2016 | 05/2016 | Planned | 1 |
| **3** | Define technical integration of AppDB, GOCDB, and eGrant. | 03/2016 | 06/2016 | Started |  |
| **4** | Integration work for of AppDB, GOCDB, and eGrant | 04/2016 | 08/2016 | Planned | 3 |
| **5** | Populate service catalog with EGI services. | 05/2016 | 06/2016 | Planned |  |
| **6** | Test release of first version of public facing marketplace for EGI. | 07/2016 | 07/2016 | Planned | 2, 4, 5 |
| **7** | First release of the EGI Service Registry and Prototype | 08/2016 | 08/2016 | Planned | 2, 4, 5 |

## Design and mockup of public facing marketplace (task 1)

Open IRIS currently requires a login in order to know what types of resources the person has access to seeing. However, resources can be registered that are also classified as “public”, so these will be allowed to be visible by all users without requiring a login. The design of the public facing portal is currently being done with feedback from the Open Access Research Infrastructure (OARI) from the university of Leiden (http://www.oari.science.leidenuniv.nl/). As well as with input from “equipment.data” web site (http://equipment.data.ac.uk) ran by JSIC (http://www.jisc.ac.uk) in the UK. Also, a lot of knowledge can be gained from the AppDB project (http://appdb.egi.eu) from EGI.

## Development of public facing marketplace (task 2)

The mockups are being designed for one central portal and customizable individual portals as desired. The public portals will use adaptive design principles for running on standard web browsers as well as mobile devices.

## Define technical integration of AppDB, GOCDB, and eGrant (task 3)

Currently Open IRIS is designed that resource providers can register resources directly into the system and control access. In this project it will also be extended to retrieve data and integrate with remote data sources. The first candidates for these are the existing EGI services AppDB, GOCDB, and eGrant. However, it will be designed to be extended to other services (e.g. http://equipment.data.ac.uk).

## Integration work for of AppDB, GOCDB, and e-GRANT (task 4)

Based on the work done defining the details of the technical integration of AppDB, GOCDB, and e-GRANT the technical integration work will proceed. There will be collaborations formed with the responsible person from each of these applications to perform the implementation. As AppDB is already extracting similar information from GOCDB it is assumed that Open IRIS can use a similar mechanism for this extraction. AppDB itself already has an extensive REST API for extracting information, so this can be leveraged to import information into Open IRIS. In terms of e-GRANT the existing functionality of Open IRIS will be used, and attempts will be made to make e-GRANT a more prominent part of Open IRIS so resource providers can select to use it for their services to provide SLAs or make offers.

## Populate service catalogue with EGI services (task 5)

EGI already has a well-defined service catalogue (https://www.egi.eu/services/catalogue). So this will be used as the basis for the services imported into Open IRIS. Computational and storage resources (e.g. grid, cloud, cluster) will be defined as applications, whereas services (e.g. consulting, support) will be defined using dynamic forms for request submissions and subsequent tracking.

## Test release of first version of public facing marketplace for EGI (task 6)

As Open IRIS has close match to many of the original requirements (Appendix III), extensive modification will not be required. In addition, the gaps identified were not considered critical to the POC based on the input from external reviewers (Appendix I). Also, as Open IRIS is in productive use by several organizations it already has a level of acceptance that is hoped to extend to a wider set of organizations and users. Therefore, no major obstacles are foreseen for the launch. The majority of the effort will centre around the extension of the public interface and integration into the EGI tools. This test release will allow these changes to be exposed to a wide audience of users to get feedback that can be incorporated before the final release.

## First release of the EGI Service Registry and Prototype (task 7)

As the prototype is to be used productively service agreements will need to be put into place, support process, and ticket workflows will be clarified in preparation for the release. Links to the system will be placed into the EGI web site to direct users to search the service catalogue for EGI services and resources.



# Summary

The goal of this deliverable was to establish the design of the EGI service registry and marketplace. This has been extensively done by examining the defined requirements from D2.3 (Appendix III) and conceptualized in screen mock-ups. These are based on the original use cases gathered before the project started (Appendix II). These can serve as the basis for the POC and have elements prioritized based on feedback on essential features from a survey performed (Appendix I).

In looking at the requirements based on the original EGI use cases and specified EGI Engage deliverable D2.3 the Open IRIS solution comes to the closest to meeting those requirements in its current state. In addition to the fact that it is an established platform with over 2000 users, and used actively by several organizations who have also committed financially to support it as a free platform available to academic and commercial organizations. AppDB is already an established solution in EGI, but does not cover the breadth of requirements of Open IRIS and would require extensive modification that would likely exceed the effort available in EGI Engage for the POC.

1. Essential features of a marketplace based on user surveys and interviews

A survey and several interviews were performed with sixteen large resource providers, projects and research communities (EGI.eu, Barcelona Supercomputing Center, SURFSara, EMBL-EBI, VENUS-C, France Grilles, DARIAH, STFC, MTA SZTAKI, Cyfronet, GRNET, CSC, the Lifescience Grid Community, Neugrid, iMarine, WeNMR). The survey feedback and interviews helped determine if there is interest in a marketplace solution from each perspective, and the requirements for such a solution.

