



Grant Agreement No.: 604590
Instrument: Large scale integrating project (IP)
Call Identifier: FP7-2012-ICT-FI



eXperimental Infrastructures for the Future Internet

D4.5: XIFI Marketplace Implementation v2

Revision: v1.2

Work package	WP 4
Task	Task 4.1
Due date	31/12/2014
Submission date	25/02/2015
Deliverable lead	Universidad Politécnica de Madrid (UPM)
Authors	Jorge Valhondo (UPM), Joaquin Iranzo Yuste (ATOS), Ömer Faruk Ozdemir (ATOS), Mirko Ardinghi (ENG), Francesco Rossi (ENG), Bassem I. Nasser (IT-INN), Susana González Zarzosa (ATOS)
Reviewers	Attilio Broglio (CREATE-NET), Sándor Laki (WIGNER), József Stéger (WIGNER)

Abstract	This deliverable covers the second release of the XIFI Marketplace described in T4.1. XIFI Marketplace is a portal to access information on services offered by the federation through a single entry point. It provides the glue to integrate the rest of modules and services provided by WP4. This second version is an upgrade and refinement of the first one, in which all defined tools are fully integrated and operational.
Keywords	Configuration, installation, user manual, specification, Generic Enabler, Specific Enabler, FI-WARE, Management, Federation, PaaS, SaaS

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	16/01/2015	Version ready for internal review	Jorge Valhodno (UPM) et al.
V1.1	05/02/2015	Final revision	Jorge Valhondo (UPM) et al.
V1.2	25/02/2015	Format editing	Jose Gonzalez (UPM)

Disclaimer

This report contains material which is the copyright of certain XIFI Consortium Parties and may only be reproduced or copied with permission in accordance with the XIFI consortium agreement.

All XIFI Consortium Parties have agreed to publication of this report, the content of which is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License¹.

Neither the XIFI Consortium Parties nor the European Union warrant that the information contained in the report is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using the information.

Copyright notice

© 2013 - 2015 XIFI Consortium Parties

Project co-funded by the European Commission in the 7 th Framework Programme (2007-2013)		
Nature of the Deliverable:		R (Report)
Dissemination Level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to bodies determined by the XIFI project	
CO	Confidential to XIFI project and Commission Services	

¹ http://creativecommons.org/licenses/by-nc-nd/3.0/deed.en_US

EXECUTIVE SUMMARY

Deliverable “D4.5 XIFI Marketplace implementation v2” contains the second software version of the XIFI Marketplace portal, which is the single entry point for accessing to services offered by the federation. It provides the glue to integrate the rest of modules and services provided by WP4. This second version is an upgrade and refinement of the first one, in which all defined tools are fully integrated and operational.

This deliverable is composed of two main parts:

- Source codes of related XIFI components that are available in the public version control system [1].
- Description of the XIFI Marketplace implementation, comprising two sections: the first one describes XIFI components involved, showing their necessity and worth value added to the Portal, and the second one outlines the different modules implemented, focusing on their functionalities related to XIFI Marketplace.

Both parts are based on the specifications defined in “D4.1b- Services and tools specification” [2], description of tools in “D4.2- Baseline Tools v1” [3] and in “D4.4- Baseline Tools v2” [4], and description of XIFI Marketplace implementation in “D4.3- XIFI Marketplace implementation v1” [5]. They are also aligned with the rest of the components from WP2 and WP3, showcases in WP6 and business model in WP8.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	3
TABLE OF CONTENTS	4
LIST OF FIGURES	6
ABBREVIATIONS	8
1 INTRODUCTION.....	9
1.1 Scope.....	9
1.2 Document Convention	9
1.3 Intended Audience and Reading Suggestions	9
2 XIFI PORTAL	11
2.1 Portal Overview	11
2.2 Cloud Portal	12
2.3 Federated Identity Management.....	14
2.4 Monitoring Dashboard	17
2.5 SLA and Accounting Dashboard	19
2.6 Security and Privacy Dashboard	27
2.7 Interoperability Tools.....	28
2.8 Infographics and Status Pages.....	30
2.9 Federation Manager GUI	32
3 INTEGRATED MODULES IN XIFI MARKETPLACE	34
3.1 Federated Identity Management.....	34
3.2 Cloud Portal	36
3.3 Deployment and Configuration Adapter - DCA	40
3.4 Federation Manager	42
3.5 Federation Monitoring API.....	43
3.6 FIWARE Catalogue	44
3.7 Marketplace and Resource Catalogue	48
4 GENERIC ENABLERS	53
5 NON CONVENTIONAL SERVICES AND SPECIFIC SERVICES.....	58
6 CONCLUSIONS	62
REFERENCES.....	63
APPENDIX A: POLICY RECOMMENDATION SERVICE.....	64
Summary	64
A.1.1 Component Leader	66



- A.1.2 Motivation 66
- A.1.3 User stories 66
- A.1.4 State of the art 67
- A.1.5 Architecture design 69
- A.1.6 Release plan 71
- A.1.7 Test case 71
- A.1.8 Installation manual 73
- A.1.9 User manual 74
- P3P policy 79
- APPEL evaluation algorithm 83



LIST OF FIGURES

Figure 1: Location of the XIFI Marketplace Portal in the XIFI Reference Architecture.....	11
Figure 2: Main page of the FIWARE Lab portal	12
Figure 3: Header page of the FIWARE Ops portal.....	12
Figure 4: Detail of the header page of FIWARE Ops portal.....	12
Figure 5: Cloud Portal: Switch region	13
Figure 6: Cloud Portal: Switch region in a Blueprint Tier.....	14
Figure 7: List of organization which the user belongs	15
Figure 8: Relationships between the Components/Roles/Organizations/users	15
Figure 9: JSON message with the user/organization and roles.....	16
Figure 10: Change between different organizations in the Security Dashboard component	17
Figure 11: Organization assigns roles to its members for one application	17
Figure 12: Monitoring Dashboard: NAM dashboard.....	19
Figure 13: Agreement Dashboard.....	24
Figure 14: Status of the agreements. Able to click on icon	24
Figure 15: Violation details	25
Figure 16: Info of the agreement.....	25
Figure 17: List of services.....	25
Figure 18: Create an agreement	26
Figure 19: List of templates for providers	27
Figure 20: Add template for providers.....	27
Figure 21: Security Dashboard: main dashboard.....	28
Figure 22: Display available patterns in the Interoperability Tool	29
Figure 23: Result of interoperability testing and monitoring.....	29
Figure 24: Infographics main page	30
Figure 25: Infographics: access to detailed information by clicking on each panel	31
Figure 26: Infographics: example cores per node	31
Figure 27: Status Pages main page	32
Figure 28: Federation Manager GUI - Compliance Survey.....	33
Figure 29: Federation Manager GUI: Federation Admin New Requests.....	33
Figure 30: Federation Manager GUI: Infrastructure Toolbox (ITBox)	33
Figure 31: Application registration.....	34
Figure 32: User's application.....	35
Figure 33: Role management	35
Figure 34: List of my Organization.....	36
Figure 35: List of others Organization.....	36
Figure 36: Switch from a user to an organization	36
Figure 37: List of available images to deploy as PaaS.....	37
Figure 38: Deploy a PaaS (Instance creation).....	37
Figure 39: Deploy a PaaS (Access & Security)	38
Figure 40: Deploy a PaaS (Customization).....	38
Figure 41: Deploy a PaaS (Summary)	39
Figure 42: List of deployed PaaS.....	39
Figure 43: DCA API example.....	41
Figure 44: Connection between WStore and CloudPortal across DCA API	41

Figure 45: DCA API flow chart.....	42
Figure 46: Federation Monitoring API: particular region data request and response	44
Figure 47: FIWARE Catalogue GUI website	45
Figure 48: Recommendation Tool model	46
Figure 49: Policy Recommender: policies view	48
Figure 50: Policy Recommender: preferences layout	48
Figure 51: Marketplace homepage.....	49
Figure 52: PaaS services table	49
Figure 53: A PaaS service.....	50
Figure 54: PaaS service page in Cloud Portal.....	50
Figure 55: PaaS services table after a PaaS published as SaaS.....	51
Figure 56: SaaS services table	51
Figure 57: GE list.....	52
Figure 58: GE list filtered by sub chapter	54
Figure 59: GE Search result	55
Figure 60: GE's filtered by node name	55
Figure 61: Sample GE.....	56
Figure 62: All-node information with GE, NCS and SE	56
Figure 63: GE detail page	57
Figure 64: Resource creation step 1	58
Figure 65: Resource created.....	58
Figure 66: Resource creation, USDL selection part.....	59
Figure 67: Resource publish page.....	59
Figure 68: SE list	60
Figure 69: NCS list	60
Figure 70: NCS search.....	61
Figure 71: Privacy Recommendation service within the Xifi Federation Platform context	65
Figure 72 Privacy Recommendation service: Rule 3 encoded.....	69
Figure 73 Recommendation service.....	69
Figure 74 Recommendation service components	70
Figure 75 Recommendation service technology used.....	70
Figure 76: Privacy Recommendation service: P3P policy example	72
Figure 77: Authentication page.....	75
Figure 78: Policies view.....	75
Figure 79: Preferences layout	76
Figure 80: Add RuleSet	77
Figure 81: Add rules	77
Figure 82: Evaluation results- Policies accepted	78
Figure 83: Evaluation results – Policies rejected	79

ABBREVIATIONS

API	Applications Programming Interface
EPR	Endpoint reference
FI	Future Internet
FI-PPP	Future Internet Public-Private Partnership Programme
FMC	Fundamental Modelling Concepts
GE	Generic Enabler
GUI	Graphical User Interface
IaaS	Infrastructure as a Service
IdM	Identity Management
PaaS	Platform as a Service
QoS	Quality of service
SaaS	Software as a Service
SBF	Service Business Framework
SLA	Service Level Agreement
SLO	Service Level Objective
SSO	Single sign-on
SE	Specific Enabler
SIEM	Security Information Event Management
UDDI	Universal Description, Discovery and Integration
USDL	Unified Service Description Language
WADL	Web Application Description Language
WSDL	Web Services Description Language
WSLA	Web Service Level Agreements
WSOL	Web Service Offerings Language

1 INTRODUCTION

This deliverable contains information about the second software version of the XIFI Marketplace Portal described in the framework of WP4. XIFI Portal is conceived as a single entry point for accessing to services offered by the federation.

XIFI Portal works as a collection of services offered to XIFI stakeholders and is in particular an aggregation point for the following services:

- Resource Catalogue and Recommendation Tool
- Monitoring Dashboard
- SLA and Accounting Dashboard
- Security and Privacy Dashboard
- Interoperability Tools
- Federated Identity Management
- Infographics and Status Pages
- Cloud Portal
- Federation Manager GUI

This deliverable is based on the specifications and definitions of the different tools aggregated by the XIFI Marketplace Portal, which are already described in “D4.1b- Services and tools specification” [2], “D4.4- Baseline Tools v2” [4] and Federated Identity Management component description [6].

1.1 Scope

This deliverable focuses on describing the status of the XIFI Marketplace version 2, relating XIFI Portal to the rest of XIFI components involved and the particular modules implemented for XIFI Marketplace. Moreover, this deliverable describes how is each XIFI tool provided by WP4 linked and integrated with FIWARE Lab portal [7] (provided by FIWARE project [8]) or FIWARE-Ops portal [9], depending on the kind of users that will make use of it.

1.2 Document Convention

The formatting of the document is compliant with the deliverable template provided by the XIFI project. No other specific convention has been applied.

1.3 Intended Audience and Reading Suggestions

The intended audience of this document comprises:

- XIFI partners involved in technical activities so as to have a first overview of the concepts, requirements and architecture of the XIFI Marketplace.
- All the XIFI partners in general in order to have a common understanding of the objective of XIFI Marketplace and to provide feedback and suggestions.
- Third party stakeholders like Future Internet Developers, Service Providers and Organizations, in order to have a clear understanding of the XIFI Marketplace inside the XIFI project.

The document is divided into the following sections:

- Section one: this is the current section. Consists of an introduction to the document, followed

by scope, document convention, intended audience and reading suggestions sections, providing a clear understanding of the deliverable D4.5.

- Section two: describes the XIFI Portal as a single entry point for accessing the services offered by the federation, and also describing its relation to the rest of XIFI components involved and their worth valued added to XIFI Portal.
- Section three: describes modules implemented concerning XIFI Marketplace for the integration of the XIFI components involved, showing also their functionalities and uses.

2 XIFI PORTAL

2.1 Portal Overview

XIFI Marketplace Portal aggregates a wide set of tools accessible and offers them from a single entry point. The set of tools is formed by the different modules provided by WP4 and the portal functionalities have a strong dependency of those modules. A whole description of the modules provided by WP4 can be found in “D4.4 Baseline Tools v2” [4] and an updated and maintained description can be found in the XIFI wiki [10].

Thus, XIFI Portal offers a set of cloud community services, which shall be accessible through a graphical representation (a Graphical User Interface - GUI) supported by this portal, and gaining importance and value for related stakeholders [11] like:

- Developers
- Technology Providers
- Infrastructure Owners & Operators

In Figure 1, the FMC compositional structure diagram highlights the location of the XIFI Marketplace portal in the XIFI Reference Architecture.

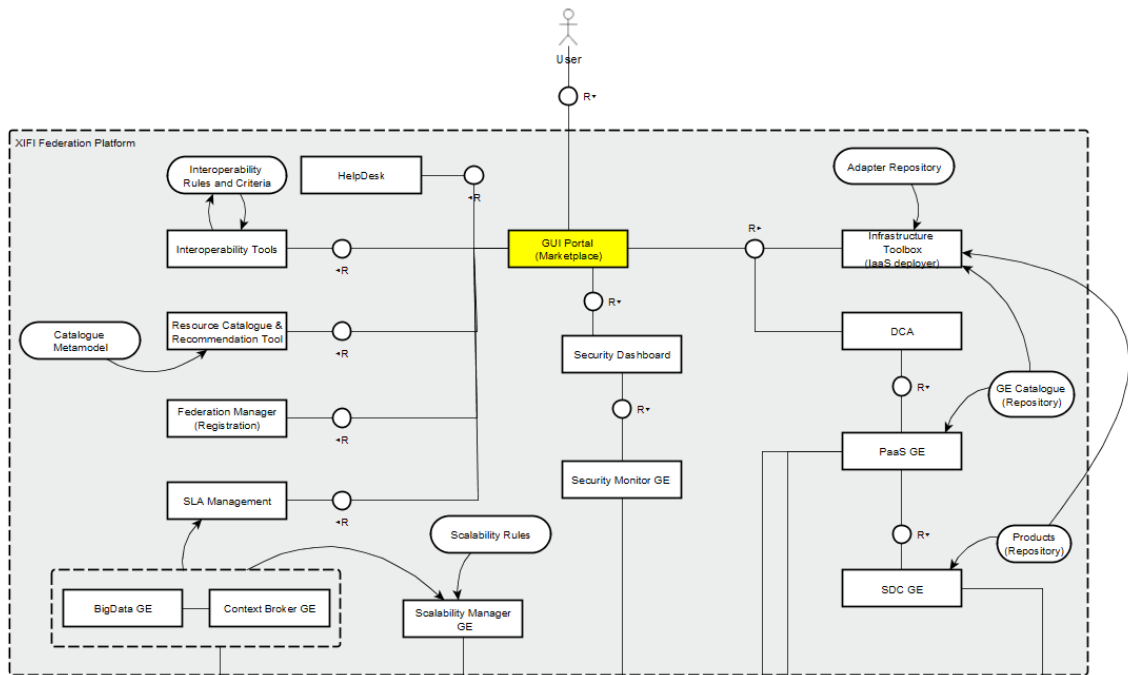


Figure 1: Location of the XIFI Marketplace Portal in the XIFI Reference Architecture

XIFI Marketplace portal aggregated tools are integrated in FIWARE Lab portal [7], that is being provided by FIWARE Project [8], and in FIWARE Ops portal depending on the set of users that will make use of it. Thus, as Infographics and Status Pages and XIFI Marketplace are related to End Users (Developers and Service Providers) these components are integrated in FIWARE Lab portal. The rest of the tools provided are mainly related to Infrastructure Owners and Federator Manager instead, so they will be integrated in FIWARE-Ops portal [9].

Figure 2 provides a view of the FIWARE Lab portal.

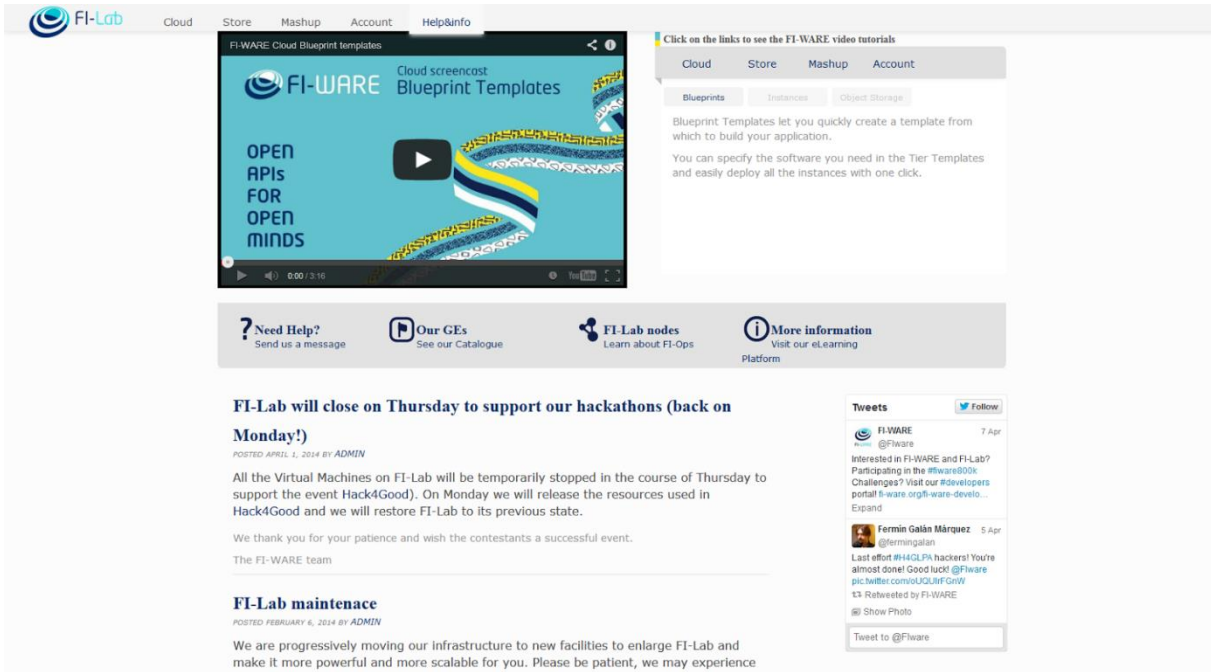


Figure 2: Main page of the FIWARE Lab portal

And concerning FIWARE Ops portal, the following figures show the common header that will be used, which is also aligned to FIWARE Lab portal and also to FIWARE project.