In order to determine the important or essential features of the marketplace, the participants indicated which features of a marketplace would be essential/important and those that were not important. The responses are indicated here:



Rating of Importance of Features in a Research Marketplace

Based on a threshold of 50% we have assigned in order of priority:

1. Service description
2. Direct link to service
3. Filter by
4. Status of the service
5. Visibility/access rules by user, group, organization, department, community, project
6. Categorization of affiliation
7. User rating
8. Usage scenarios

In the course of the EGI-Engage project, different marketplace usage scenarios were developed based on the experience of EGI working with research resource users and providers. Different scenarios were developed from the perspective of the end user, resource provider, and platform administrator. These usage scenarios were then used to elucidate needed features and formulate detailed requirements for the system.

The scenarios of use that individual resource providers use to fund their activities are extremely diverse (pay for use, free at point of delivery, academic, commercial, co-financed, etc.). The solution developed will need to have the ability to allow for all the different usage models, and the resource provider will need to decide which tools to use based on how they finance the provisioning of their services. In addition, many times resource providers use a mixture of scenarios to fund their activities.

The following marketplace user stories have been assembled by EGI based on the requirements from the e-Infrastructure space. These have been analysed and the detailed requirements summarised in Appendix III.

Scenario 1

* The service provider publishes a service
  + The service provider (SP) registers a new service in the service catalogue
  + The SP can assign a service level agreement (SLA) to the new service
  + The SP assigns a price to the new service (can be zero for free services or different prices for different user segments like SMEs, internal users, external academic users, etc.)
  + The SP can define a policy to access the service
  + The new service is available in the service catalogue
* The customer discovers the existing services
  + The customer accesses the service catalogue and gets the list of offered services
  + The customer can look for a service with specific characteristics/requirements (search engine) and compare
  + The customer can read the details of a service: description, how to accept SLA and penalties for underperformance of SLA, price, SPs list, creation date, last update, etc.

Scenario 2

* The customer directly selects and buys a service
  + The customer accesses the service catalogue and looks for a service
  + The customer chooses the service to buy
  + The customer may negotiate the SLA through a broker or accept a pre-defined SLA associated to the service
  + The customer buys the service
* The customer selects a service through a broker
  + The customer lists the requirements that should be satisfied by the service they are looking for
  + The broker identifies the best service according to the customer’s requirements
  + The broker offers the selected service to the customer
  + The customer evaluates the offered service, may negotiate the SLA through a broker or accept the pre-defined SLA
  + The customer buys the service

Scenario 3

* The customer reviews and rates a service and/or a SP
  + The customer selects a service from the list of bought services
  + The customer reviews and rates the service and/or the SP
* The customer wants to check the status of his orders
* The customer consults the consumption/usage
* The customer manages the service
* A SP manages the published services
  + A SP registers into the service catalogue
  + Hides previously published services / changes the conditions associated / highlights the services / announces a maintenance break

Scenario 4

* A SP checks the information associated to services
  + The SP controls the accounting information related to their services (usage, number of users, average consumption) and they can control either by service published or total