Figure 3: Header page of the FIWARE Ops portal

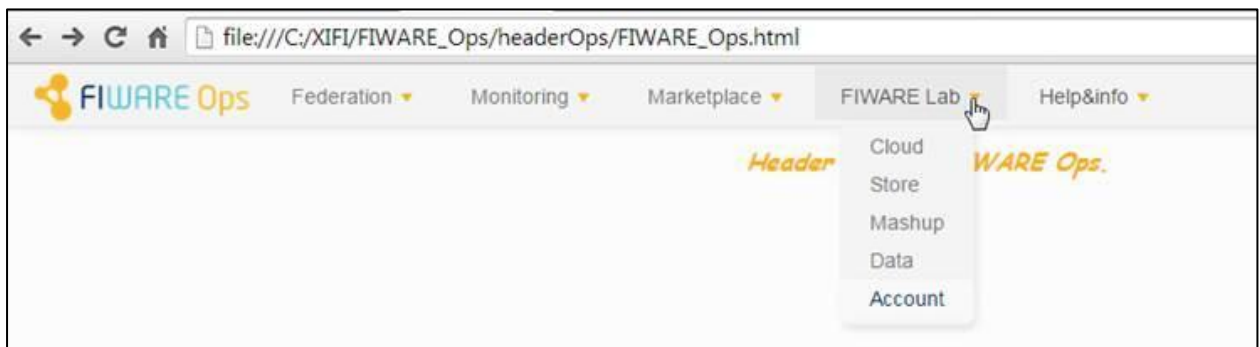


Figure 4: Detail of the header page of FIWARE Ops portal

2.2 Cloud Portal

The main functionality of Cloud Portal is to facilitate the management of IaaS and PaaS resources to the cloud user and to perform operations over the underlying infrastructure as:

- Launching instances on a base of images
- Creating images in the image repository

- Retrieving flavours from the resource
- Blueprints and Tiers management

All these actions can be done in any of the XIFI regions registered in the system in a uniform manner.

Each XIFI node has an installation of the OpenStack services (Nova, Glance, Neutron...) in order to provide cloud resources to the users such as: virtual machines, images, networks, etc. In addition Openstack also provides a Dashboard to help the users managing their resources. However, in XIFI there are resources and advanced features that are not included in OpenStack:

- Platform as a Service management
- Public and private Region support
- Integration with the other XIFI portals
- OAuth2 support to authenticate the users using their XIFI accounts
- High availability mode between regions

Therefore, the goal of the Cloud Portal component is to provide a web interface that allows the users to manage all these Cloud resources and tools in an easy and intuitive way and be authenticated using their federated XIFI accounts. Cloud Portal is accessible from XIFI Portal including all the value-added functionalities already described and is integrated in FIWARE Lab portal header.

Figure 5 and Figure 6 show Cloud Portal graphical user interface.

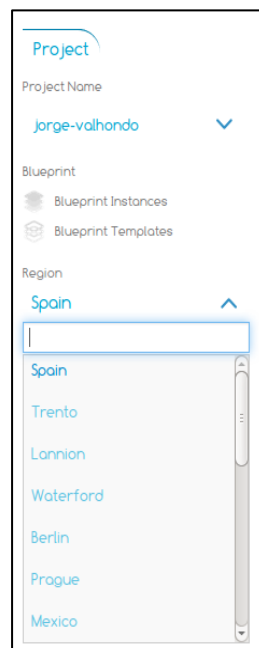


Figure 5: Cloud Portal: Switch region

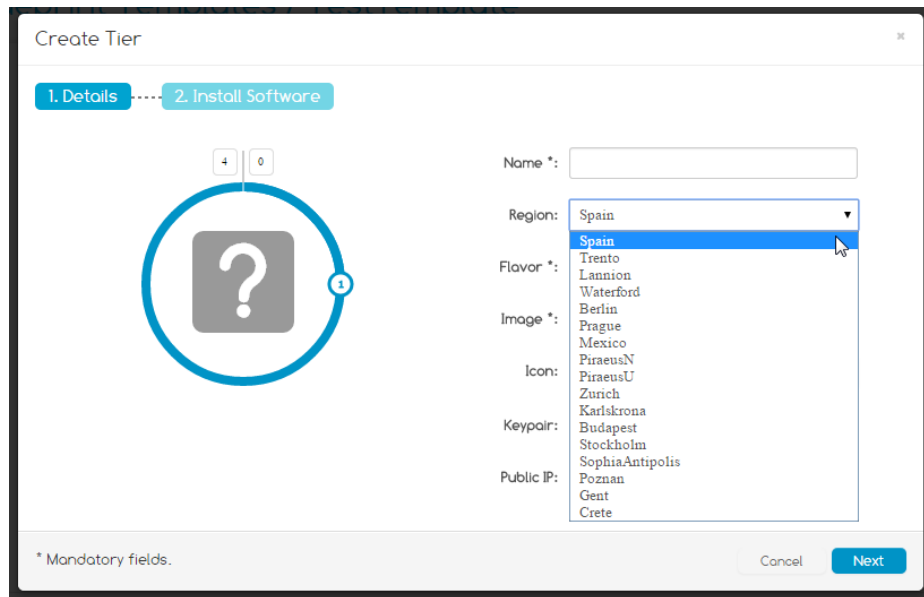


Figure 6: Cloud Portal: Switch region in a Blueprint Tier

2.3 Federated Identity Management

The Federated Identity Management tool (developed in WP2) is one of the key components for the XIFI Marketplace Portal / FIWARE Lab. Federated Identity Management main functionality is providing a federated management of identities for members of XIFI federation. This component can be distributed on all the nodes of the federation and stores identities of authorized users. The Federated Identity Management component extends the FI-WARE IdM GE [19] to support XIFI requirements.

All the components have followed the same approach to deal with roles, users and organizations, in order to be completely integrated in the Marketplace Portal. We can consider two conceptual classifications in the XIFI Federation.

- Single users (as developers, experimenters, end users, etc.) are users that don't belong to any organization, and want to interact with the Portal. Therefore, they are registered in IdM. For each component, they can see the functionalities that are assigned directly to these IdM users.
- Organization is a legal entity or a group of individuals, whose employees or members can get an account and interact with the federated platform on behalf of the group and not individually, as the previous one. This approach allows managing different scopes. In the federated environment, we consider the following organizations:
 - Infrastructure owners and operators are part of the federation. They provide the capabilities of the different regions. There is only one organization per node (region) and the organization name follows a common pattern 'RegionName'+ 'Node', for example TrentoNode, LannionNode, BerlinNode, etc.
 - Technology Providers are responsible to create and publish new GEs/SEs in the platform. The stakeholders can find and use them, through the portal,
 - Infrastructure Tools (EU Projects FI-PPP Phase 2 as FITMAN, FISPACE, etc.) want to take advanced and use the capabilities of the platform. They make the most out of the cloud capabilities to manage their own instances and they can also publish and register their own SE in the public catalogue.

- Other organizations: There are more kinds of organization associated to the stakeholders [11] as SME, research institutions, web entrepreneurs groups, etc.
- Federator is not directly identified as a kind of stakeholders. However, it is necessary to define an internal IdM organization, which is responsible to manage and maintain the federated platform.

Every IdM user can belong to different organizations, for example he can belong to the Trento Node, and also, be part of the Federator Organization. Hence, the user will have different permissions for this component depending on the organization that he uses to connect with the platform (TrentoNode and Federator), as it is depicted in Figure 7.

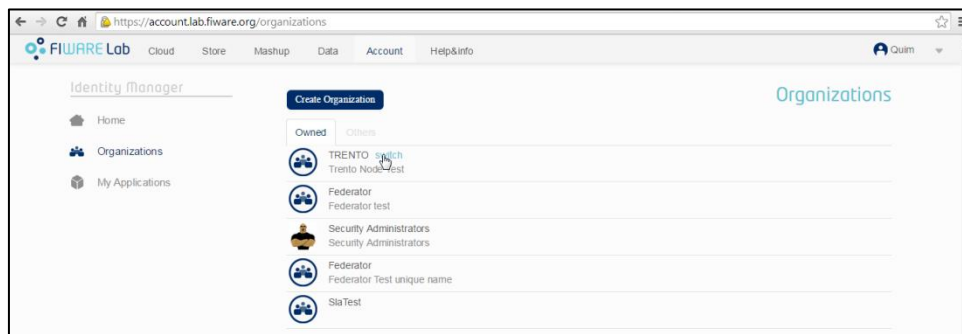


Figure 7: List of organization which the user belongs

The different components have to define and create the appropriate roles, which cover the components' functionalities (these roles are described in the D4.4 BaseLine tools v2 [4]). Afterwards, the components have to authorize the members to see these functionalities, through the associated roles. Members of this components can be directly users or/and organizations. These relationships are depicted in the Figure 8.

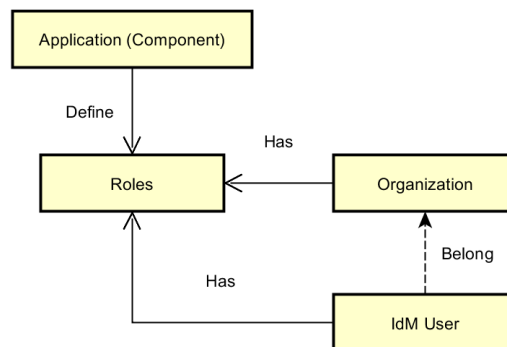


Figure 8: Relationships between the Components/Roles/Organizations/users

When the different components ask to the IdM for the roles of one user, the IdM will return the roles of this IdM user as individual user and also the roles as member of each organization to which he belongs. As it is depicted in the Figure 9:

```

{
  id: 2826,
  actorId: 4213,
  nickName: "Quim",
  displayName: "Quim",
  email: "qiranzo@yahoo.com",
  app_id: 767,
  app_slug: "sla-idm",
  roles: [
    {
      id: 106,
      name: "Provider"
    },
    {
      id: 15642,
      name: "Consumer"
    }
  ],
  organizations: [
    {
      id: 524,
      actorId: 4059,
      displayName: "TRENTO",
      roles: [
        {
          id: 8252,
          name: "IO"
        }
      ]
    },
    {
      id: 496,
      actorId: 3841,
      displayName: "Federator",
      roles: [ ]
    }
  ]
}

```

Figure 9: JSON message with the user/organization and roles

This IdM message (JSON format) includes all the information that the application (component) needs to know in order to identify the authenticated user and his roles. For example, in the above figure, the component can identify:

- **actorId**: unique identifier into the IdM.
- **nickname**: Nickname of the user.
- **displayName**: Name of the user.
- **app_id**: identifier of the component (application).
- **roles**: List of roles for this user as ‘single user’ for this component (app_id).
- **organization**: List of organizations which this user belongs to. The component can identify in these elements the following data:
 - *actorId*: unique identifier of this organization.
 - *displayName*: Name of this organization.
 - *roles*: List of roles for this user when he wants to connect on behalf of this organization.

Hence, the user has the roles “Provider” and “Consumer” as individual user and the role of “IO” when he is identified as a part of the “TRENTO” organization.

The component allows the user to change between the different organizations, through a list of available organization in the header, as it is depicted in Figure 10.

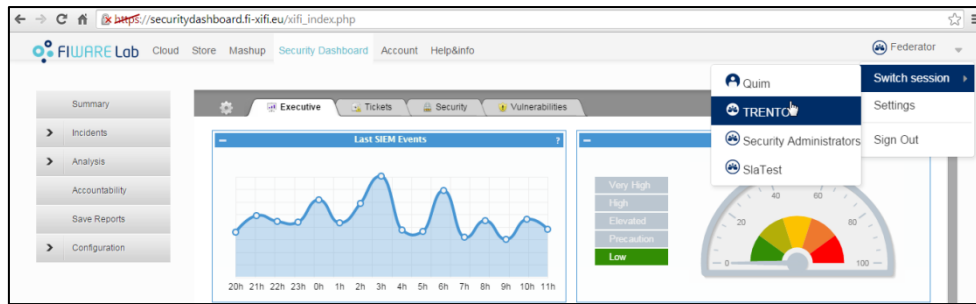


Figure 10: Change between different organizations in the Security Dashboard component

The organizations are responsible to maintain their members and their roles. Hence, the applications delegate the task of maintenance (assign roles and members) to the different organizations, which have been registered previously by this application. In the following figure, it is depicted how the 'TRENTO' organization assigns the 'IO' role for the 'SLA Dashboard' application.

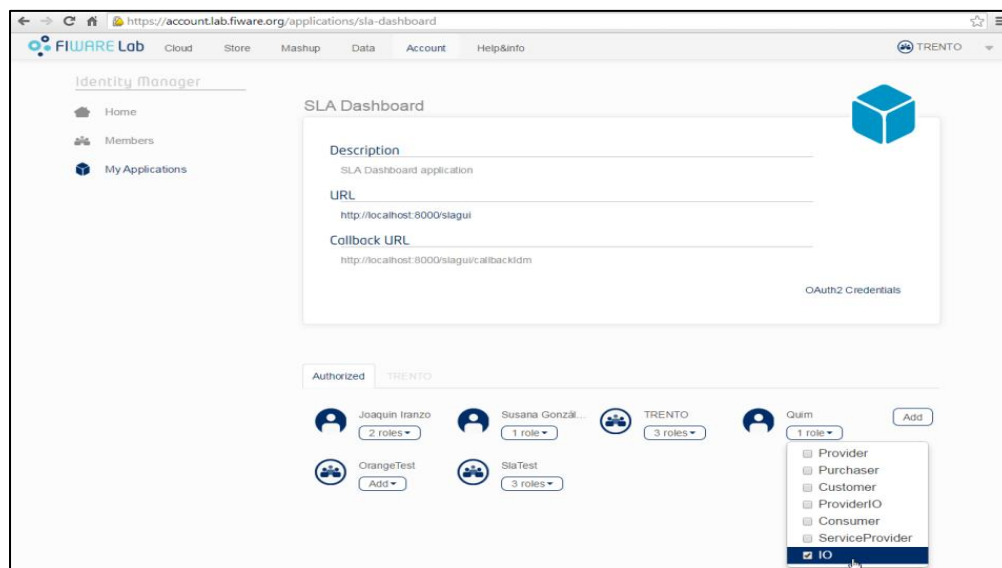


Figure 11: Organization assigns roles to its members for one application

2.4 Monitoring Dashboard

The Monitoring Dashboard is a collection of widgets (graphical user interfaces) of different monitoring tools for the various XIFI components deployed in the Federation. It provides Developers, Infrastructures Owners and Federation Manager with live and historical data about the status of the Federation environment resources in XIFI. It retrieves data from XIFI monitoring components such as Federation Monitoring and Network Active Monitoring.

The monitoring Dashboard consists of two components: VMs monitoring tool and NAM monitoring tool:

- **VMs monitoring tool** (integrated in the Cloud Portal) is in charge of monitoring the performance of deployed virtual machine instances. It obtains data from Federating Monitoring API in order to provide the user with the CPU, Memory and Disk status of its

deployed Virtual Machine in real-time or based on historical data.

- **NAM monitoring tool** supporting XIFI Infrastructure Owners provides information about the network status of the connections between XIFI nodes, including monitoring data about bandwidth, latency and packet loss, according to real-time data or historical information.

Depending on the user role, data is shown to the user as follows, and that is the added value to the XIFI Portal.

- ***Federation manager***
 - View records of current/historical all nodes monitoring data.
 - View records of current/historical data for all VMs (virtual machine) running on all nodes.
 - View records of current/historical data for each service running on all the nodes.
 - View records of current/historical data for each network element (interfaces) running on all nodes.
 - View records of current/historical data for each other network element (not only interfaces) running on all nodes.
 - View records of current/historical data for end2end connectivity tests.
- ***Infrastructure owner***
 - View records of current/historical local node monitoring data.
 - View records of current/historical data for each VM running on a local node.
 - View records of current/historical data for each service running on a local node.
 - View records of current/historical data for each network element (interfaces) running on a local node.
 - View records of current/historical for each other network element (not only interfaces) running on a local node.
- ***Future Internet developer***
 - View records of current/historical data for his/her VM running on a node.
 - View records of current/historical data for his/her service running on a node.

The Monitoring Dashboard component is part of the Monitoring GUI architectural component, thus it is integrated in both portals: FIWARE Lab and FIWARE Ops.

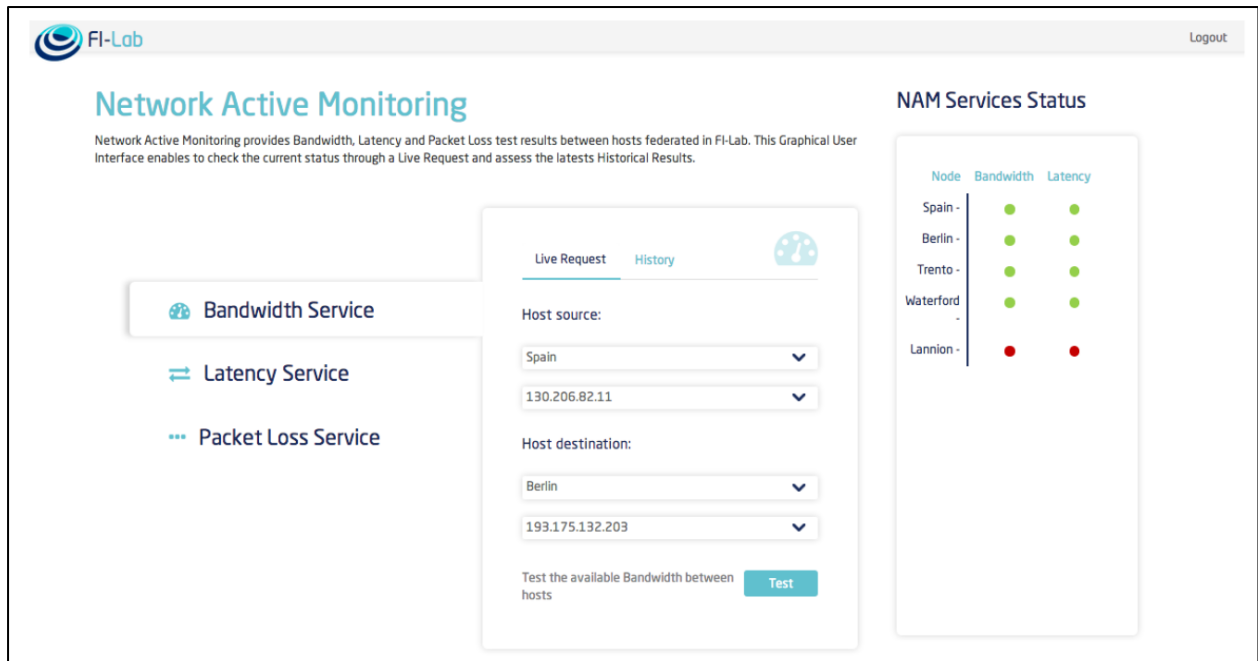


Figure 12: Monitoring Dashboard: NAM dashboard

2.5 SLA and Accounting Dashboard

SLA and Accounting Dashboard provides mechanisms to support the management of service level agreements in the federated environment, based on WS-Agreement specification [17]. It allows direct interaction with different actors through the graphical user interface that is part of the FIWARE Lab portal.

Currently, and depending on the user role:

- **Future Internet developers:** are able to visualize their agreements and/or violations through a GUI, which also provides recovery actions (actions after a violation). Developers can view the available metrics for the service and decide the boundaries which should be monitored.
- **Infrastructure Owners:** are able to efficiently manage their services, and to take action if it is deemed necessary in case of service level agreement violations. Infrastructure Owners are also able to indicate the QoS metrics that can be monitored on their facilities.
- **Providers:** indicate the QoS metrics that can be monitored on their infrastructures.
- **The End Users:** define and monitor SLA's.

Based on SLA management standards, it covers the SLA lifecycle and allows the evolution of the lifecycle according to the project needs. The component relies on the knowledge of the federated environment and of the infrastructures in order to define service characteristics and QoS for the offered service.

This tool is also integrated with the Resource Catalogue and Recommendation Tool component for providing SLA details of the resources to users while they are browsing, searching and comparing them, and also for the Recommendation Tool for supporting the user taking care also about SLA parameters.

SLA and Accounting Dashboard is a tab/section accessible from the header of FI-Lab portal, thus users will easily identify this tool on a direct link from the portal. It will be integrated in the portal and with FI-Lab IdM for XIFI Marketplace next implementations.

SLA is composed of the following components:

- **SLA Dashboard (GUI Layer)** shows developers and providers all the functionalities through the GUI. It interacts with the SLA Manager (backend layer) and then process the result visually.
- **SLA Management (Federation Layer)** provides all the backend functionalities. It can be considered as part of the Federation Layer, being responsible for both collecting the information of Federation Monitoring and exposing the SLA information to the rest of the components (SLA Dashboard, Cloud Portal, Recommenders, etc.)

The SLA Dashboard subcomponent is composed by the following modules:

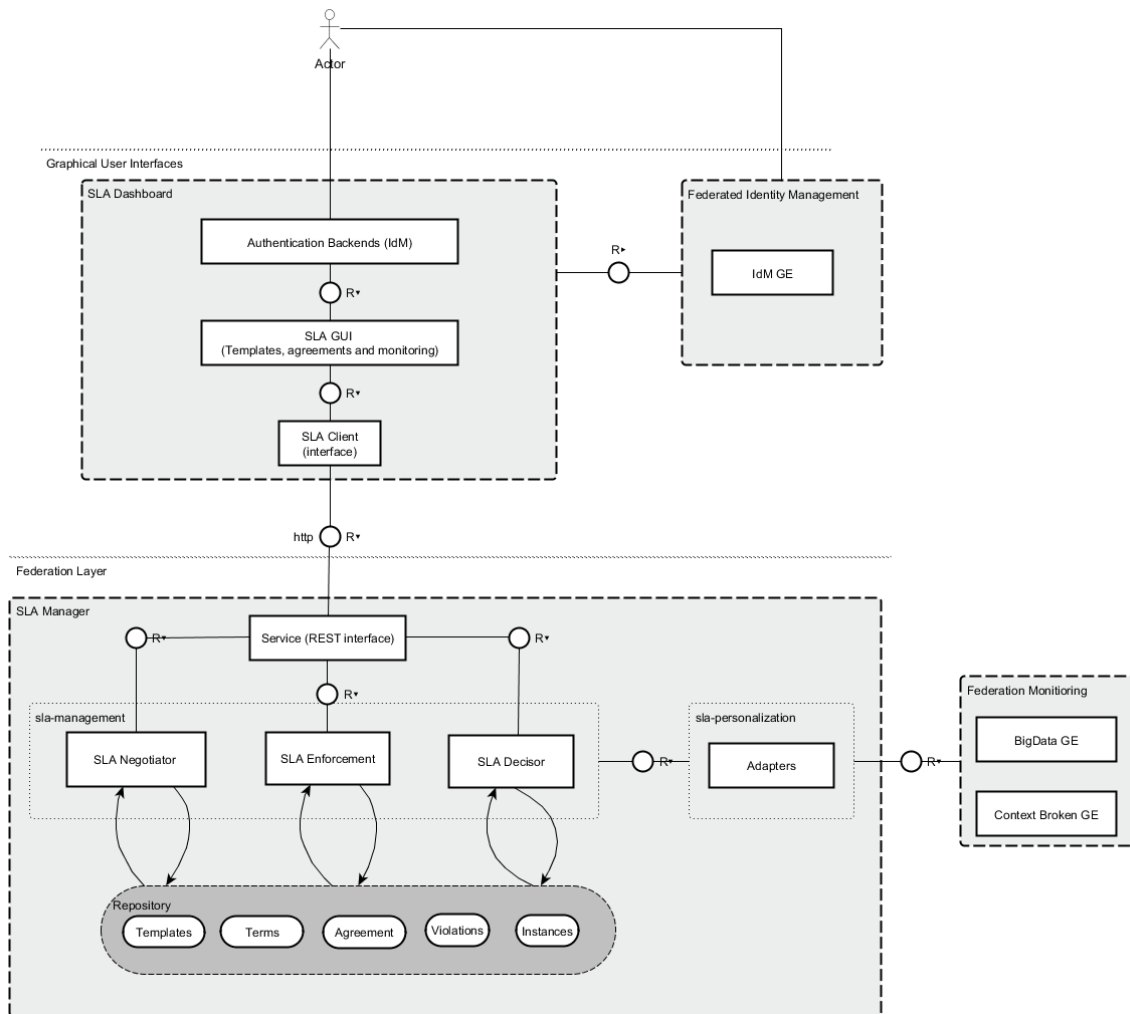
- **Authentication Backend (IdM)** identifies users and delegates their authentication to the Federation Identity Manager. It manages the authorization of functionalities and organizations.
- **SLA GUI visualizes the SLA lifecycle**, enabling the users to manage templates, agreements and to show their status through the dashboard.
- **SLA Client interacts with the SLA Manager**, since the SLA Dashboard doesn't contain the service layer, SLA Client consumes the API that is exposed by SLA Manager

The SLA Management subcomponent is divided in the following modules:

- *service (REST interface)* exposes the backend functionalities through the REST interfaces. It uses the "sla-core" to execute actions without a direct access to the repository.
- *sla-management* contains the backend business code (publish, negotiation, enforcement, recovery).
- *repository* accesses the main entities (defined in the class diagram).
- *sla-personalization* contains all the specific adapters for every project. This helps to change the adapters easily even if they are changed in the origin of the monitoring, without affecting the sla-management modules.

The SLA Management subcomponent interacts with other modules:

- The SLA Management relies on monitoring data in order to guarantee the service level agreement agreed on the previous interactions. Hence the Federation Monitoring component provides to this module all the required monitoring information.
- BigData GE is responsible for providing historical monitoring data.
- Context Broker GE is responsible for providing the current monitoring data.



There are several API methods exposed for the SLA dashboard usage:

- Adding a new provider:

```
$curl -u user:password -H "Accept: application/xml" -H "Content-type: application/xml" -d@provider-trento.xml localhost:8080/sla-service/providers -X POST
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<message code="201" message="The provider has been stored successfully in the SLA Repository Database. It has location http://localhost:8080/sla-service/providers/Trento"/>

<provider>
  <uuid>Trento</uuid>
  <name>Trento</name>
</provider>
```

- Listing existing providers:

```
$ curl -u user:password localhost:8080/sla-service/providers
```

- Adding a new template:

```
$ curl -u user:password -H "Content-type: application/xml" -d@Template_vm_Trento.xml localhost:8080/sla-service/templates -X POST
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<message code="201" message="The agreement has been stored successfully in the SLA Repository Database. It has location http://localhost:8080/sla-service/templates/template_vm-Trento:193.205.211.85"/>
```

- Listing existing templates:

```
$ curl -u user:password -H "Accept: application/xml" localhost:8080/sla-service/templates
```

- Adding a new agreement:

```
$ curl -u user:password -H "Content-type: application/xml" -d@agreement_vm_Trento.xml localhost:8080/sla-service/agreements -X POST
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<message code="201" message="The agreement has been stored successfully in the SLA Repository Database. It has location http://localhost:8080/sla-service/agreements/agreement-vm-Trento:193.205.211.85-MonitoringB"/>
```

- Getting a specific agreement

```
$ curl -u user:password -H "Accept: application/xml" localhost:8080/sla-service/agreements/agreement-vm-Trento:193.205.211.85-MonitoringB
```

- Getting an enforcement job

```
$ curl -u user:password -H "Content-type: application/xml" localhost:8080/sla-service/enforcements/agreement-vm-Trento:193.205.211.85-MonitoringB
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<enforcement_job>
  <agreement_id>agreement-vm-Trento:193.205.211.85-MonitoringB</agreement_id>
  <enabled>>false</enabled>
  <last_executed xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
</enforcement_job>
```

- Starting an existing enforcement job

```
$ curl -u user:password -H "Content-type: application/xml" localhost:8080/sla-service/enforcements/agreement-vm-Trento:193.205.211.85-MonitoringB/start -X PUT
The enforcement job with agreement-uuid agreement-vm-Trento:193.205.211.85-
```

MonitoringB has started

- Checking an enforcement job's status

```
$ curl -u user:password -H "Content-type: application/xml" localhost:8080/sla-
service/enforcements/agreement-vm-Trento:193.205.211.85-MonitoringB

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<enforcement_job>
  <agreement_id>agreement-vm-Trento:193.205.211.85-
MonitoringB</agreement_id>
  <enabled>true</enabled>
  <last_executed xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
</enforcement_job>
```

- Getting the list of violations

```
$ curl -u user:password localhost:8080/sla-service/violations?agreementId=agreement-
vm-Trento:193.205.211.85-MonitoringB
<?xml version="1.0" encoding="UTF-8"?>
<collection href="/violations">
<items offset="0" total="1">

<violation>
  <uuid>ab39160f-2340-43bb-8e1f-94f41146d808</uuid>
  <contract_uuid>agreement-vm-Trento:193.205.211.85-Monitoring1</contract_uuid>
  <service_scope></service_scope>
  <metric_name>Performance</metric_name>
  <datetime>2014-09-11T11:28:22Z</datetime>
  <actual_value>80</actual_value>
</violation>
</items>
</collection>
```

Before users and providers register in the application, the SLA Dashboard has to be registered and configured in the Federated Identity Manager (see the Installation guide section).

These are the roles that exist in the SLA Manager:

- **Consumer.** This role indicates that the user, (previously identified), is the consumer in the agreement. Thus, the dashboard shows the associated agreements and their status. Consumers are able to create and manage the agreements.
- **Service Provider / IO:** These 2 roles indicate that the user, (previously identified), is the provider or infrastructure owner. Hence, the dashboard allows him to manage and see all the agreements for this provider and lets the user create and manage the templates.
- **Federator:** Responsible to manage the application in the IdM and maintenance.

User is redirected to the agreements index page after a successful login

Dashboard for Consumers

The user can see his agreements and their status.

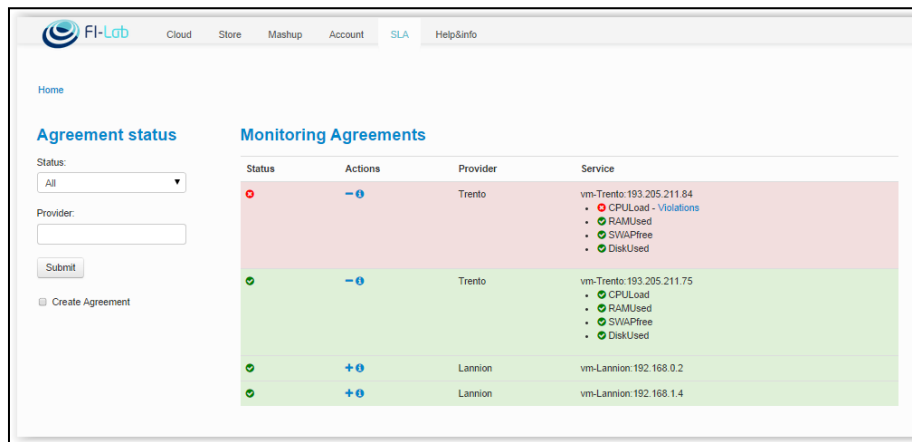



Figure 13: Agreement Dashboard

- In the left panel, the customer can filter his agreements by the status and the providers.
- In the center panel, there are all his agreements with the summary and their status.
- In order to see the status of the agreement, the customer



Figure 14: Status of the agreements. Able to click on icon 

- The consumer can see the associated metrics and whether these have been violated or not. If the metric has not been fulfilled, the customer can see the violations details; clicking in the "violations" link.

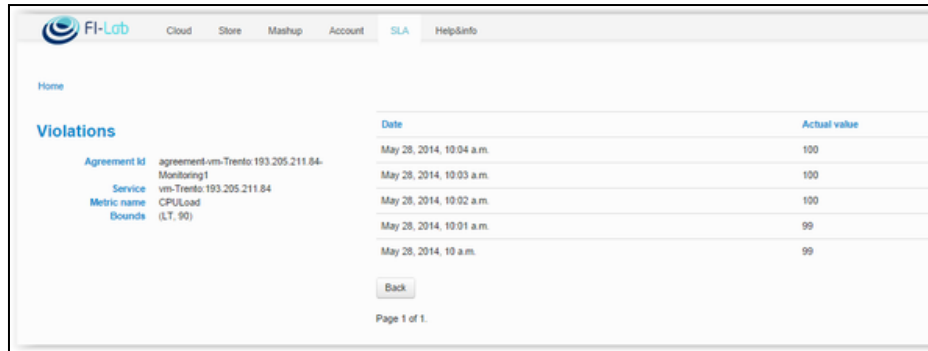



Figure 15: Violation details

- The customer can click on the icon  and see the agreement details, a summary of the guarantee terms and violations per date.

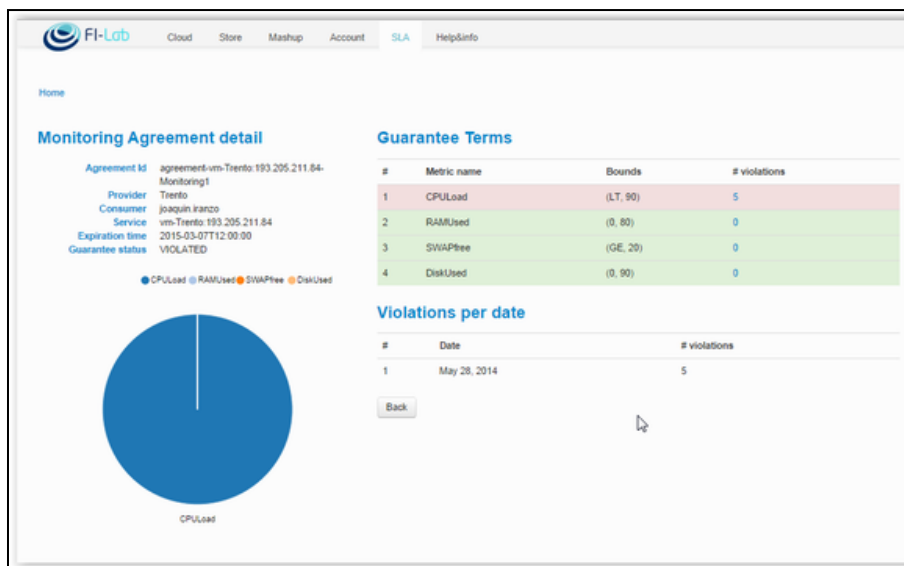


Figure 16: Info of the agreement

- The customer can create the agreements clicking in the check box "Create Agreement":

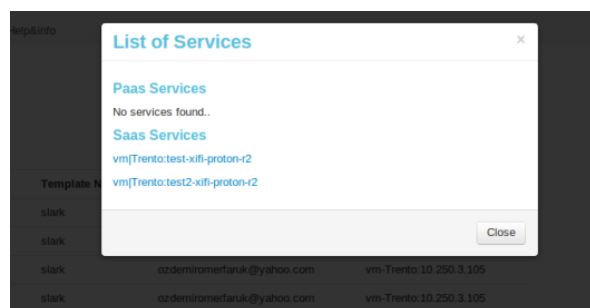


Figure 17: List of services

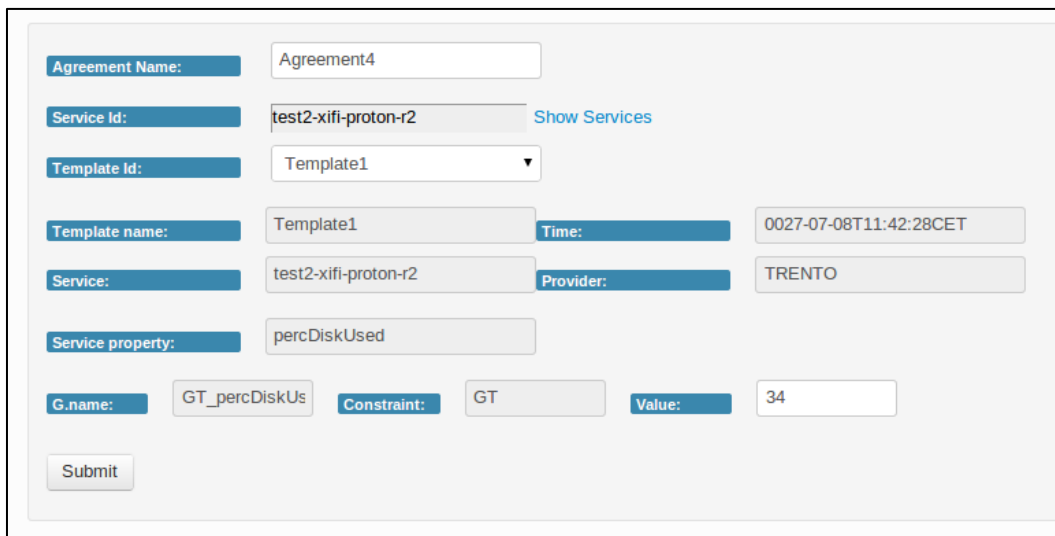




Figure 18: Create an agreement

Dashboard for Providers

There are two main pages in the dashboard for the providers.

- Main page (home menu)

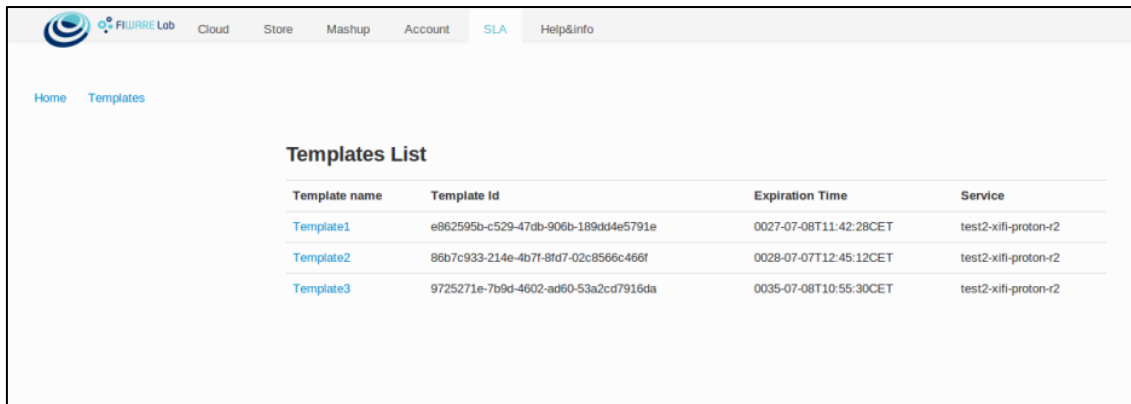
The provider can see his agreements and the status of them.

- It is possible to search by status of the agreement and by associated customer.
- View the detailed summary of the agreement, clicking in the icon 
- The provider can see the details of the violation, clicking in the “Violations” link.
- The provider can click on the  and see the agreement details and a summary of the guarantee terms and the violations per date.

- Template page (Template menu)

The providers can manage their templates.