1. Requirements from sections 2 and 3 mapped to tools evaluated

A “y” for application indicates a fit or that minor effort would be required to extend feature to match requirements based on input from the developers of the solution.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Application Match** | | | **Section** | **Topic** | **Requirement** |
| **GOCDB** | **AppDB** | **Open IRIS** | **2** | **Service Management** | |
|  |  |  | **2.1** | **Service Provider** | Provisioning of services and service management |
| y | y | y | 2.1.1.1 | Service Registration | Provider registration: Users can register a resource provider. |
| y | y | y | 2.1.1.2 | Service Registration | Service registration: The service provider is able to register a service in the service catalogue and can specify detailed information and display options for his service including: |
| y | y | y | 2.1.1.3 | Service Registration | Service name |
| y | y | y | 2.1.1.4 | Service Registration | Service description |
| y | y | y | 2.1.1.5 | Service Registration | Service instructions |
| y | y | y | 2.1.1.6 | Service Registration | Service visibility |
| y |  | y | 2.1.1.7 | Service Registration | Assign pricing |
| y | y | y | 2.1.1.8 | Service Registration | Usage policy |
| y | y | y | 2.1.1.9 | Service Registration | Picture |
|  |  |  | 2.1.1.0 | Service Registration | Highlight service |
| y | y | y | 2.1.1.1 | Service Registration | Hide a service |
|  |  | y | 2.1.1.12 | Service Registration | Check order status |
|  |  | y | 2.1.2.1 | Usage policies and SLAs | The service provider is able to manage service level agreements and usage policies for his services: |
|  |  | y | 2.1.2.2 | Usage policies and SLAs | Usage policies and SLAs: The service provider is able to manage service level agreements and usage policies for his services: |
|  |  | y | 2.1.2.3 | Usage policies and SLAs | Create service level agreements |
|  |  | y | 2.1.2.4 | Usage policies and SLAs | Modify service level agreements |
|  |  |  | 2.1.2.5 | Usage policies and SLAs | Assign service level agreements to services |
|  |  | y | 2.1.2.6 | Usage policies and SLAs | Modify terms of usage and policies |
| y |  | y | 2.1.3.1 | Availability | Availability: The service provider can define a time window for when his services are unavailable: |
| y |  | y | 2.1.3.2 | Availability | Define unavailability (Time and Day, e.g. 2015-07-03  8:00 AM to 5:00 PM) |
|  |  | y | 2.1.3.3 | Availability | Inform users |
|  |  | y | 2.1.4.1 | Reporting | Reporting: The service provider is able to view usage reports for users, groups and communities that are using his service. The usage report shows data about: |
|  |  | y | 2.1.4.2 | Reporting | Number of users |
|  |  | y | 2.1.4.3 | Reporting | Services used |
|  |  | y | 2.1.4.4 | Reporting | Service consumption |
|  |  | y | 2.1.5.1 | Tickets | Tickets: The service provider is able to manage his tickets: |
|  |  | y | 2.1.5.2 | Tickets | reply to tickets |
|  |  | y | 2.1.5.3 | Tickets | close tickets |
|  |  | y | 2.1.5.4 | Tickets | delete tickets |
|  |  |  | **2.2** | **User** | User of provisioned services |
|  |  | y | 2.2.1 | Tickets | Tickets: The user is able to open support tickets. |
|  | y |  | 2.2.2 | Publications | Publications: Claim publications that were possible as a result of services (e.g. backend integrated with OpenAIRE, data fed into OpenAIRE) |
| y | y | y | 2.2.3 | List | List: The user can list all services they have access to see details (pricing, how to access, SLAs, associated information, creation date, last update, etc.): |
| y |  | y | 2.2.3.1 | List | Services |
| y | y | y | 2.2.3.2 | List | Applications |
| y | y | y | 2.2.3.3 | List | Appliances |
| y | y | y | 2.2.4 | Search | Search: The user is able to search and filter for resources on the basis of: |
| y | y | y | 2.2.4.1 | Search | characteristics |
| y | y | y | 2.2.4.2 | Search | search term |
| y | y | y | 2.2.4.3 | Search | virtual organization |
|  | y | y | 2.2.5 | Status | Status: The user is able to view the status of services: |
|  | y | y | 2.2.5.1 | Status | Availability (available, unavailable, service outage, maintenance, etc.) |
|  |  |  | 2.2.6 | Negotiation | Policies: Negotiate the SLA through a broker or accept the pre-defined SLA associated to the service |
|  |  | y | 2.2.7 | Policies | Policies: The user can accept or decline usage policies. |
|  |  | y | 2.2.8 | Contact | Contact: The user is able to contact the service provider via messages. |
|  | y |  | 2.2.9 | Rating | Rating: The user is able to review and rate services and service providers: |
|  | y |  | 2.2.9.1 | Rating | Rating and commenting system |
|  | y | y | 2.2.10 | Usage | Usage: The user is able to view his own usage information including: |
|  |  | y | 2.2.10.1 | Usage | Services used |
|  | y |  | 2.2.11 | Documentation | Documentation: access documentation (knowledge base) |
|  |  | y | 2.2.12 | Finance | Finance: The user is able to review incurred expenses and pay for a service. |
|  |  | y | 2.2.13 | Resource access | Resource access: request authorization to access one or more services |
|  |  |  | **2.3** | **Directory Manager** | Management of organizations, groups, departments, projects, and communities |
|  |  | y | 2.3.1 | Reporting | Reporting: The directory manager is able to view reports for: |
|  | y | y | 2.3.1.01 | Resources available | Resources available to his virtual organization / users |
|  |  |  | 2.3.1.02 | Resources used | Resources used by his virtual organization / users |
|  |  |  | 2.3.1.03 | Resources booked | Resources booked by his virtual organization / users |
|  |  | y | 2.3.1.04 | Number of users | Number of users |
|  |  |  | 2.3.1.05 | Services used | Services used |
|  |  |  | 2.3.1.06 | Consumption / usage | Consult consumption / usage |
|  |  |  | **2.4** | **Administrator** | Manage resource providers |
|  |  | y | 2.4.1 | manage provider requests | Provider management: The administrators is able to manage resource provider requests: |
|  |  | y | 2.4.1.1 | accept resource requests | Accept resource provider request |
|  |  | y | 2.4.1.2 | decline resource requests | Decline resource provider request (with reason) |
|  |  | y | 2.4.1.3 | disable provider | Disable a provider (with reason) |
|  |  |  | **3** | **Access Management** | |
|  |  | y | **3.1** | Service Provider | Manage of service providers |
| y |  | y | 3.1.1 | Service access request | Service access request: The service provider can manage service access requests for an individual user, group or community: |
| y |  | y | 3.1.1.1 | Service access request | Allow access |
| y |  | y | 3.1.1.2 | Service access request | Decline access (with reason) |
| y |  | y | 3.1.2 | Directory Management | Directory management: The service provider can manage all users, groups and communities that have access to his service: |
| y |  | y | 3.1.2.1 | Directory Management | List of all members |
| y |  | y | 3.1.2.2 | Directory Management | Remove members |
| y |  | y | 3.1.2.3 | Directory Management | Invite members |
|  |  | y | 3.1.3 | Usage limits | Usage limits: The service provider can manage usage quotas for his resources and assign them to users, groups and communities: |
|  |  | y | 3.1.3.1 | Usage limits | Define quota (e.g. time) |
|  |  | y | 3.1.3.2 | Usage limits | Assign quota |
|  |  | y | 3.1.4 | Service access | Service access: The service provider is able to publish his services in the service catalogue and can manage to whom his services are visible as well as hide services: |
| y |  | y | 3.1.4.1 | Service access | Users |
| y |  | y | 3.1.4.2 | Service access | Groups |
|  |  | y | 3.1.4.3 | Service access | Virtual organizations |
| y |  | y | 3.1.4.4 | Service access | Everyone |
|  |  |  | **3.2** | **Directory Manager** | Manage access to organizations, groups, departments, projects, and communities |
| y | y | y | 3.2.1 | Manage directory requests | Directory requests: A directory manager can manage membership requests for users: |
| y | y | y | 3.2.1.1 | Accept directory requests | Accept membership request |
| y | y | y | 3.2.1.2 | Decline directory requests | Decline membership request (with reason) |
| y | y | y | 3.2.2 | Manage memberships | Directory management: The directory manager is able to manage the memberships: |
| y | y | y | 3.2.2.1 | List members | List of all members |
| y | y | y | 3.2.2.2 | Remove members | Remove members |
| y |  | y | 3.2.2.3 | Invite members | Invite members |
| y | y | y | 3.2.2.4 | Promote to administrator | Promote member to administrator |
|  |  |  | **3.3** | **User** | User of provisioned services |
| y | y | y | 3.3.1 | Create a provider | Provider registration: Request Users can request to become a resource provider. |
|  |  | y | 3.3.2 | Directory search | Directory search: search for suitable VO |
|  |  | y | 3.3.3 | Directory requests | Directory requests: The user is able to request access to: |
|  | y | y | 3.3.3.1 | Request access to services | Services |
|  |  | y | 3.3.3.2 | Request access to VO | Virtual Organizations |
|  | y | y | 3.3.3.3 | Request access to groups | Groups |
|  |  |  | **3.4** | **Administrator** | Directory manager |
| y | y | y | 3.4.1 | Manage directory | Directory administration: Able to manage all aspects of the directory to support directory managers and users. |