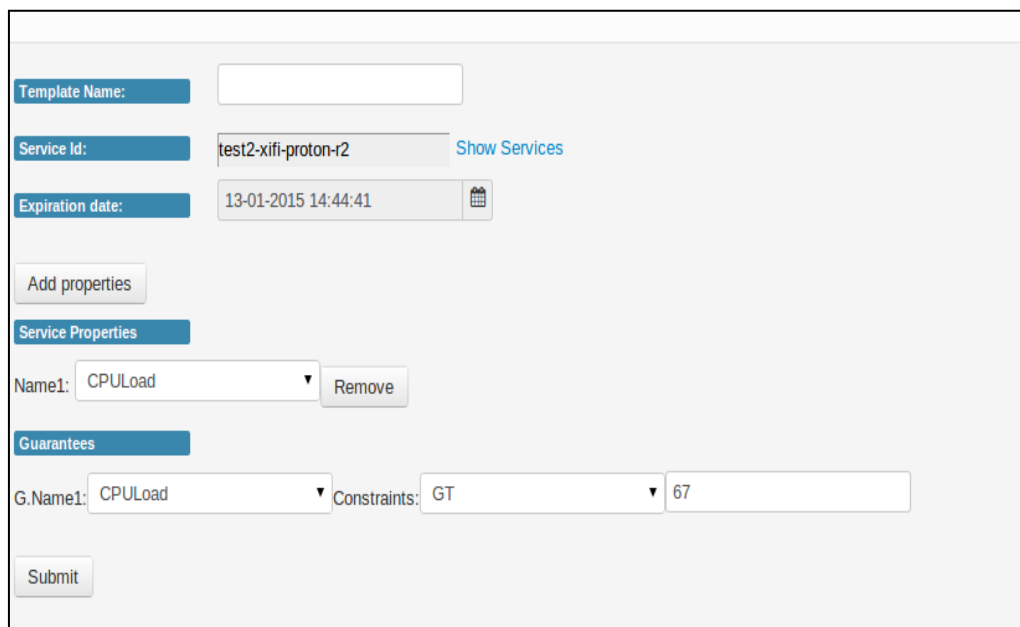
- See the list of templates available for the provider and its details.



Template name	Template Id	Expiration Time	Service
Template1	e862595b-c529-47db-906b-189dd4e5791e	0027-07-08T11:42:28CET	test2-xifi-proton-r2
Template2	86b7c933-214e-4b7f-8fd7-02c8566c466f	0028-07-07T12:45:12CET	test2-xifi-proton-r2
Template3	9725271e-7b9d-4602-ad60-53a2cd7916da	0035-07-08T10:55:30CET	test2-xifi-proton-r2


Figure 19: List of templates for providers

- Create a new template



Template Name:

Service Id: test2-xifi-proton-r2 [Show Services](#)

Expiration date: 13-01-2015 14:44:41 

[Add properties](#)

Service Properties

Name1: CPUload

Guarantees

G.Name1: CPUload Constraints: GT

Figure 20: Add template for providers

2.6 Security and Privacy Dashboard

Security and Privacy Dashboard supports the management and visualization of security incident-related events and data. Beside the core functionalities offered by such a dashboard, the Security Dashboard also provides reporting capabilities.

The FIWARE Security Monitoring GE (which includes the component Service Level SIEM) is, working together with Security Probes installed in the slave nodes, responsible for the collection and correlation of security monitoring data. These Security Probes are installed in each slave Node as follows:

- At the level of the entire node in a specific server, to collect accountability events generated

by the Access Control GE and events generated by network devices such as firewalls installed in the infrastructure. This server will have installed a syslog server so the events can be sent from the different sources. Each VM deployed will be preconfigured to send all the logs through syslog to this Security Probe in order to be monitored.

- Inside each virtualization server in the node, to collect events from different data sources depending on the plugins activated (e.g. syslog, nagios, snort...).

Security and Privacy Dashboard is also a tab/section accessible from the header of FIWARE-Ops portal, because it is focused on Infrastructure Owner and Federator Manager user roles, providing security and privacy monitoring and reporting added value.

Figure 21 shows Security and Privacy Dashboard graphical user interface.

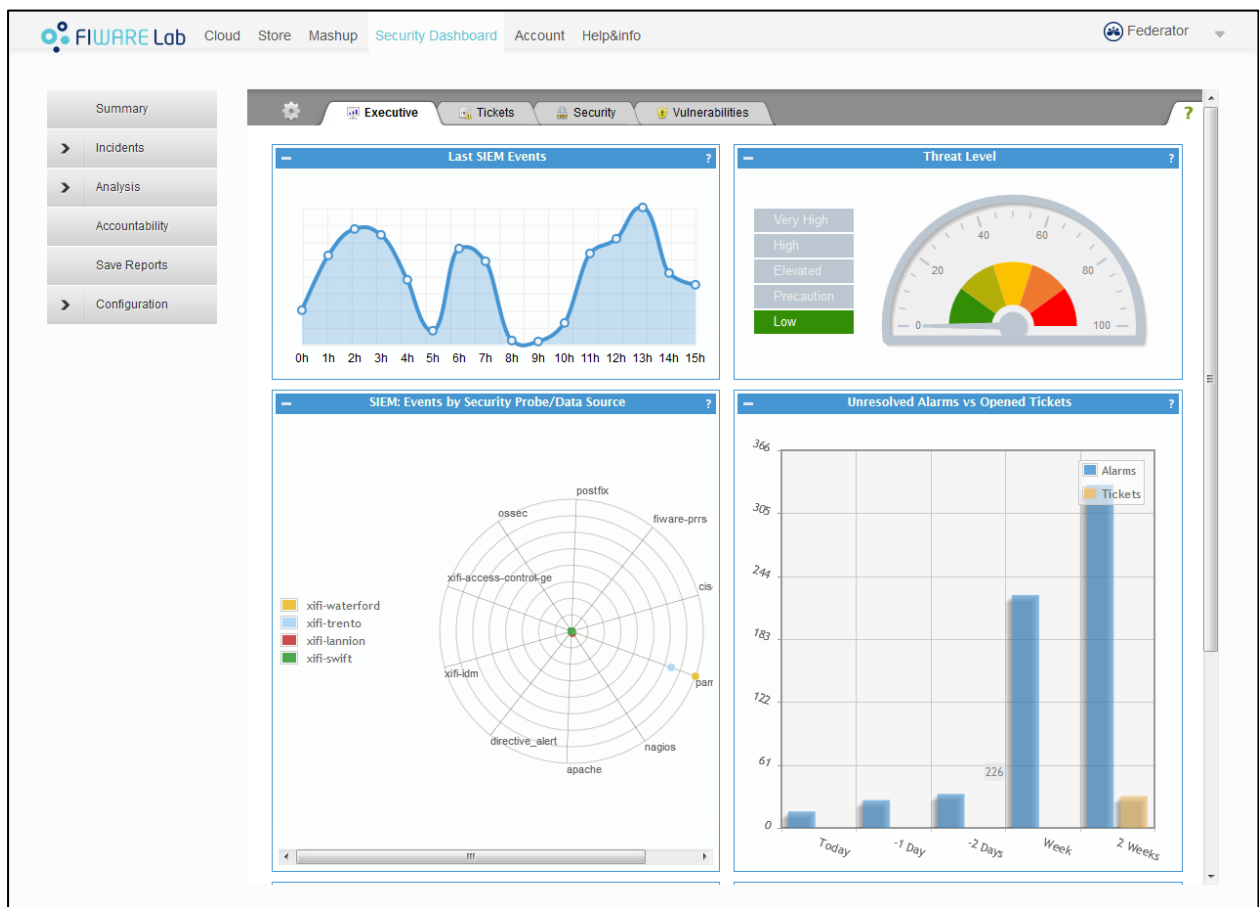


Figure 21: Security Dashboard: main dashboard

2.7 Interoperability Tools

The Interoperability Tool supports development and testing of FI-WARE based applications and services; in particular the tool focuses on interoperability problems. The main users of the tool are Future Internet Developers, supporting their work in two ways:

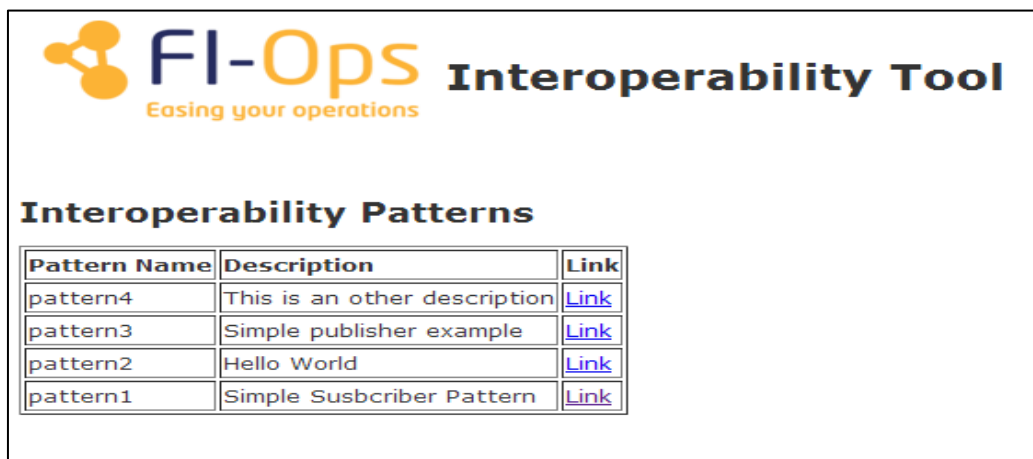
- The developers can test whether their application interoperates correctly with GE instances deployed across XIFI federated facilities.

- During design and implementation of new applications and services, the developer can use the tool to observe and learn best practices on the usage of GEs; in order they quickly build their applications based upon these established patterns of behaviour.

Infrastructure owners can also use the tool to specify compliance of developed and deployed services. The tool provides suites of executable patterns that can determine the extent to which a single GE can interoperate with applications and other GEs.

Interoperability Tools can be reached through the FIWARE-Ops portal, thus users will easily identify this tool on a direct link from the portal, providing added value to solve interoperability problems.

Figure 22 and Figure 23 show Interoperability Tool graphical user interface.

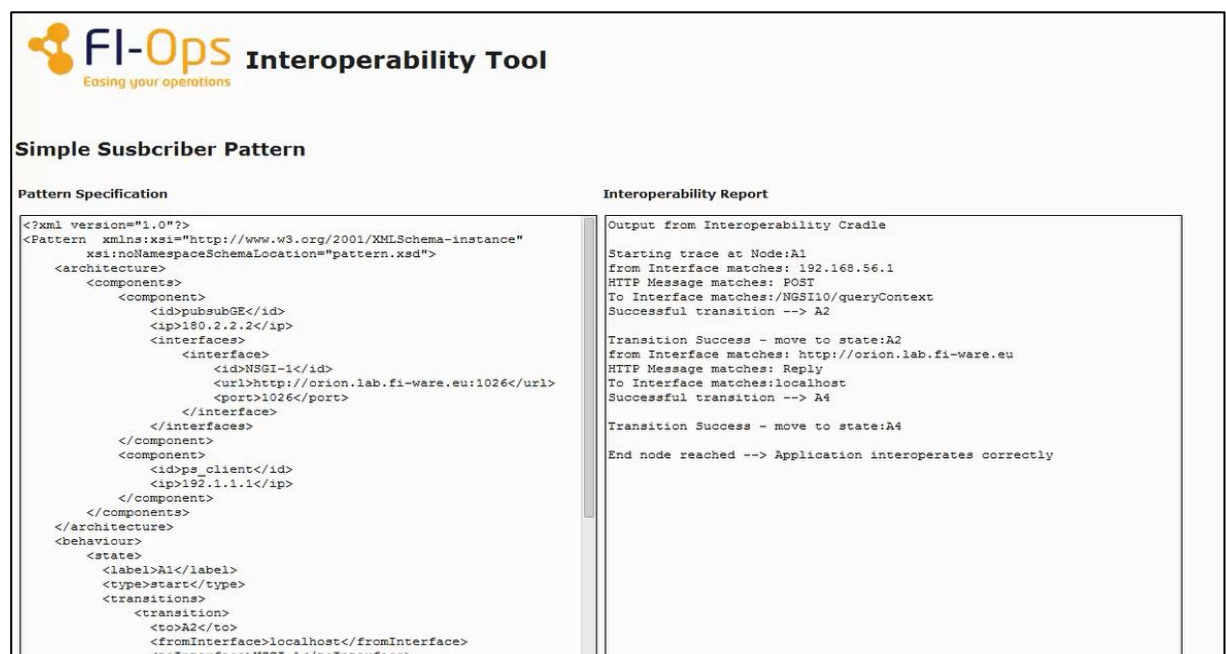


FI-Ops Interoperability Tool
Easing your operations

Interoperability Patterns

Pattern Name	Description	Link
pattern4	This is an other description	Link
pattern3	Simple publisher example	Link
pattern2	Hello World	Link
pattern1	Simple Subscriber Pattern	Link

Figure 22: Display available patterns in the Interoperability Tool



FI-Ops Interoperability Tool
Easing your operations

Simple Subscriber Pattern

Pattern Specification

```
<?xml version="1.0"?>
<Pattern xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="pattern.xsd">
  <architecture>
    <components>
      <component>
        <id>pubsubGE</id>
        <ip>180.2.2.2</ip>
        <interfaces>
          <interface>
            <id>NSGI-1</id>
            <url>http://orion.lab.fi-ware.eu:1026</url>
            <port>1026</port>
          </interface>
        </interfaces>
      </component>
      <component>
        <id>ps_client</id>
        <ip>192.1.1.1</ip>
      </component>
    </components>
  </architecture>
  <behaviour>
    <state>
      <label>A1</label>
      <type>start</type>
      <transitions>
        <transition>
          <to>A2</to>
          <fromInterface>localhost</fromInterface>
          <toInterface>NSGI-1</toInterface>
        </transition>
      </transitions>
    </state>
  </behaviour>
</Pattern>
```

Interoperability Report

```
Output from Interoperability Cradle
Starting trace at Node:A1
from Interface matches: 192.168.56.1
HTTP Message matches: POST
To Interface matches:/NGSI10/queryContext
Successful transition --> A2

Transition Success - move to state:A2
from Interface matches: http://orion.lab.fi-ware.eu
HTTP Message matches: Reply
To Interface matches:localhost
Successful transition --> A4

Transition Success - move to state:A4
End node reached --> Application interoperates correctly
```

Figure 23: Result of interoperability testing and monitoring

2.8 Infographics and Status Pages

The Infographics and Status Pages component provides information on the infrastructure capacities and status of FIWARE Lab infrastructure services. The service is mainly intended for Developers and Federation Manager. The Infographics page provides via an infographics the available infrastructure capacities in FIWARE Lab. The Status page provides information on the status of the FIWARE Lab infrastructure services (e.g. Nova, Neutron, Cinder, Glance and others for a given node) and offers direct access to Jira support both for a specific node and for general portal issues. While the Status page is a service normally offered by cloud providers and other services providers, the Infographics presents an innovative and intuitive way to publish high-level information on the infrastructure.

Infographics and Status Pages is a tab/section accessible from the header of FIWARE Lab portal, thus it is easily accessible from the portal.

Below, Infographics and Status Pages images show its graphical user interface.

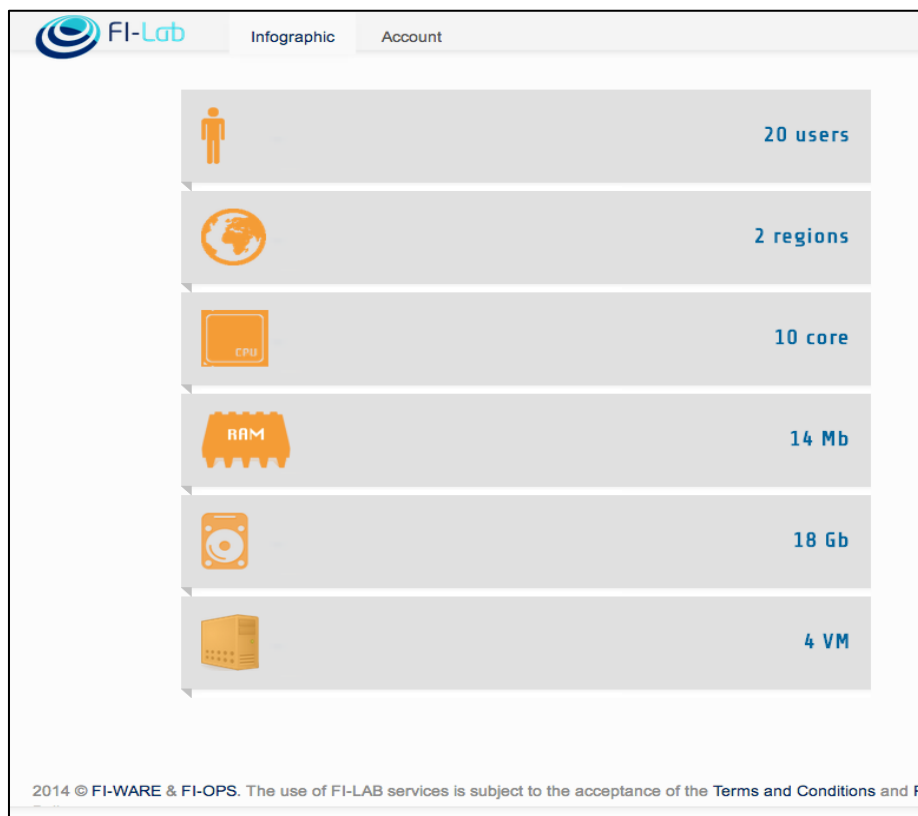


Figure 24: Infographics main page

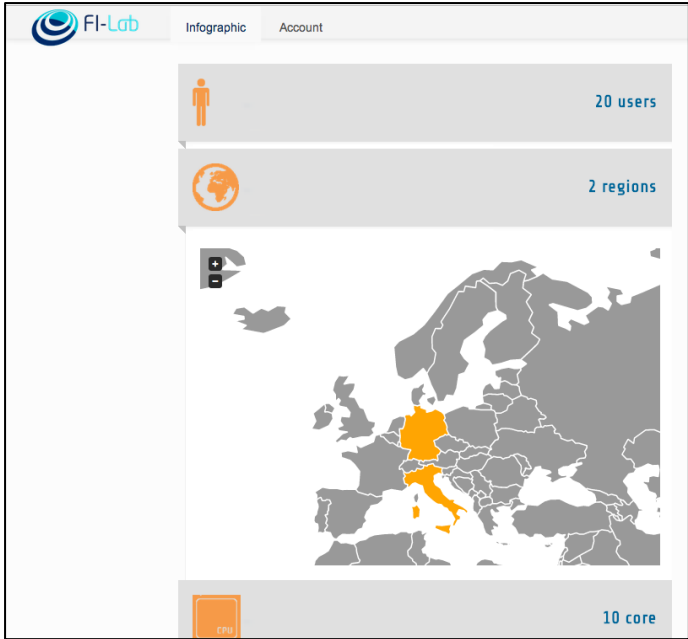


Figure 25: Infographics: access to detailed information by clicking on each panel

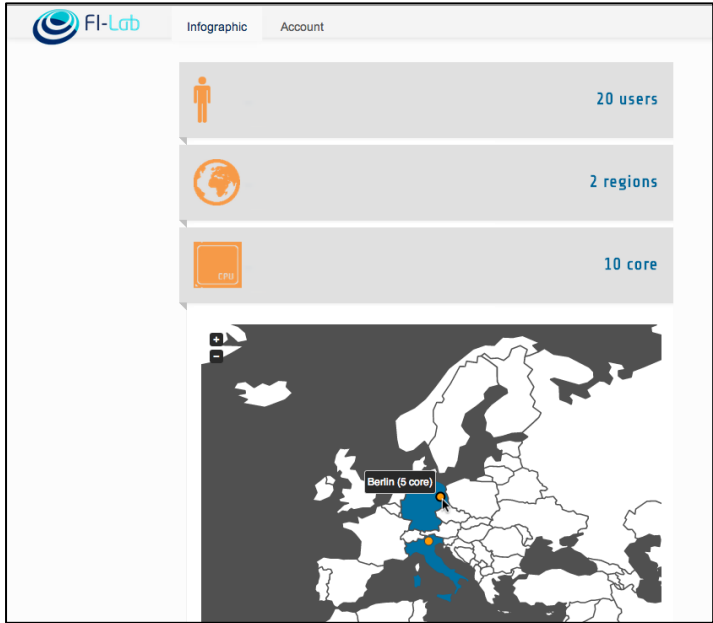


Figure 26: Infographics: example cores per node

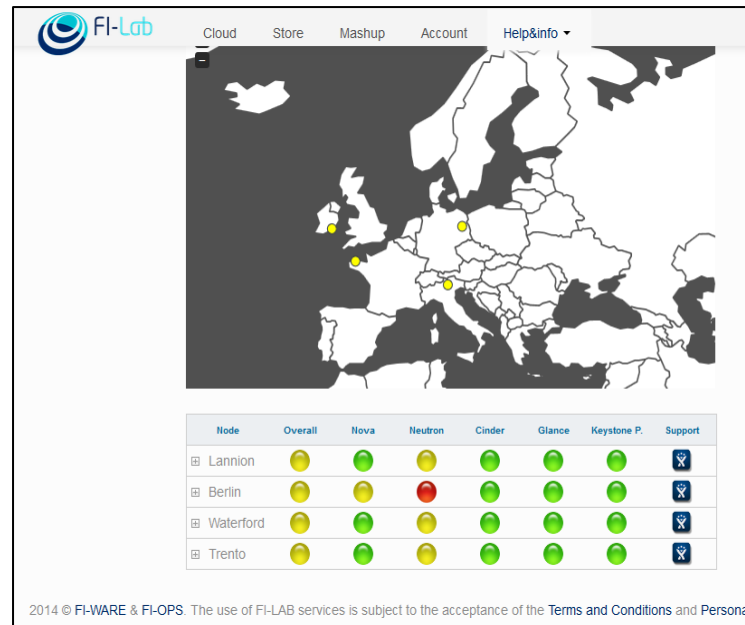


Figure 27: Status Pages main page

2.9 Federation Manager GUI

The Federation Manager GUI provides access to the federation through a graphical user interface and is the central registration point for new infrastructures and their services. The Federation Manager GUI is included in the Federation Manager component.

Once an Infrastructure Owner has registered at the Federation Manager GUI, has to fill in a compliance survey based on the Quick Online Test component. In this survey the Infrastructure Owner declares legal and technical compliance to the terms of the XIFI federation and provides contact information (e.g. management and support contacts).

When the infrastructure owner has completed the survey, the new request is shown to the Federation Administrator who can approve the incoming request. This approval activates the next step for the Infrastructure Owner, who is required to download, install and configure the Infrastructure Toolbox (ITBox).

After the successful installation and configuration, the Infrastructure Owner has to provide information concerning federated services currently running on the infrastructure. When this data is committed the Federation Administrator is able to trigger remote tests on these services.

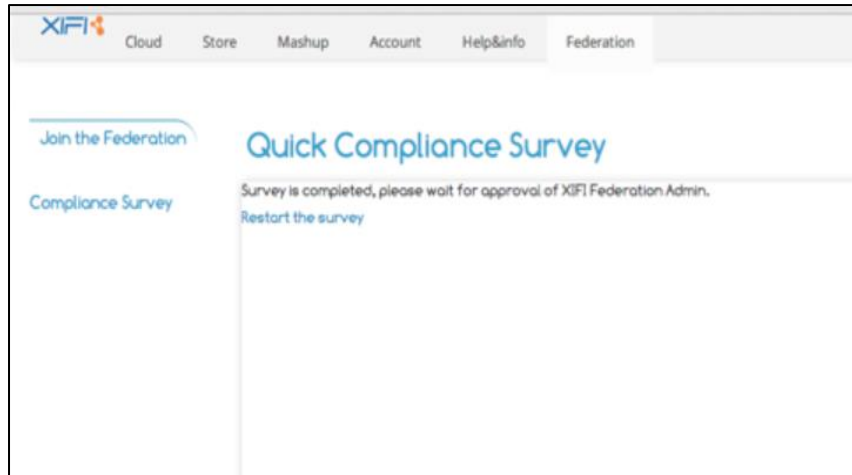


Figure 28: Federation Manager GUI - Compliance Survey

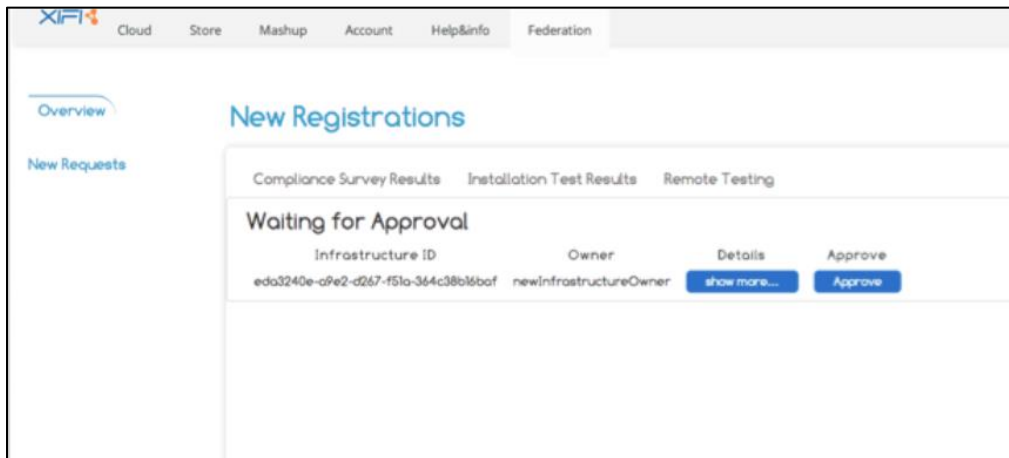


Figure 29: Federation Manager GUI: Federation Admin New Requests

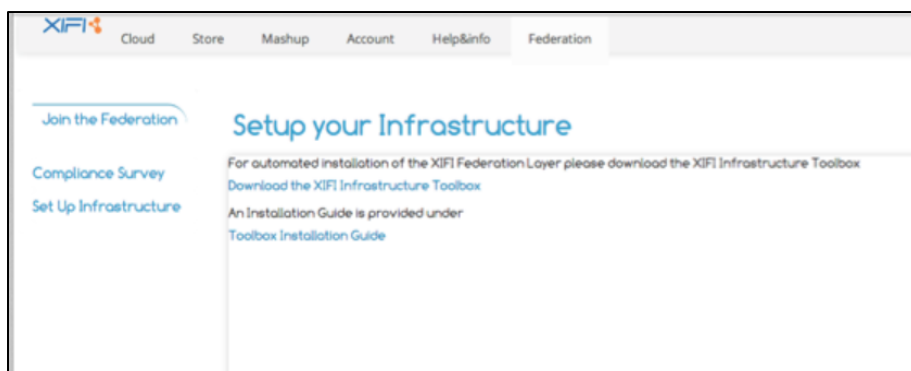


Figure 30: Federation Manager GUI: Infrastructure Toolbox (ITBox)

3 INTEGRATED MODULES IN XIFI MARKETPLACE

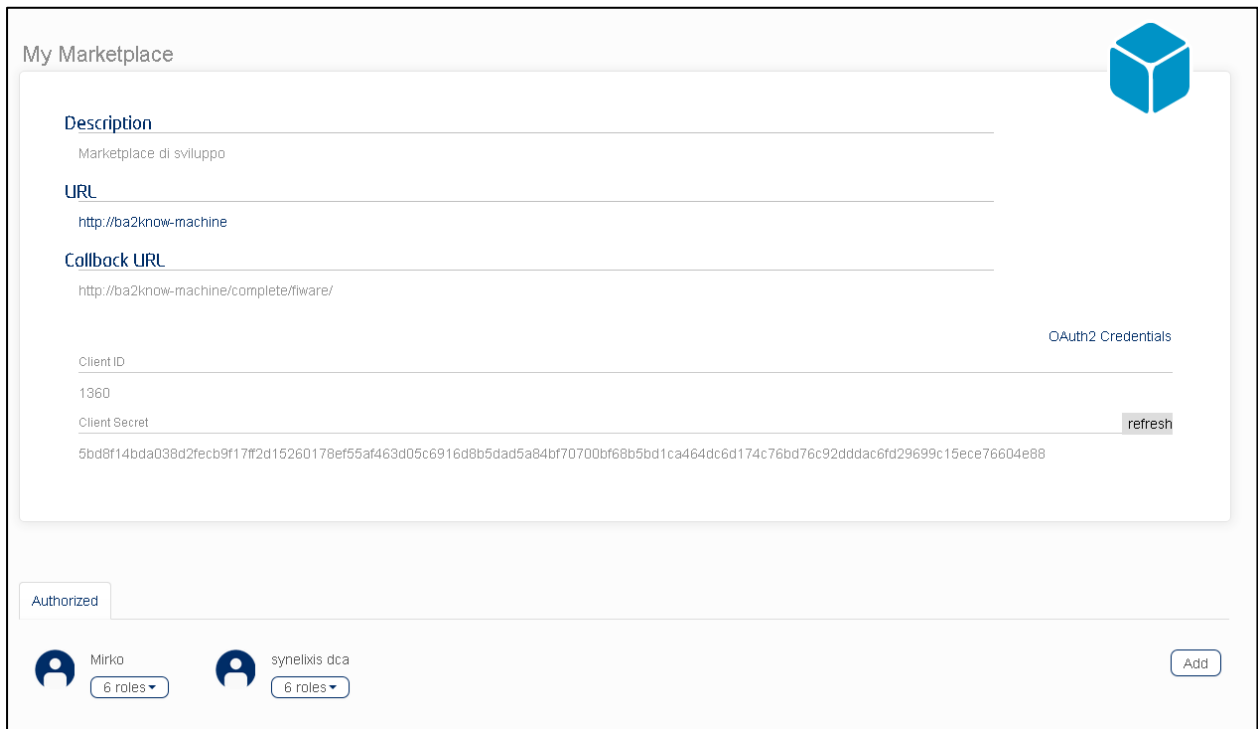
This section of the deliverable aims to describe the set of modules that has been implemented concerning XIFI Marketplace, in order to provide a complete description of the integration and relation of XIFI Marketplace with the different components involved. Moreover, each module description highlights the functionalities implemented, and their added value.

3.1 Federated Identity Management

Before accessing the WStore a user must be logged through a Federated Identity Managed, located in a XIFI node, or through an external federation (Single Sign On access).

Federated Identity Management enables to manage the user/organization permissions concerning a specific GE (in this case the WStore).

To do this you have to register the GE in the IDM through the “My Application” page; after a successful registration the IDM will generate a new couple of OAuth credentials Figure 31, these credentials will be used from the GE to “validate” an user and take his permissions.



My Marketplace

Description
Marketplace di sviluppo

URL
http://ba2know-machine

Callback URL
http://ba2know-machine/complete/fiware/

OAuth2 Credentials

Client ID
1360

Client Secret
5bd8f14bda038d2fecb9f17ff2d15260178ef55af463d05c6916d8b5dad5a84bf70700bf68b5bd1ca464dc6d174c76bd76c92dddac6fd29699c15ece76604e88

refresh

Authorized

Mirko 6 roles ▾

synelixis dca 6 roles ▾

Add

Figure 31: Application registration

Will be possible to assign new users to our application and assign to each user different roles Figure 32.

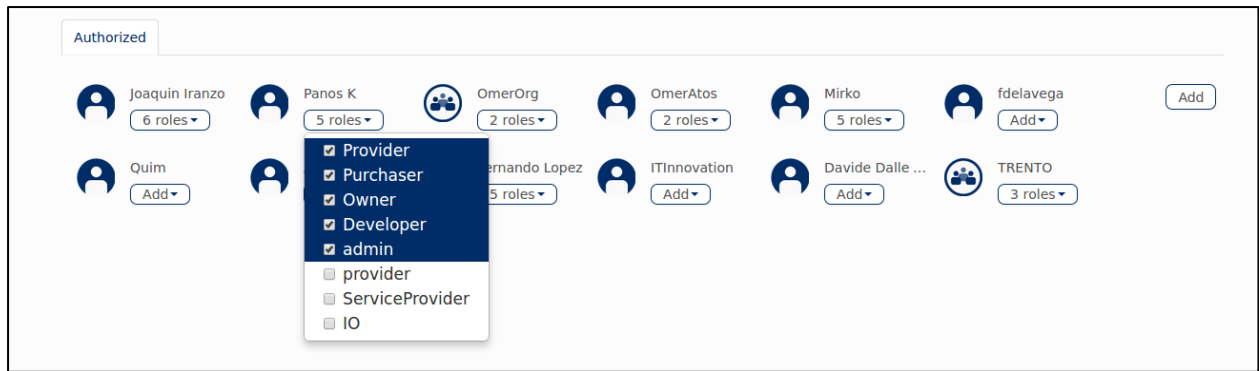


Figure 32: User's application

Each role determines a set of application functionalities that can be accessed (e.g. Developer role and Service Provider role).

In the WStore an user with Developer role can register a new SaaS but can't publish a PaaS as SaaS (Service Provider can).

Roles are managed through a specific page Figure 33; we can create new roles, remove roles and update existing roles, for each role we can define new permissions (functionalities).

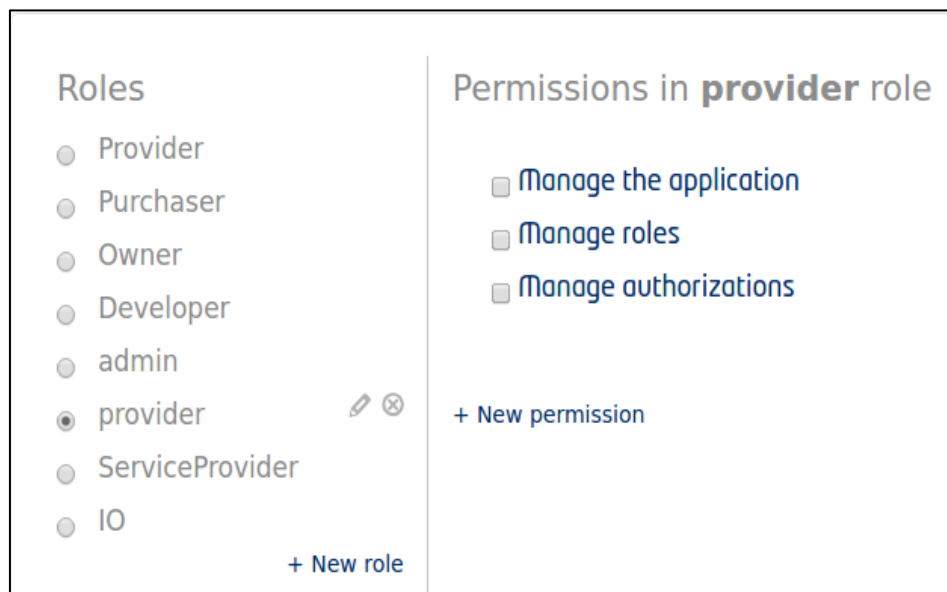


Figure 33: Role management

Through the IDM is possible to manage (create/update/delete) our organizations (group of users). Figure 34 list of my Organization and access to organizations of other users Figure 35: List of others Organization.

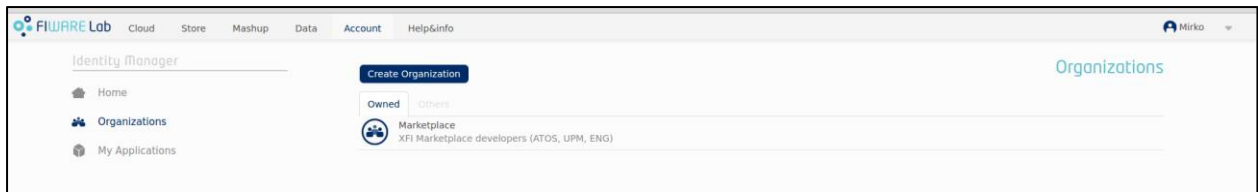


Figure 34: List of my Organization



Figure 35: List of others Organization



Figure 36: Switch from a user to an organization

It is possible to assign new users to organizations and modify their roles.

3.2 Cloud Portal

The Cloud Portal provides a support for the users of the XIFI cloud infrastructure and platform to manage their services and resources deployed in the cloud. The basic objective of the Cloud Portal is to facilitate the user to perform operations over the underlying infrastructure, including the creation of PaaS instances by IaaS images or by Blueprints templates. These PaaS instances will be shown in the Marketplace homepage and Resource Catalogue [22]:

Following steps show how a user can create an instance in the Cloud Portal.

- **Create a PaaS through an image**

First of all you have to select a node in which you will deploy a PaaS and after select an image to be deployed as PaaS.

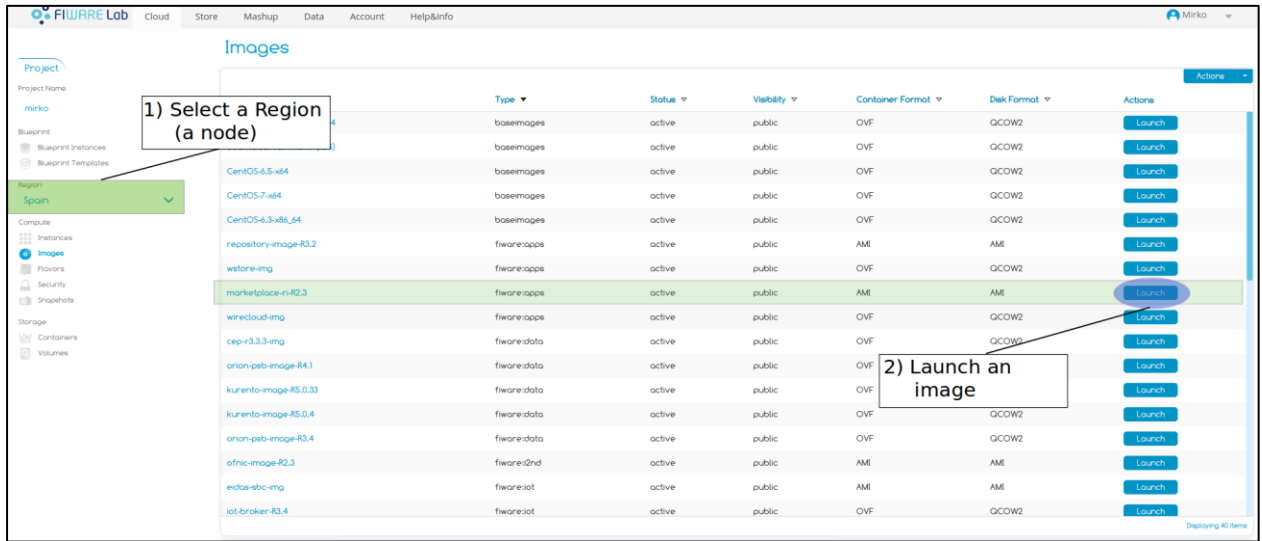


Figure 37: List of available images to deploy as PaaS

Below the steps to create a PaaS instance Figure 38, Figure 39, Figure 40 and Figure 41.