1. Consultation of experts to assess the technologies to implement the EGI marketplace

**Expert 1**

|  |  |
| --- | --- |
| **Adequacy of the solution against requirements** | |
| AppDB | 3 |
| GOCDB | 2 |
| FIWARE Marketplace Generic Enabler | 2 |
| Open IRIS | 5 |
| PrestaShop | 4 |
| WooCommerce | 4 |
| **Motivation / comments on rating:**  The solutions proposed for evaluations are very different from each other both in terms of target and content. This is very interesting and rewarding for the marketplace. The concept of marketplace is not only used in its original sense of commercial sales but rather as a forum for exchange of knowledge.  I have noted quite severely GOCDB and FIWARE because they must make a major effort to meet the requirements. Indeed, the service description of these services is really not clear to non-specialists. A major effort must be made to make available all the tools.  The OpenIRIS solution is by far the most complete on compliance requirements. This solution provides a set of original and interesting elements for the management of equipments.  The evaluation of Prestashop and WooCommerce solutions raise the question of the scope of the EGI marketplace. Indeed, these solutions are in my opinion not in the main scope of the EGI marketplace. I thought that EGI marketplace scope was: "Establishment of a marketplace where Researchers can discover and exchange services falling to Their research, Ideally Applying the one-stop-shop concept for data and services." (taken from https://wiki.egi.eu/wiki/EGI\_Marketplace). | |

|  |  |
| --- | --- |
| **Possible costs in terms of licenses and support, 1 (high cost for EGI) - 5 (low cost for EGI)** | |
| AppDB | 4 |
| GOCDB | 4 |
| FIWARE Marketplace Generic Enabler | 4 |
| Open IRIS | 5 |
| PrestaShop | 4 |
| WooCommerce | 4 |
| **Motivation / comments on rating:**  The cost evaluation of the different solutions seems difficult to assess with the available information. It seems to me that all solutions offer free use and access. Moreover, it is not clear whether hosting solutions is carried out by EGI or external hosts. Indeed, if the hosting is external to EGI, the cost is almost virtually zero as EGI will only collect information about solutions that follow defined requirements. It seems to me more interesting if EGI could also offer hosting for data confidentiality. This is fundamental for projects such as OpenIRIS, which makes available state-of-the-art scientific and technological instruments.  Furthermore, the accessibility of the service is a crucial point for proper operation. Management of the number of connection, frequency and the need for a connection without interruption varies by project. The AppDB community will surely be diverse and large rather GOCDB is addressed more to a specialist community. For its part, OpenIRIS requires frequent connection number and a service available 7/7 with a well-defined community.  I noted OpenIRIS with a score of 5 because the development of this solution has already obtained funding and various European research institutes are getting organized to sustain this solution. | |

|  |  |
| --- | --- |
| **Is the solution supportable in terms of expertise within EGI, 1 (poor) - 5 (good)** | |
| AppDB | 4 |
| GOCDB | 4 |
| FIWARE Marketplace Generic Enabler | 4 |
| Open IRIS | 3 |
| PrestaShop | 2 |
| WooCommerce | 2 |
| **Motivation / comments on rating:**  The information available does not allow me to answer this question accurately. Indeed, I do not know the expertise of EGI.  However, it seems to me that one can categorize solutions in three categories (in view of the proposed solutions):   * Solutions related to software development (eg AppDB, etc). This is actually a kind of marketplace similar to AppStore (Apple). Making available software with different levels of validation. The community can be broad or specialized according to the type of available softwares. I suppose EGI has a strong expertise in the field. * Solutions for commercial development (eg Prestashop, WooCommerce.). Customized software that can have a strong economic impact. * Original solution allowing the development of an innovative service in a community (eg OpenIRIS.). Software that provides a solution to a need that was not addressed before. Strong societal impact. | |

|  |
| --- |
| **Additional general comments, feedback, or suggestions** |
| Given my experience with this assessment. I find it difficult to accurately assess all solutions. Indeed, business expertise is clearly indispensable to answer the following questions: What is the target of the solution? How this solution improves your productivity? etc.  Moreover, it becomes clear that a marketplace is not only a list of materials/softwares that are available. This can be much more by offering tools to users to enable them to promote their products in a marketplace. The most interesting examples of this point of view are OpenIRIS (facility management), PrestaShop and WooCommerce (web shop).  The establishment of a marketplace by EGI is therefore important at European level to promote exchanges and knowledge. EGI should help hosting and promote software available in the marketplace. The service description must clearly appear and must be understandable by all although its use will be dedicated to a specialist public. Furthermore, all solutions will be highlighted on a dedicated portal (the EGI marketplace of the marketplace) that will enable search by categories and keywords. |