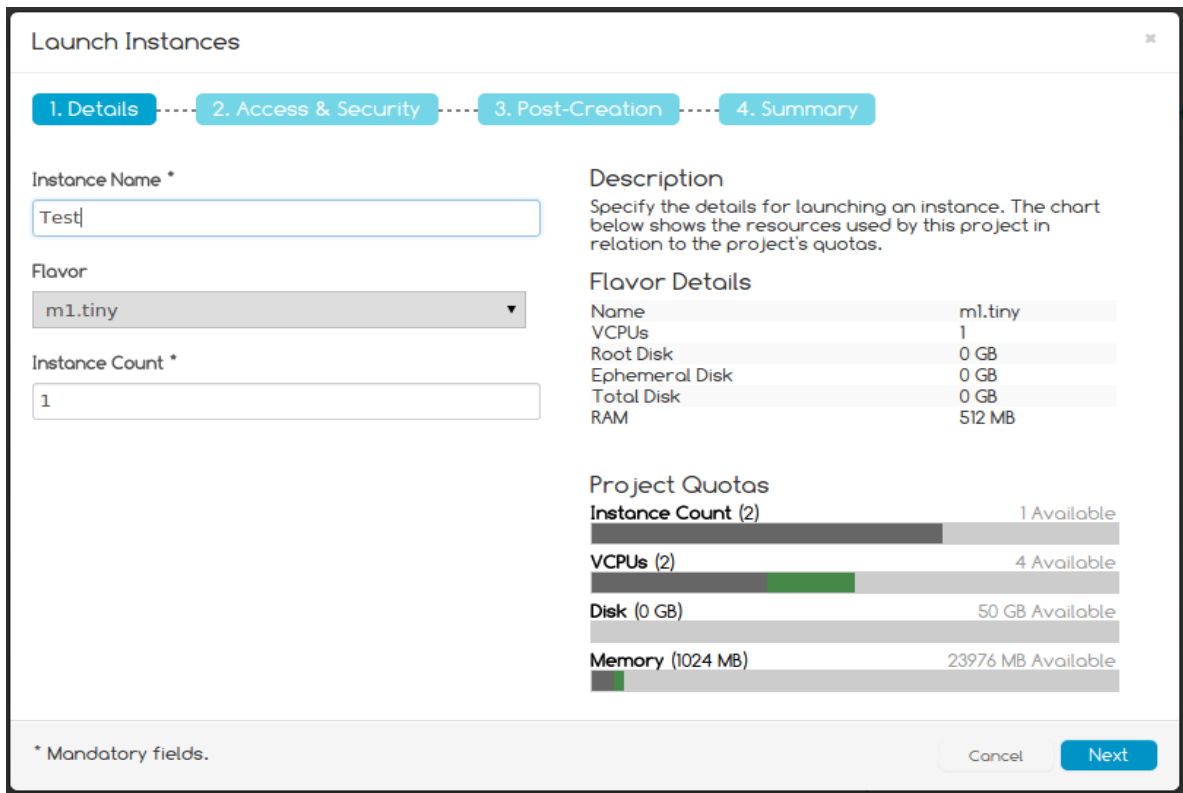


Figure 38: Deploy a PaaS (Instance creation)

Launch Instances ✕

1. Details
2. Access & Security
3. Post-Creation
4. Summary

Keypair

No keypairs available. ▾

Security Groups

default

Add new Security Group

Description

Control access to your instance via keypairs, security groups, and other mechanisms.

* Mandatory fields.

Back
Next

Figure 39: Deploy a PaaS (Access & Security)

Launch Instances ✕

1. Details
2. Access & Security
3. Post-Creation
4. Summary

Customization Script

Description

You can customize your instance after it's launched using the options available here. The "Customization Script" field is analogous to "User Data" in other systems.

* Mandatory fields.

Back
Next

Figure 40: Deploy a PaaS (Customization)

Launch Instances

1. Details
2. Access & Security
3. Post-Creation
4. Summary

Instance Name: Test
 Image: repository-image-R3.2
 Flavor: m1.tiny
 Instance Count: 1
 Keypair: No keypair selected. You will need a keypair to access the instance.
 Security Group: No security group selected. You will need a security group to access the instance.

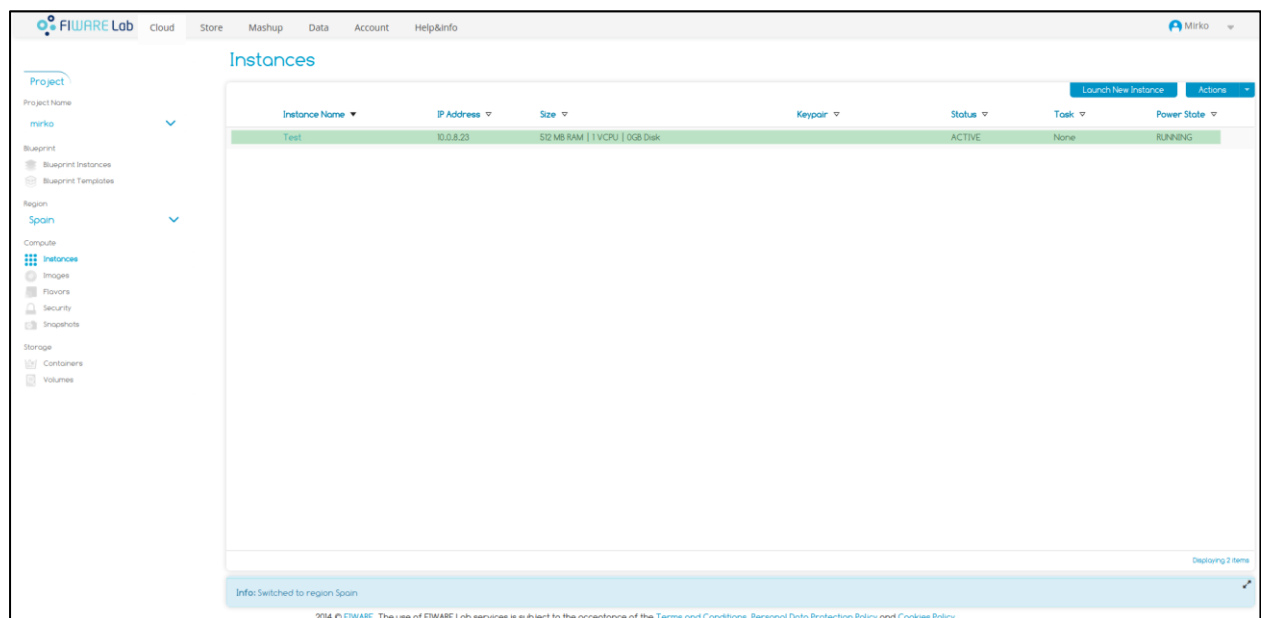
To access the instance:

You need to include a security group with port 22 opened to access via SSH.
 You need to assign a floating IP to access from a external network.

* Mandatory fields.
Back
Launch instance

Figure 41: Deploy a PaaS (Summary)

If everything will be successful you will see a new instance in the instances tab Figure 42



The screenshot shows the 'Instances' tab in the FIWARE Lab Cloud interface. The interface includes a navigation sidebar on the left with options like Project, Region, Compute, and Storage. The main content area displays a table of instances.

Instance Name	IP Address	Size	Keypair	Status	Task	Power State
Test	10.0.8.23	512 MB RAM 1 VCPU 10GB Disk		ACTIVE	None	RUNNING

At the bottom of the interface, there is a status bar indicating 'Deploying 2 items' and a footer with copyright information: '© 2014 FIWARE. The use of FIWARE Lab services is subject to the acceptance of the Terms and Conditions, Personal Data Protection Policy and Cookies Policy.'

Figure 42: List of deployed PaaS

After a scheduled time the DCA server will gather this new instance and you'll see it in your Marketplace.

3.3 Deployment and Configuration Adapter - DCA

The Deployment and Configuration Adapter (DCA) is the XIFI component that caters for the enhanced deployment functionality, as needed by the project users forming in parallel a Deployment Registry. Briefly the DCA provides

- Deployment of multiple GEs and XIFI components upon XiFi infrastructure. The DCA supports the deployment and configuration of multiple GEs in a batch mode (as images, through recipes or in a combination), allowing the user to select details (including the sequential or parallel deployment and the notification capabilities). Such multi-GE deployment can take place in a single node or upon federated XIFI nodes. The DCA can also be used to deploy XIFI components upon the infrastructure.
- Check of Available Resources prior to the Deployment. The DCA performs check on the resources that are available to the user, prior to the deployment of one or multiple GEs and according to the documented hardware requirements of the GEs. This functionality can protect the user from receiving errors (by the platform) after invoking the deployment procedure. The resource availability check is performed considering the user's quota upon the XIFI infrastructure, the resources that have been already reserved by the user and the hardware needs of the GEs under deployment (currently quantified in CPU cores, memory and storage). The checks can be performed per node and / or federated nodes.
- Persistency of information related to the deployed GE instances The DCA holds all pertinent information from the whole lifecycle of GE deployment. This includes the requests on behalf of the users (through the portal) and the system responses as related to the GE instances (going well beyond the typical awareness of the VM instances).

In Figure 43: DCA API example the DCA API to gather the "List SaaS per region": The response is in json format and it is an array of JSON object. Each object describes a SaaS.


```

Request
1 | GET /dca/saas/region/nodeOne HTTP/1.1

Response
1 | 200 (OK)
2 | Content-Type: application/json

1 | [
2 |   {
3 |     "nid": "155",
4 |     "uuid": "90d4865d-5e7b-4d95-af2c-69753e1740d6",
5 |     "available": "true",
6 |     "name": "service_name",
7 |     "description": "service description",
8 |     "imgURL": "HTTP://mysite.com/photos/phot1.jpg",
9 |     "region": "nodeOne",
10 |    "endpoint": "http://121.121.121.121",
11 |    "policy_type": "free"
12 |   }
13 | ]

```

Figure 43: DCA API example

A more detailed documentation could be find at [63].

WStore GE uses these API to communicate with the Cloud Portal, in Figure 44 is shown schema about the interaction between WStore and Cloud Portal through the DCA API.

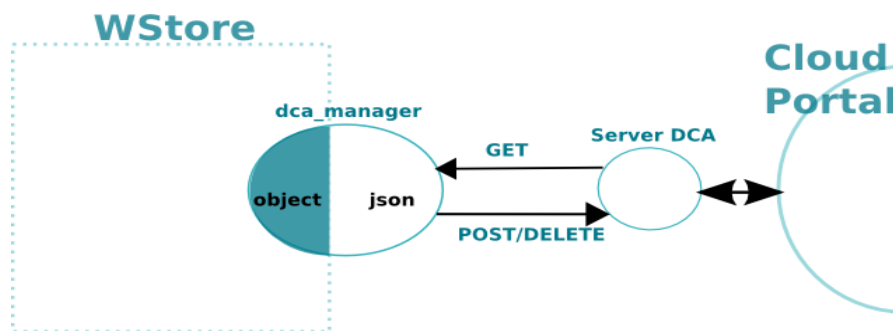


Figure 44: Connection between WStore and CloudPortal across DCA API

A python class (`dca_manager`) has been developed to manage the DCA API.

Through the `dca_manager` the WStore is able to recover data from the Cloud Portal (GET operations) and can modify data in the Cloud Portal (POST/DELETE operations). Another task of the `dca_manager` is to convert the JSON message (got from the DCA API) to a WStore class. With this module integrates WStore and DCA API together, allowing high scalability.

To execute any operation we must add a special token (`access_token`) in the request headers, without this token our HTTP request will be rejected. Before any HTTP operation the `dca_manager` recover, locally or remotely, a valid access token as depicted in Figure 45:

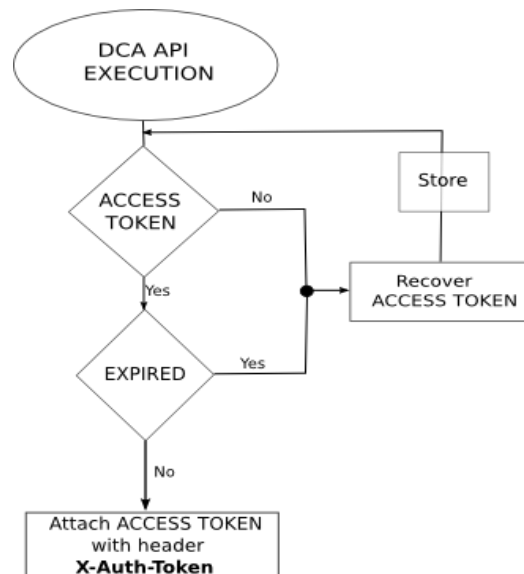


Figure 45: DCA API flow chart

The access token is recovered with the following request:

```

POST: http://cloud.lab.fi-ware.org:4730/v2.0/tokens HTTP/1.1
Content-Type: application/json
{"auth": {"passwordCredentials": {"username":"*****", "password":"*****"}}}
  
```

The returned valid token is cached locally until it expires.

3.4 Federation Manager

Federation Monitoring aims at providing a common framework for storing, aggregating and publishing the monitored data collected by the different monitoring adapters provided by the XIMM module. This component is distributed on all the nodes of the federation and elaborates monitoring data leveraging on big data analysis techniques. This component implements the monitoring functionality at the federation level, providing a common view and data model of the monitoring data. This component elaborates the raw data collected by the individual monitoring adapters, provide historical support and persistence and offer some aggregation functions on the data [23].

SLA Manager - Federation Monitoring Integration: For each template creation user needs to select a service that is whether a PaaS or SaaS service. regarding the role of the user(if has IO role) he can create PaaS services that are retrieved through FM API (by specifying a node name like Trento, Berlin etc.) as listed below:

<http://193.205.211.69:1026/monitoring/regions/Trento/hosts> This is the GET request to retrieve the hosts under Trento node in Federation Monitoring component.

```

200 (OK)
Content-Type: application/hal+json
{
  
```

```
"_links": {
  "self": { "href": "/monitoring/regions/{regionid}/hosts" }
},
"hosts": [
  {
    "_links" : {
      "self": { "href": "/monitoring/regions/Trento/hosts/12345" }
    },
    "id": "12345"
  }
]
```

3.5 Federation Monitoring API

Federation Monitoring aims to provide a common framework for storing, aggregating and publishing the monitored data collected by the different monitoring adapters provided by the XIMM module [23]. This component is distributed on all the nodes of the federation and elaborates raw monitoring data collected by the individual monitoring adapters leveraging on big data analysis techniques, in order to provide historical support and persistence and offer some aggregation functions on the data.

Concerning the user manual of the Federation Monitoring API, a complete description can be found here [24]. According to that, a service in XIFI Marketplace has been implemented ('fmonitoring_service.py') for collecting data is used by the infrastructure recommendation tool. Below an example for retrieving monitoring data from a particular region.

```

Request
1 | GET /monitoring/regions/{regionid}{?since} HTTP/1.1

Response
1 | 200 (OK)
2 | Content-Type: application/hal+json

1 | {
2 |   "_links": {
3 |     "self": { "href": "/monitoring/regions/Trento" },
4 |     "hosts": { "href": "/monitoring/regions/Trento/hosts" }
5 |   },
6 |   "id": "Trento",
7 |   "name": "Trento",
8 |   "country": "Italy",
9 |   "latitude": "xyz",
10 |  "longitude": "xyz",
11 |  "measures": [
12 |    {
13 |      "timestamp" : "2013-12-20 12.00",
14 |      "nb_cores": "100",
15 |      "nb_cores_enabled": "100",
16 |      "nb_ram": "1000",
17 |      "nb_disk": "10000",
18 |      "nb_vm": "100000",
19 |      "power_consumption": "123",
20 |      "percCPULoad": {
21 |        "value": "123",
22 |        "description": "average of the percCPULoad for all the hosts"
23 |      },
24 |      "percRAMUsed": {
25 |        "value": "123",
26 |        "description": "average of the percCPULoad for all the hosts"
27 |      },
28 |      "percDiskUsed": {
29 |        "value": "123",
30 |        "description": "average of the percCPULoad for all the hosts"
31 |      }
32 |    }
33 |  ]
34 | }

```

Figure 46: Federation Monitoring API: particular region data request and response

3.6 FIWARE Catalogue

The FIWARE Catalogue is the central repository for implementations of Generic Enablers (GE) that are part of the FIWARE platform. FIWARE Catalogue provides a set of API's for the 3rd party users that helps the integration of GE's in their applications. FIWARE Catalogue is integrated directly with Resource Catalogue, enabling to retrieve the list of GEs, show them to the user, search with keywords, manipulate the retrieved data and present it in the Resource Catalogue.



Home Enablers Bundles Tools Forum Login / Register FIWARE Catalogue

Hosting enablers for creation of FUTURE INTERNET APPLICATIONS

Welcome to the FIWARE Catalogue! Here you will find all the information, documentation and tools you need as a developer to start using a Generic Enabler Implementation.

About the Catalogue

The FIWARE Catalogue is the central repository for implementations of Generic Enablers (GE) that are part of the FIWARE platform. Apart from the Generic Enablers, you will also find tools and best practices which will help you develop the applications of the Future Internet.

View the Enablers

No registration is necessary, simply start browsing the list of Generic Enabler Implementations to see for yourself what the FIWARE platform offers.

View Generic Enabler Implementations

Tools

FIWARE offers tailor-made tools for developing applications for the Future Internet. Whether it be custom Eclipse plug-ins, software testing suites or guidelines and best practices, odds are you will find what you need.

See the Tools

Publishing a Generic Enabler

Anyone is free to create an implementation of a Generic Enabler and publish it on the Catalogue. Look at our video overview or read through our publication process to get you started.

Read about the publication process

The FIWARE Platform

FIWARE will deliver a novel service infrastructure, building upon elements (called Generic Enablers) which offer reusable and commonly shared functions making it easier to develop Future Internet Applications in multiple sectors – building a true foundation for the Future Internet.

Read more about FIWARE

Future Internet PPP | Main FI-WARE site | Publication process

Figure 47: FIWARE Catalogue GUI website

The list of API methods that are used in Resource Catalogue are listed below (urls in the example requests uses the IP address of test server for for FIWARE Catalogue):

- http://130.206.80.235/rest_node_api/enabler_list.json: This API call returns the list of all GE's as a JSON format.
- http://130.206.80.235/rest_node_api/search_node/retrieve.json?keys=XX%20type:enabler: This API call makes the search request with a given word(replaced by 'XX' key) towards the FIWARE Catalogue. Response for this search request with a 'cloud' word is listed below:
- Recommendation Tools

Two different recommendation tools have been integrated in XIFI Marketplace: one for recommending infrastructures, and the other one for recommending policies. Each recommendation tool is described below.

- *Infrastructure recommender:*

The infrastructure recommendation tool aims at supporting users by informing the XIFI federated node that fits the best according to the particular application requirements. The recommendation tool is mainly based on Federated Monitoring data (Physical Hosts and Network data). In Figure 39 the recommendation tool model is shown.

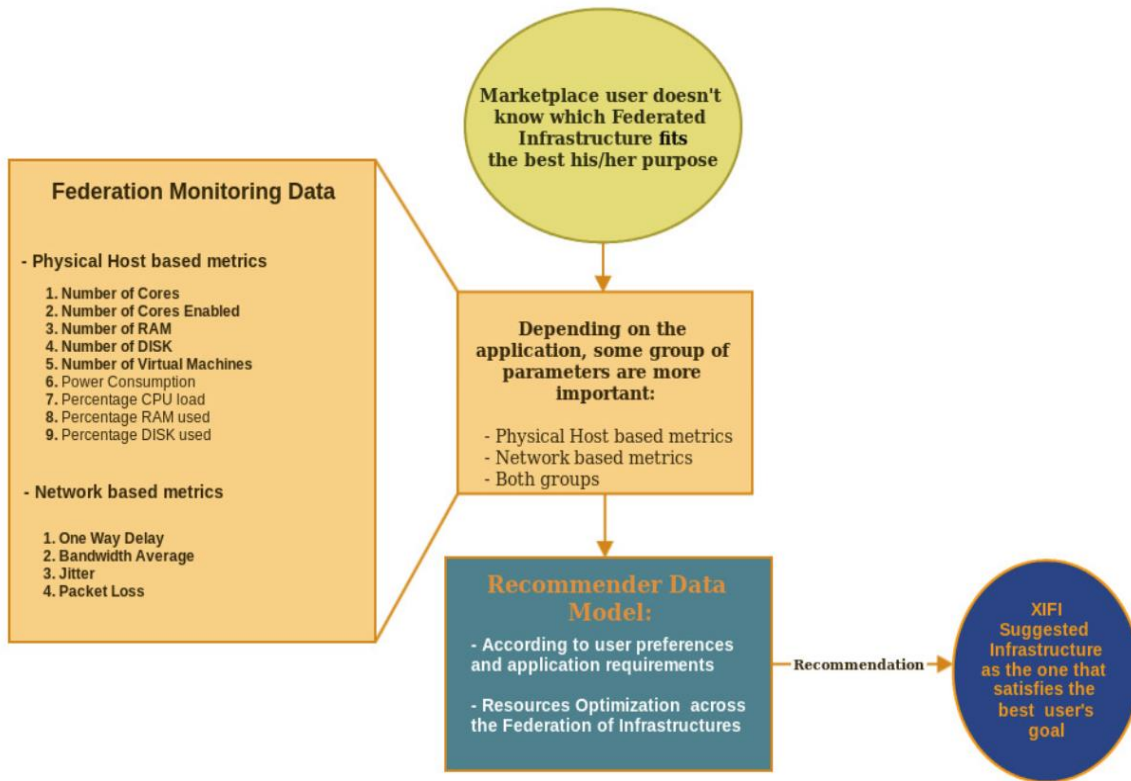


Figure 48: Recommendation Tool model

As it can be extracted there are two main groups of metrics that can be gathered from the Federation Monitoring API module:

- Physical Host based: related to computational resources in the regions
- Network based: related to network capabilities for transmitting data

The Recommendation Tool is integrated in the Resource Catalogue, and supports two kinds of XIFI users:

- **Developers:** in this case, a future internet developer wants to make use of a public instance or launch a particular one. However, the developer may not know which available instance or infrastructure fits the best to the requirements. At this point, the recommendation tool asks the developer to weight a set of parameters related to the application requirements. Based on this and up-to-date monitoring data from the Federation Monitoring API, the tool can identify the infrastructure or the infrastructures with the particular instance that satisfies all the requirements and fits the best to it. The result provided by the tool is a sorted list of infrastructures.
- **Service Providers:** in this case, a service provider wants to offer a new service, but it is not aware of the infrastructure that fits the best service requirements. Then, user is asked to weights a set of parameters that are used by the recommendation tool, which also gathers data from the Federation Monitoring API. The result provided by the tool is a sorted list of infrastructures.

Furthermore, for recommending the infrastructure that fits the best, the Recommendation Tool optimizes resources across the XIFI Federation. To sum up, the Recommendation Tool suggests the

XIFI facility that satisfies the best user's goal, and also avoids wasting federation resources.

The recommendation tool is available from XIFI Marketplace and Resource Catalogue as an API, where weighted parameters gathered from the user has JSON format.

- *Policy recommender:*

The XIFI federation brings together computational infrastructures (e.g. computational resources, networking resources, and storage resources) in order that they can be leveraged in combination by the developers of Future Internet technology. The central objective is to support industry stakeholders who require Future Internet resources to perform large-scale trials in order to evaluate and validate their technologies before they are transferred to market.

The technical compliance has been addressed by the interoperability tool in XIFI; however, the legal and ethical compliance is still an issue for experimenters and application providers mainly when their applications process data that is private (personal) or commercially sensitive. In such instances, the experimenter would currently be obliged to consider a hybrid cloud, with the private data used only within their private cloud. This may not, however, be practical or even desirable all the time and defeats the object of federated XIFI resource.

Before deploying their applications or their data onto the federation infrastructure, the experimenters and application providers will need to understand the different applicable privacy policies.

Therefore, the policy recommendation service allows experimenters to find suitable infrastructures within the federation that fulfill their specific privacy preferences. The service thus allows the identification of the most appropriate set of infrastructures for a concrete experiment. XIFI Marketplace and Resource Catalogue use the privacy policy management tool (in conjunction with other XIFI tools) to allow experimenters/application providers to select the most suitable combination of infrastructures according to their privacy requirements following a guided process of recommendations.

This service requires the nodes' privacy policies to be available in machine readable format that can express a sophisticated set of terms and conditions associated with clear semantics for consistent processing.

The recommendation service can be applied in the following scenarios:

- Privacy policy be associated with the whole node
- Privacy policy is associated with a specific service offered by the node.

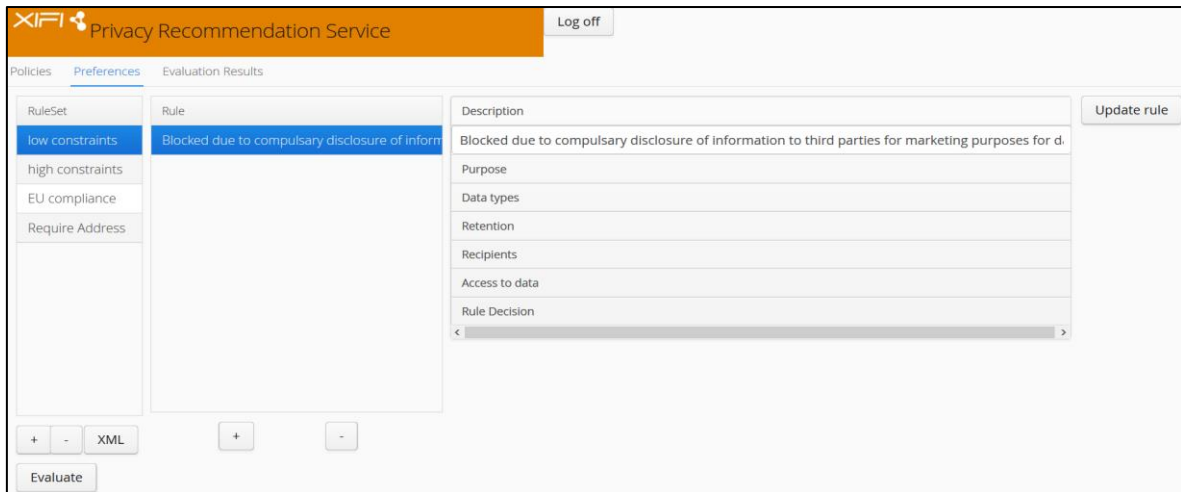


Figure 49: Policy Recommender: policies view

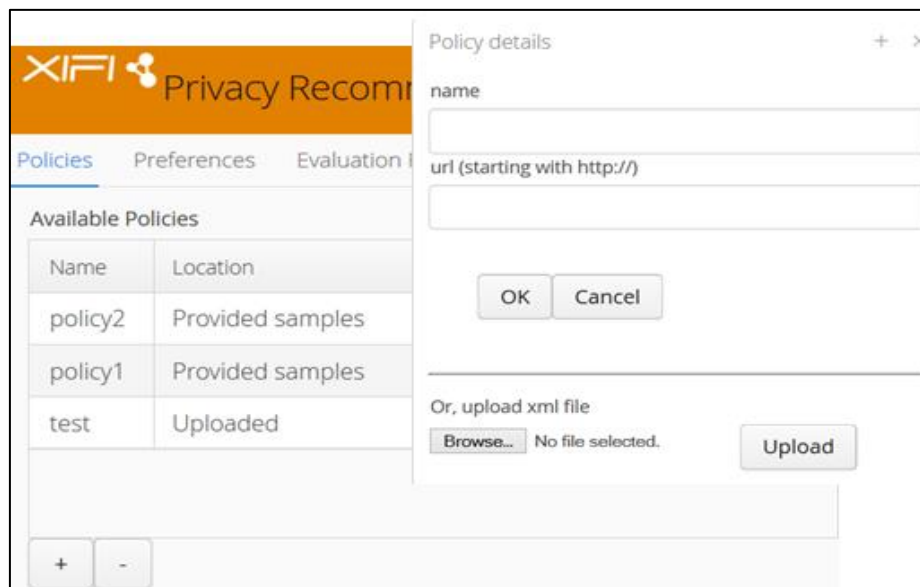


Figure 50: Policy Recommender: preferences layout

3.7 Marketplace and Resource Catalogue

XIFI Marketplace portal objective is to become a single entry point for accessing to services offered by the federation of the different facilities involved in the XIFI project. It offers a set of cloud community services, which shall be accessible through a graphical representation (a Graphical User Interface - GUI) supported by this portal.

Resource Catalogue provides a catalogue with all available services offered by the federation, tailored by federated nodes. Thus Future Internet Developers are able to browse, search and compare services and also to access to services details, and moreover, Infrastructure Owners are able to publish new services offered by his/her own node.

The marketplace homepage Figure 51 is composed by two main parts: on the left there is the map of XIFI node and on the right there are the PaaS services table and SaaS services table.

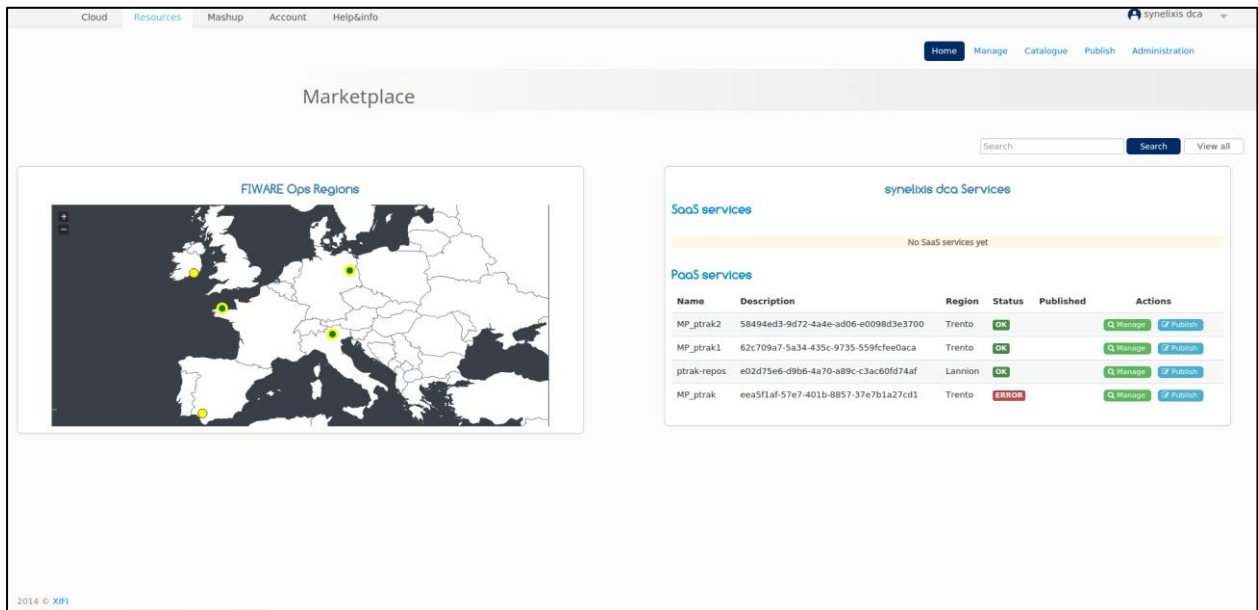


Figure 51: Marketplace homepage

The PaaS table Figure 52 will contain the user’s PaaS images (created in the Cloud Portal [36]).

The table shows (for each PaaS image):

- name: the name of image
- description: a description
- node: the node where it has been deployed
- status: the state of the instance (OK, ERROR, DELETED)
- published: if the PaaS images has been published as SaaS
- actions : the actions that an user can be apply on the PaaS image

The actions will be different depending on the user roles.

Name	Description	Region	Status	Published	Actions
MP_ptrak2	58494ed3-9d72-4a4e-ad06-e0098d3e3700	Trento	OK		Q Manage Publish
MP_ptrak1	62c709a7-5a34-435c-9735-559fcfee0aca	Trento	OK		Q Manage Publish
ptrak-repos	e02d75e6-d9b6-4a70-a89c-c3ac60fd74af	Lannion	OK		Q Manage Publish
MP_ptrak	eea5f1af-57e7-401b-8857-37e7b1a27cd1	Trento	ERROR		Q Manage Publish

Figure 52: PaaS services table

In the following images is shown the details of a PaaS image.

Name	Description	Region	Status	Published	Actions
MP_ptrak2	58494ed3-9d72-4a4e-ad06-e0098d3e3700	Trento	OK		Manage Publish

Figure 53: A PaaS service

A user with the Service Provider role can manage the PaaS image (through the Cloud Portal)

With the “Manage” button Figure 52 the user will be redirected to the cloud portal where he can manage the PaaS instance Figure 53. These actions are not permitted to the Developer role.

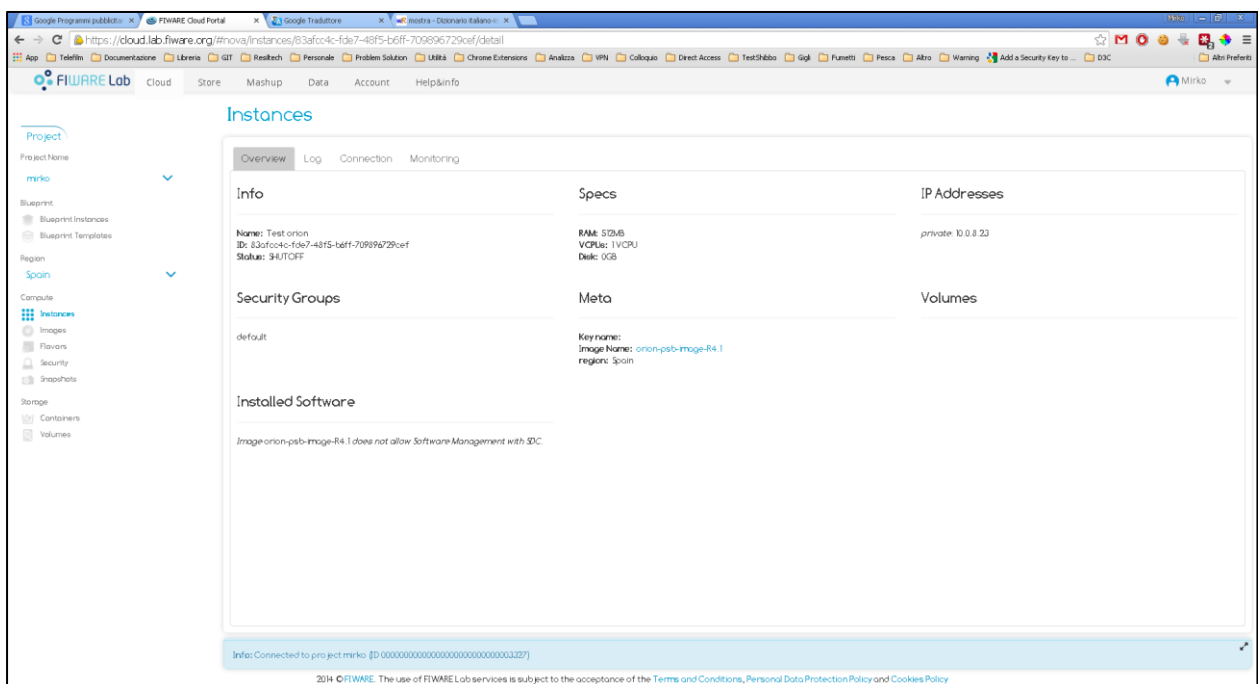


Figure 54: PaaS service page in Cloud Portal

The “Publish” button Figure 52 allow to publish a PaaS service as SaaS service (by a specific DCA API). This information will be stored locally in the Marketplace, this action is allowed to both roles (Devolper and Service Provider).

When a PaaS is published as SaaS it is shown in the column “Published” and the “Publish” button is disabled Figure 54.

PaaS services						
Name	Description	Region	Status	Published	Actions	
MP_ptrak2	58494ed3-9d72-4a4e-ad06-e0098d3e3700	Trento	OK	Yes	Q Manage	Publish
MP_ptrak1	62c709a7-5a34-435c-9735-559fcfee0aca	Trento	OK		Q Manage	Publish
ptrak-repos	e02d75e6-d9b6-4a70-a89c-c3ac60fd74af	Lannion	OK		Q Manage	Publish
MP_ptrak	eea5f1af-57e7-401b-8857-37e7b1a27cd1	Trento	ERROR		Q Manage	Publish

Figure 55: PaaS services table after a PaaS published as SaaS

As we can see in Figure 55 this SaaS (published=yes) is eligible to be registered by a user. The SaaS table Figure 56 will show the services registered by the user, the registration is made in the Resource Catalogue.

SaaS services			
Name	Description	Region	Actions
MP_ptrak2		Trento	Unregister

Figure 56: SaaS services table

Through this table a user can unregister a SaaS service and make it available to others.

The screenshot displays the 'Generic Enablers' section of the XIFI Marketplace. The page is organized into a grid of cards, each representing a different generic enabler. The left sidebar provides navigation and filtering options, including 'Items per page' (set to 8), 'Browse By Type' (Generic Enablers, Other Services, Specific Enablers), 'Browse by Chapter' (All), and 'Filter by Node' (All). The top navigation bar includes links for Home, Manage, Catalogue, Publish, and Administration. The main content area shows the following generic enablers:

- Access Control - THA Implementation:** Administration & Enforcement of RESTful API Authorization Policy. Rating: 5 stars, 0 votes.
- API mediation:** The API mediation component deals with publishing, discovering and exposing APIs to stakeholders as well as managing some non functional aspects such as provisioning and monitoring. Rating: 5 stars, 0 votes.
- BigData Analysis - Cosmos:** Monitoring and control of the BigData Analysis GE. Rating: 5 stars, 0 votes.
- Cloud Edge:** Sort of "Super Gateway", located at the edge between the WAN/Cloud and the LAN and able to locally execute applications. Rating: 5 stars, 0 votes.
- Complex Event Processing (CEP) - IBM Proactive Technology Online:** Complex Event Processing GE. Rating: 5 stars, 0 votes.
- Compressed Domain Video Analysis - Codoan:** Provides a set of tools for analyzing video streams in the compressed domain. Rating: 5 stars, 0 votes.
- Configuration Manager - Orion Context Broker:** Orion Context Broker is an implementation of NGSI9 and NGSI10 with persistence storage based in MongoDB. Rating: 5 stars, 0 votes.
- Data Handling - PPL:** The Data Handler GE is a privacy-friendly attribute-based access control and usage control system to (sensitive) data. Rating: 5 stars, 0 votes.

At the bottom of the grid, there is a pagination control showing page 1 of 5.

Figure 57: GE list

4 GENERIC ENABLERS

Generic Enablers are divided into 6 sub-chapters and listed in the combo-box menu:

- All
- Services Ecosystems and Delivery Framework
- Cloud-hosting
- Context Management
- Interface to Networks and Devices
- Internet of Things Services Enablement
- Security

Each Generic Enabler is deployed as a different instance on the nodes (Berlin, Trento, Waterford etc.). Each instance of GE is shown on the node that is deployed and running

This map has information about where the GE has been deployed (nodes) and how many instances do exist in every node.