**Expert 2**

|  |  |
| --- | --- |
| **Adequacy of the solution against requirements** | |
| AppDB | 4 |
| GOCDB | 3 |
| FIWARE Marketplace Generic Enabler | 3 |
| Open IRIS | 5 |
| PrestaShop | 3 |
| WooCommerce | 2 |
| **Motivation / comments on rating:**  As far as I can estimate, OpenIRIS looks like a very good fit regarding the requirements (most comprehensive package).  With some programmatoric extensions AppDB could eventually be extended to achieve the same amount of features.  All other tools would most probably have to be extended to much or tweaked to much to fit the extensive list of requirements.  I was quite impressed by the features of PrestaShop even it is obviously for non-scientific purposes. I do think that the visually very attractive design of this web shop software would also be helpful for the selected candidate for the marketplace. In the end, it is also about attracting users. | |

|  |  |
| --- | --- |
| **Possible costs in terms of licenses and support, 1 (high cost for EGI) - 5 (low cost for EGI)** | |
| AppDB | 5 |
| GOCDB | 5 |
| FIWARE Marketplace Generic Enabler | 3 |
| Open IRIS | 4 |
| PrestaShop | 4 |
| WooCommerce | 2 |
| **Motivation / comments on rating:**  My rating is based on the fact that EGI-based software should be the cheapest with respect to cost of licenses.  OpenIRIS' increasing user base and its collaboration model should allow for a rather small running cost regarding licensing.  The PrestaShop license model as well as large community seems to be a strong model.  For FIWARE Marketplace it is not clear how big the user base is. | |

|  |  |
| --- | --- |
| **Is the solution supportable in terms of expertise within EGI, 1 (poor) - 5 (good)** | |
| AppDB | 5 |
| GOCDB | 5 |
| FIWARE Marketplace Generic Enabler | 3 |
| Open IRIS | 4 |
| PrestaShop | 3 |
| WooCommerce | 2 |
| **Motivation / comments on rating:**  I reckon the EGI knows best about already implemented solutions. OpenIRIS should be relatively easy to get along with since strong interactions are already in place. | |

|  |
| --- |
| **Additional general comments, feedback, or suggestions** |
| If more feedback is necessary, I'm happy to dive into an TOP3 choice if reauested. Access to all platforms would be helpful at that moment. |

**Expert 3**

|  |  |
| --- | --- |
| **Adequacy of the solution against requirements** | |
| AppDB | 3 |
| GOCDB | 2 |
| FIWARE Marketplace Generic Enabler | 4 |
| Open IRIS | 4 |
| PrestaShop | 1 |
| WooCommerce | 1 |
| **Motivation / comments on rating:**  APPDB: seems to have the framework and could be further developed to become a marketplace but needs quite some development effort to do so.  Fiware Marketplace: I found it to be a bit confusing and rather infused into Fiware, which could be both pro and con/  GOCDB: is a service registry and as such could be part of the ecosystem but I do not see who it can developed in a marketplace.  Open IRIS: seems to be a service publisher and scheduler and a such it could be the most appropriate tool for this task. Not sure however what is the cost; how easy it is to customise it to cater for our needs/policies/business models.  PrestaShop and WooCommerce: I don't see any value in them, they are online shops which is rather different than a market place. | |

|  |  |
| --- | --- |
| **Possible costs in terms of licenses and support, 1 (high cost for EGI) - 5 (low cost for EGI)** | |
| AppDB | 3 |
| GOCDB | 2 |
| FIWARE Marketplace Generic Enabler | 3 |
| Open IRIS | 3 |
| PrestaShop | 1 |
| WooCommerce | 1 |
| **Motivation / comments on rating:**  I'm not aware of the pricing model for each So i considered all to be free for use and evaluated the development effort that might be required. | |

|  |  |
| --- | --- |
| **Is the solution supportable in terms of expertise within EGI, 1 (poor) - 5 (good)** | |
| AppDB | 3 |
| GOCDB | 2 |
| FIWARE Marketplace Generic Enabler | 5 |
| Open IRIS | 3 |
| PrestaShop | 1 |
| WooCommerce | 1 |
| **Motivation / comments on rating:**  N.A. | |

|  |
| --- |
| **Additional general comments, feedback, or suggestions** |
| N.A. |

**Expert 4**

|  |  |
| --- | --- |
| **Adequacy of the solution against requirements** | |
| AppDB | 4 |
| GOCDB | 2 |
| FIWARE Marketplace Generic Enabler | 3 |
| Open IRIS | 3 |
| PrestaShop | 1 |
| WooCommerce | 1 |
| **Motivation / comments on rating:**  Although the AppDB was not designed as a marketplace, its existing features are very important for the future marketplace. It supports storing, sharing and publishing applications and virtual appliances. The latter ones are coming together with information on which clouds these appliances can be deployed and executed. This very important feature is completely missing from the other systems. More than that, this feature of AppDB could be easily extended with Occopus (developed at TRL6 level by MTA SZTAKI) features enabling extremely easy and automatic deployment of those appliances in the clouds of the EGI FedCloud. Furthermore, these appliances could easily be organized into collaborative services that are also deployed in the EGI FedCloud by Occopus. This integration of AppDB and Occopus is already on its way and close to be finished. The required marketplace functionalities could easily be developed for AppDB provided that the source code is available and the developers of AppDB provide at least consultancy support for the development.  FIWARE is similar to AppDB in many senses. It also supports applications and VMs but the main problem is that it is a product of a project that is not related to EGI and it is not clear what will happen to FIWARE after the end of the project. In addition, it is questionable how much the developers would be motivated to do the required extensions and modifications for the sake of the EGI community. That’s why I gave less score for it than for AppDB.  Open IRIS has a very nice, well developed user interface for marketplace purposes. However, it misses the support for virtual appliances and their execution in the EGI FedCloud. This will be an extremely important feature in the future as the user communities more and more move and migrate into clouds.  GOCDB’s features are very limited compared to the generic requirements of the marketplace. Furthermore, it provides many features that are not needed at all for the marketplace and therefore they just cause useless complications for the potential users.  PrestaShop and WooCommerce strongly target commercial applications and commercial web design. Although they are well designed and user-friendly environments they miss many features required for the marketplace. | |

|  |  |
| --- | --- |
| **Possible costs in terms of licenses and support, 1 (high cost for EGI) - 5 (low cost for EGI)** | |
| AppDB | 5 |
| GOCDB | 4 |
| FIWARE Marketplace Generic Enabler | 3 |
| Open IRIS | 3 |
| PrestaShop | 1 |
| WooCommerce | 1 |
| **Motivation / comments on rating:**  AppDB is in principle open source (although I have not seen its code in any public repository yet) and developed by EGI community members. Therefore, its further development could easily be organized inside the EGI community (for example, SZTAKI is already voluntarily works with the developers of APPDB to enhance the AppDB features with Occopus services). The same is true for GOCDB although the work needed for extending it towards the required marketplace would require more efforts than for AppDB. FIWARE and OpenIRIS are also open source software based services but they are developed by communities which are not directly involved in the work of the EGI community and hence motivating them for doing the required developments could be more costly than for the EGI members. Alternatively, EGI members can do the required development based on the OSS but then the learning time will be long and costly. Finally, PrestaShop and WooCommerce are commercial code based systems. The EGI communities have no access to those codes and the companies are certainly not motivated to do code development for EGI communities or it will be very expensive. | |

|  |  |
| --- | --- |
| **Is the solution supportable in terms of expertise within EGI, 1 (poor) - 5 (good)** | |
| AppDB | 5 |
| GOCDB | 5 |
| FIWARE Marketplace Generic Enabler | 3 |
| Open IRIS | 3 |
| PrestaShop | 1 |
| WooCommerce | 1 |
| **Motivation / comments on rating:**  As explained for the previous question, AppDB and COGDB are in the hand of EGI community members and hence EGI expertise is available to further develop them.  FIWARE and OpenIRIS are OSS codes and hence after a certain learning phase EGI community members would be able to further develop them according to the EGI needs.  PrestaShop and WooCommerce are closed codes and hence EGI communities will never have the required expertise to further develop and support them. | |

|  |
| --- |
| **Additional general comments, feedback, or suggestions** |
| My final recommendation: choose AppDB under the following conditions:   1. The developers of the AppDB code place the source code of AppDB into a public open repository (Sourceforge, Github, etc.); 2. At least two independent EGI member organizations will work on the extension of the existing code towards the marketplace. This guarantees that even if one of the developer organization leaves the project the work can go on. At this stage again a new, second organization should be involved in the code development. It would be better to start the code development with three organizations; 3. The current code owners of AppDB should provide initial training for the other organizations to learn the source code. In this way, after the training, the new organizations will be able to further develop the AppDB code. |

**Expert 5**

|  |  |
| --- | --- |
| **Adequacy of the solution against requirements** | |
| AppDB | 3 |
| GOCDB | 2 |
| FIWARE Marketplace Generic Enabler | 4 |
| Open IRIS | 4 |
| PrestaShop | 3 |
| WooCommerce | 3 |
| **Motivation / comments on rating:**  None of the selected software have a ready-to-use solution compliant with the requirements identified for the EGI Marketplace. Some of these functionalities but some additional effort is still needed to include the missing ones and have a production EGI Marketplace complaint with the EGI ecosystem.  In more details:  GOCDB:  • It is an EGI Service provided by STFC;  • It is accessed by all the project actors (end-users, site-managers, NGI managers, support teams and VO managers);   * It is possible to assign different roles;   • No support for Identity Federations authentication;   * Authentication based on X.509 digital certificates;   • It is focused on HTC and Cloud resources;  • It doesn’t include functionalities to compare different resources;  • Provides functionalities to implement the basic Service Registration;   * Can search and list for a HTC and Cloud-based resources;   • Does not provide functionalities to manage usage policies and SLA:   * Can’t create, modify, negotiate and assign SLA for his service; * Can’t manage terms of usage and policies;   • Provides filtering functionalities;   * The user is able to search and filter for resources on the basis of characteristics;   • Rating is not supported;  • No reports are available for users, groups and communities;  • No support for tickets is available;  • Provides functionalities to help SP to specify the availability for his service.  • Provides functionalities to scope resources and make them visible/invisible;  • It is no possible to contact service providers via messages (only via e-mail);  • Accounting, monitoring and billing services are not available;  • The solution is open-source.  AppDB:  • Accounting, Monitoring and Billing services are not available.  • It is already integrated with the EGI SSO.  • Provides functionalities to implement the basic Service Registration;   * Listing services/applications/VAs/…;   • Does not provide functionalities to manage usage policies and SLA:   * Can’t create, modify, negotiate and assign SLA for his service;   • Provides functionalities to contact the service provider and customers via messages;  • Provides functionalities to rate services;  • Provides filtering functionalities;   * The user is able to search and filter for resources on the basis of characteristics, VO, search term;   • Offer the possibility to upload publications;  • Very basic reporting functionalities are available;  • It supports a very reliable and rich API;  • It does offer correlated information from many services (e.g. OpsPortal, Perun, GOCDB, BDII and OpenAIRE).  • The solution is open-source.  WMarket:  • It supports an authentication based on username and password or with an OAuth2 token;   * It can be configured to uses external authentication system;   • Does not provide any services to manage usage policies and SLA, specify a time window when the service is available or unavailable;  • No ticketing system is available;  • Customer can browse the available offerings organized in different categories:   * Clicking on a generic offering it is possible to get its details and review; * Customer can book offerings, compare different offerings, browse and review stores;   • The SW is distributed under the BSD license.  Open IRIS:  • The Open IRIS solution does not support Identify Federation Authentication;  • The solution is focused on resources (instruments and applications) sharing and create virtual pools of resources for collaborations and communities.   * Every functionalities can be easily accessed from a central dashboard.   • Different profile of users are supported;  • From the dashboard different topology of users can:   * track and manage any requests and functionalities for resource Listing, Booking, Pricing Service requests and produce usage statistics; * access to the service Usage Policy; * specify the time window when the resource is available, define unavailability * manage all the orders; * book a service; * manage tickets; * access report statistics.   • Provides functionalities to implement the basic Service Registration;  • The solution is open-source.  PrestaShop:  • It provides an e-commerce platform to manage different products.  • Very limited security policies are adopted.   * No support for Identity Federation Authentication schema;   • Mobile friendly;  • PrestaShop is a free Open-Source e-commerce solution  • Support for real-time analytics;  • Customizable with different modules  • Easy to customize theme (More 2,000+ professional e-commerce templates);  • It implements many Merchant functionalities.  WooCommerce:  • Open-source e-Commerce platform and a powerful Content Management System (CMS) easy to adapt and customize (70Millions+ of websites);  • REST API available for developer;  • Free;  • Open Source;  • Thousands of plugins, themes and widgets available;  • Based on WordPress;  • Complaint with W3C standards. | |
|  | |

|  |  |
| --- | --- |
| **Possible costs in terms of licenses and support, 1 (high cost for EGI) - 5 (low cost for EGI)** | |
| AppDB | 5 |
| GOCDB | 5 |
| FIWARE Marketplace Generic Enabler | 3 |
| Open IRIS | 3 |
| PrestaShop | 3 |
| WooCommerce | 3 |
| **Motivation / comments on rating:**  All the selected solutions are open-source. | |

|  |  |
| --- | --- |
| **Is the solution supportable in terms of expertise within EGI, 1 (poor) - 5 (good)** | |
| AppDB | 5 |
| GOCDB | 2 |
| FIWARE Marketplace Generic Enabler | 2 |
| Open IRIS | 2 |
| PrestaShop | 2 |
| WooCommerce | 2 |
| **Motivation / comments on rating:**  AppDB and GOCDB are solutions widely used by EGI members.  More than 52,000 websites have been created using PrestaShop and the trend is still increasing.  Over the 30% of all online stores have been created with WooCommerce. This is the most popular e-Commerce platform on the web (http://trends.builtwith.com/shop). | |

|  |
| --- |
| **Additional general comments, feedback, or suggestions** |
| N.A. |

**Expert 6**

|  |  |
| --- | --- |
| **Adequacy of the solution against requirements** | |
| AppDB | 5 |
| GOCDB | N.A. |
| FIWARE Marketplace Generic Enabler | 5 |
| Open IRIS | 5 |
| PrestaShop | 5 |
| WooCommerce | 5 |
| **Motivation / comments on rating:**  I haven't been able able to access GOCDB (server not found). I even haven't been asked for a certificate. | |

|  |  |
| --- | --- |
| **Possible costs in terms of licenses and support, 1 (high cost for EGI) - 5 (low cost for EGI)** | |
| AppDB | 3 |
| GOCDB | 3 |
| FIWARE Marketplace Generic Enabler | 3 |
| Open IRIS | 3 |
| PrestaShop | 4 |
| WooCommerce | 4 |
| **Motivation / comments on rating:**  All of them are open source and/or offer community version with potentially commercial paid support especially for for PrestaShop and WooCommerce.  AppDB and GOCDB are already paid by EGI so the additional cost should be low.  OpenIRIS is currently supported by the Swiss Federation.  However for these 3, the cost may be considered as high as we have to consider long-term funding and support as they do not (yet) rely on a large and commercially supported model.  The main cost may come also from the high-availability (HA) configuration to support thousands or hundreds of thousands of users. The support of HA is not described anywhere for more EGI-related products (AppDB, GOCDB, Open IRIS) or FIWARE MP | |

|  |  |
| --- | --- |
| **Is the solution supportable in terms of expertise within EGI, 1 (poor) - 5 (good)** | |
| AppDB | 5 |
| GOCDB | 5 |
| FIWARE Marketplace Generic Enabler | 3 |
| Open IRIS | 4 |
| PrestaShop | 5 |
| WooCommerce | 5 |
| **Motivation / comments on rating:**  I'm not sure about the expertise people have within EGI about Prestashop and WooCommerce but they are definitely supported by huge worldwide communities.  AppDB and GOCDB are already supported by EGI people, therefore the expertise is there but the user communities are currently much narrow, as for OpenIRIS.  I'm not sure if people within EGI have any expertise using FIWARE MP | |

|  |
| --- |
| **Additional general comments, feedback, or suggestions** |
| N.A. |

**Expert 7**

|  |  |
| --- | --- |
| **Adequacy of the solution against requirements** | |
| AppDB | 4 |
| GOCDB | 1 |
| FIWARE Marketplace Generic Enabler | 3 |
| Open IRIS | 3 |
| PrestaShop | 3 |
| WooCommerce | 3 |
| **Motivation / comments on rating:**  N.A. | |

|  |  |
| --- | --- |
| **Possible costs in terms of licenses and support, 1 (high cost for EGI) - 5 (low cost for EGI)** | |
| AppDB | 4 |
| GOCDB | 3 |
| FIWARE Marketplace Generic Enabler | 2 |
| Open IRIS | 2 |
| PrestaShop | 2 |
| WooCommerce | 2 |
| **Motivation / comments on rating:**  N.A. | |

|  |  |
| --- | --- |
| **Is the solution supportable in terms of expertise within EGI, 1 (poor) - 5 (good)** | |
| AppDB | 5 |
| GOCDB | 3 |
| FIWARE Marketplace Generic Enabler | 3 |
| Open IRIS | 2 |
| PrestaShop | 2 |
| WooCommerce | 2 |
| **Motivation / comments on rating:**  N.A. | |

|  |
| --- |
| **Additional general comments, feedback, or suggestions** |
| N.A. |

**Expert 8**

|  |  |
| --- | --- |
| **Adequacy of the solution against requirements** | |
| AppDB | 4 |
| GOCDB | 1 |
| FIWARE Marketplace Generic Enabler | 3 |
| Open IRIS | 4 |
| PrestaShop | 4 |
| WooCommerce | 5 |
| **Motivation / comments on rating:**  Features selected for the evaluation (which I fully share) shows we need simple solution, not full ecosystem with complicated functionality. That is why my preference is in WooCommerce.  OpenIRIS seems too much for our goals. AppDB is simple enough, but might missing some of the features in look&fell, which one: responsive design seems critical. GOCDB is completely different purpose and mixing them with marketplace might lead to lost of understanding what are element of infrastructure and what is the offering. | |

|  |  |
| --- | --- |
| **Possible costs in terms of licenses and support, 1 (high cost for EGI) - 5 (low cost for EGI)** | |
| AppDB | 3 |
| GOCDB | 3 |
| FIWARE Marketplace Generic Enabler | 2 |
| Open IRIS | 3 |
| PrestaShop | 2 |
| WooCommerce | 5 |
| **Motivation / comments on rating:**  EGI solutions needs to be developed using mid to large programing effort.  To integrate WooCommerce is not programming, is web site, that every interactive agency can do cheap. Maintenance is also important -- WooCommerce has big community to support solutions, and almost everyone can fix/extent, if not extended by wordpress pluggin.  PrestaShop seems more integrated and not relevant.  It is hard too judge FIWARE Marketplace, but support might be poor. | |

|  |  |
| --- | --- |
| **Is the solution supportable in terms of expertise within EGI, 1 (poor) - 5 (good)** | |
| AppDB | 5 |
| GOCDB | 5 |
| FIWARE Marketplace Generic Enabler | 1 |
| Open IRIS | 4 |
| PrestaShop | 3 |
| WooCommerce | 5 |
| **Motivation / comments on rating:**  For AppDB and GOCDB it is obvious expertise, but the question for me is the effort available to provide such development. In case OpenIRIS I believe situation is similar, but AFAIK developer team is outsourced.  In case WooCommerce I believe almost every partner staff, and for sure egi.eu staff, has some experience with doing websites with wordpress. Even if not, every interactive agency will do it. | |

|  |
| --- |
| **Additional general comments, feedback, or suggestions** |
| Having in mind long quest for searching appropriate tool for the marketplace, we need finally decide on the solution (simpler - better) and just start publishing the content, now. So, time-to-market(place) should also be essential in this evaluation. WooCommerce wins also at this criteria.  Open platform, such as wordpress with WooCommerce, with essential functionality will give us opportunity to focus on clarifying and shaping attractive content.  Functionality related technical details such exacting information about service elements, searching for appropriate image do deploy in specific site, or agree on details on allocation, should be still provided in specialised tools (GOCDB, AppDB, e-GRANT, respectively). |

1. https://catalogue.clouds.geant.net/ [↑](#footnote-ref-1)
2. https://www.theubercloud.com [↑](#footnote-ref-2)
3. https://www.scienceexchange.com/ [↑](#footnote-ref-3)
4. http://www.internet2.edu/vision-initiatives/initiatives/internet2-netplus/ [↑](#footnote-ref-4)
5. http://hnx.helix-nebula.eu/ [↑](#footnote-ref-5)
6. https://www.digitalmarketplace.service.gov.uk/ [↑](#footnote-ref-6)
7. https://azure.microsoft.com/en-us/marketplace/ [↑](#footnote-ref-7)
8. https://www.fortissimo-marketplace.com [↑](#footnote-ref-8)
9. http://strategic-project.eu/ [↑](#footnote-ref-9)