User can apply filters in order to view the resources; moreover can set other parameters such as:

- configuring the number of items per page
- viewing sub-chapters (we can see that subchapter:Services Ecosystems and Delivery Framework is selected)

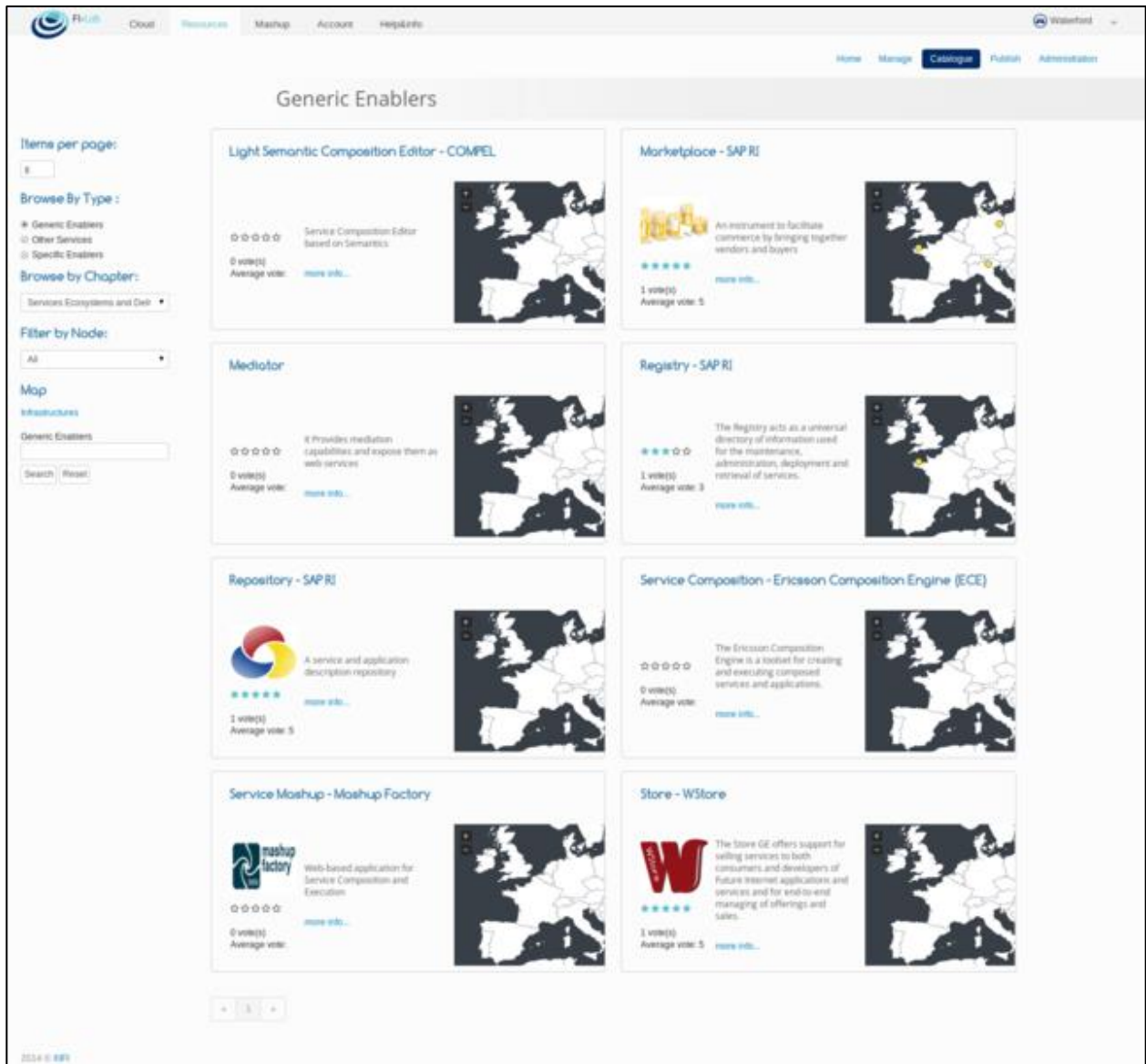


Figure 58: GE list filtered by sub chapter

- Searching with the specified keywords.(Below an example of search with the keyword "Event", all the GE's contain that word listed after the search)

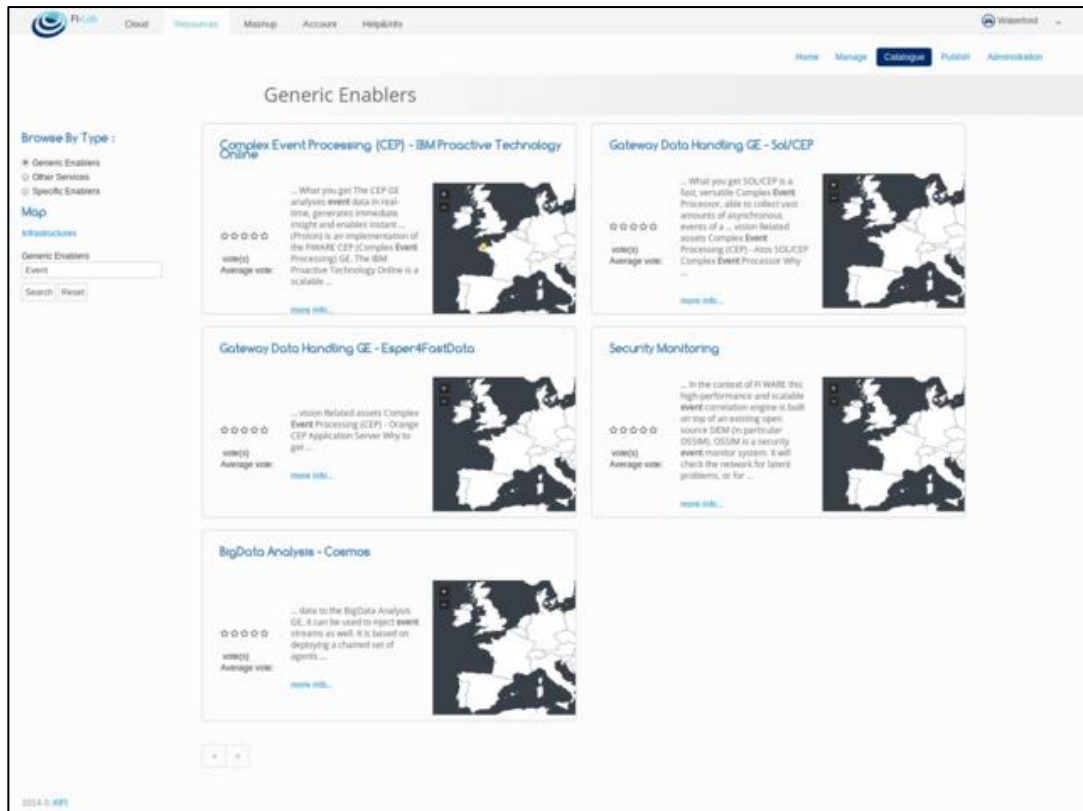


Figure 59: GE Search result

- Selecting node name in order to view the GE's that are deployed in that node (We can see that Trento node is selected in the combo-box and 2 of the GE's that are currently deployed to Trento node is listed with the number of instance)

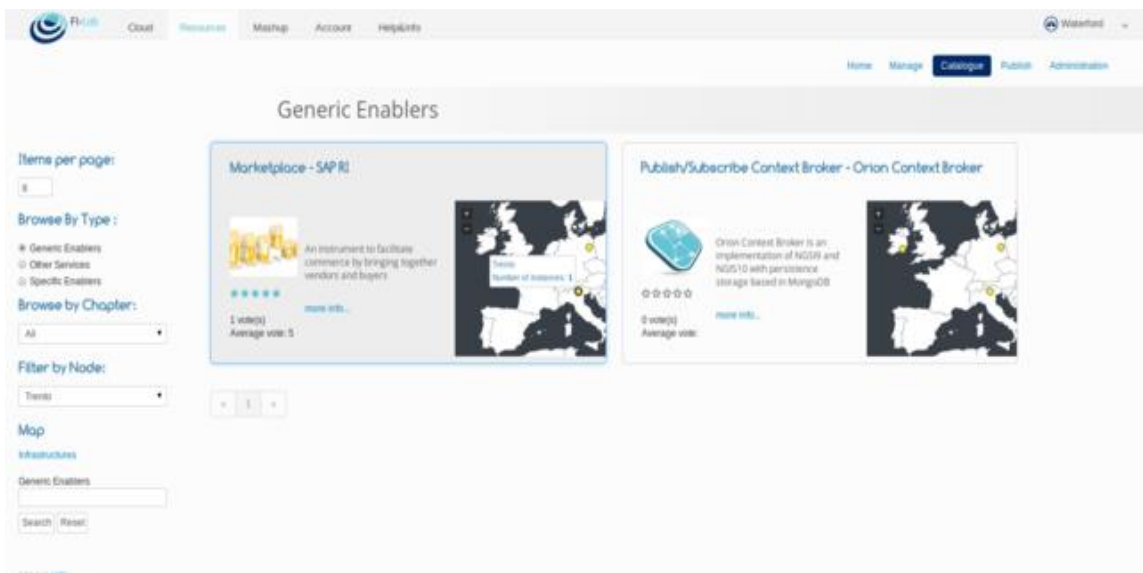


Figure 60: GE's filtered by node name

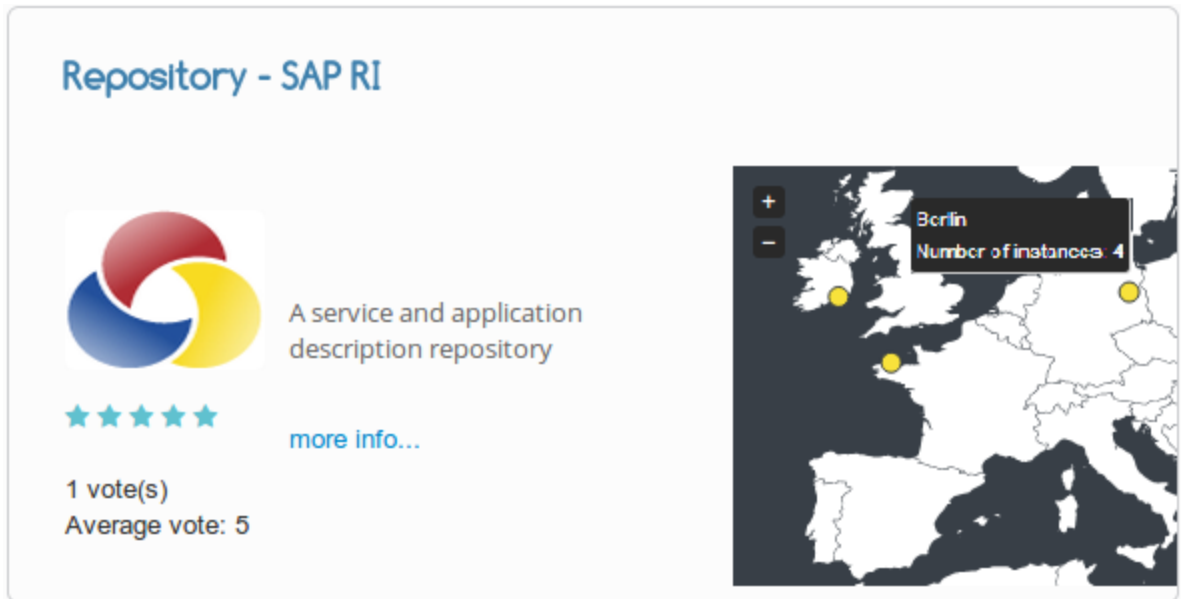


Figure 61: Sample GE

- User can click the Map link on the left panel to see the details of the all nodes with the information about GE, SE and NCS's. We show the number of the instances of SE, GE and NCS that are deployed for each node.

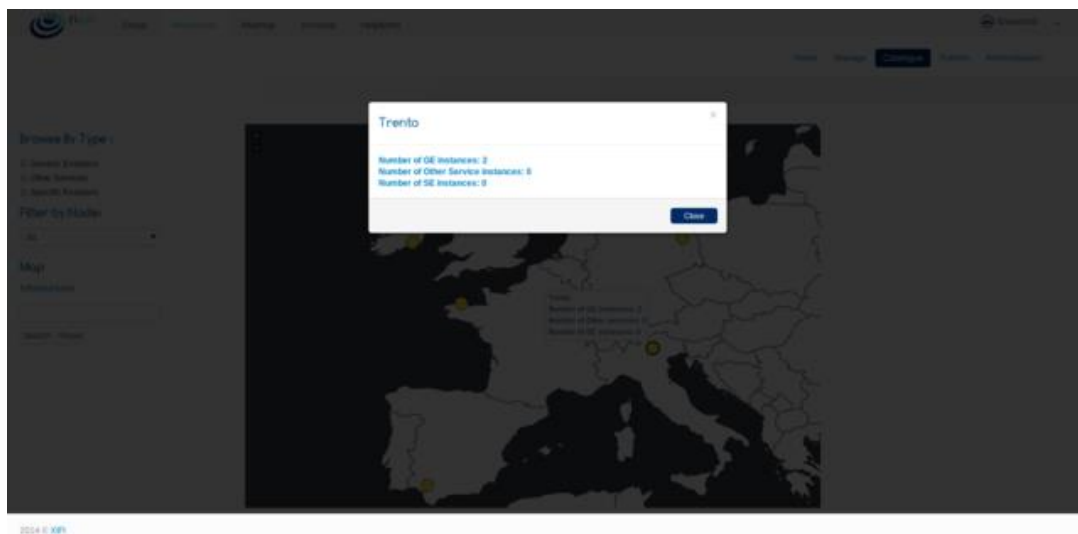


Figure 62: All-node information with GE, NCS and SE

- User can click on the link to see the details of the Generic Enabler

This section is the place where user can see more information about the GE, for example:

- Logo and voting info about the GE
- In which nodes this GE is deployed
- Comments from users
- Terms and conditions

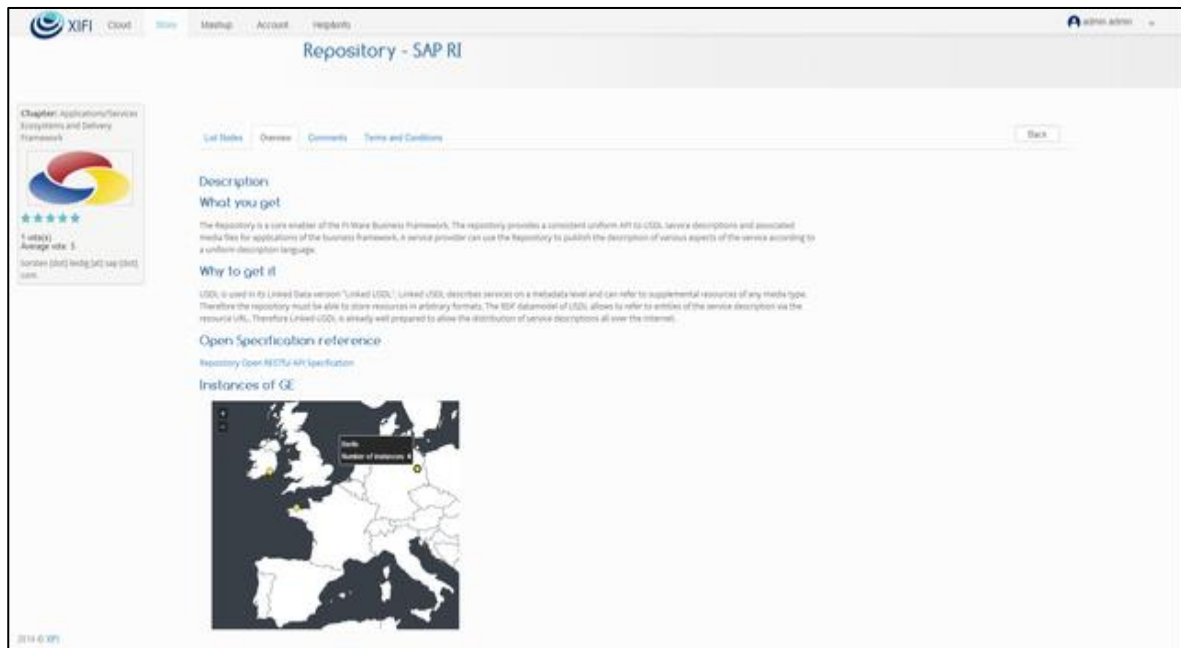


Figure 63: GE detail page

5 NON CONVENTIONAL SERVICES AND SPECIFIC SERVICES

It is possible to register two types of services: i) Specific Enablers and ii) Advanced/Non-Conventional Services. There is an option to specify if this is a SE or NCS, if the user has the rights see this selection and selects this option then it will be created as a SE otherwise it will be a NCS.

- Specific Enabler can be published by Software Providers and Infrastructure Owners.
- Register the image in the Cloud Portal, to be deployed automatically for the stakeholders that want to use and monitoring.
- Advanced/non-conventional capabilities (e.g. sensor networks) can be published by the Infrastructure Owners. These don't need to be registered in the Cloud Portal, since it is only necessary to provide their properties and characteristics. This allow to obtain the contact information of the infrastructure owner in order to ask him or negotiate with him the offer details.
- In both case, register the service description through the USDL
- User can select an USDL file and register the services also, here is an example of a service that is registered by using an USDL file that is provided by Orange company. Here are the 2 screenshots related with this action.

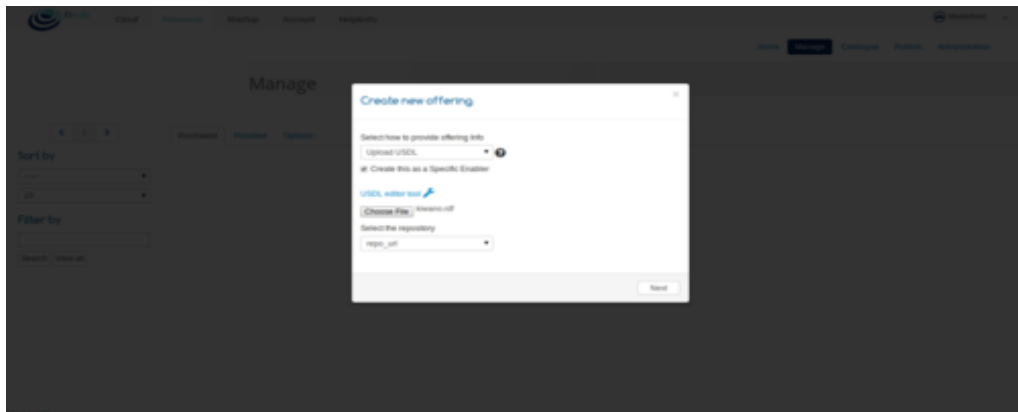


Figure 64: Resource creation step 1

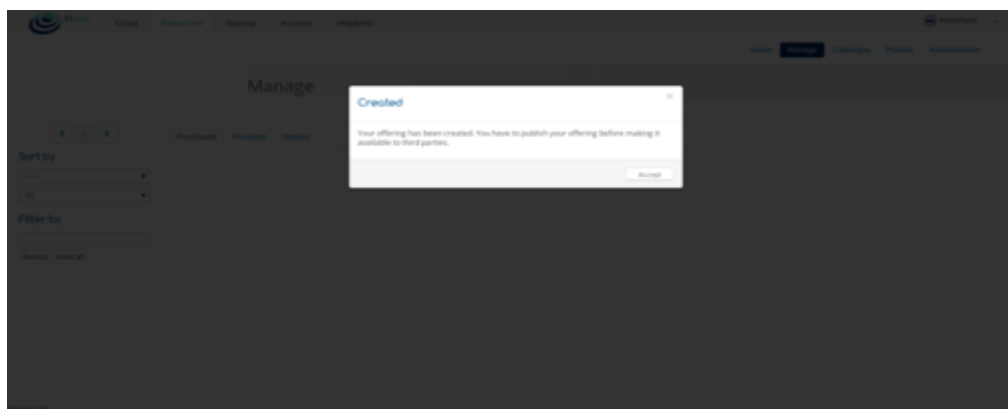


Figure 65: Resource created

- User are not forced to provide USDL files all the time, they can be created by clicking "Create

basic USDL" option in the combo-box and provide information related with the USDL content

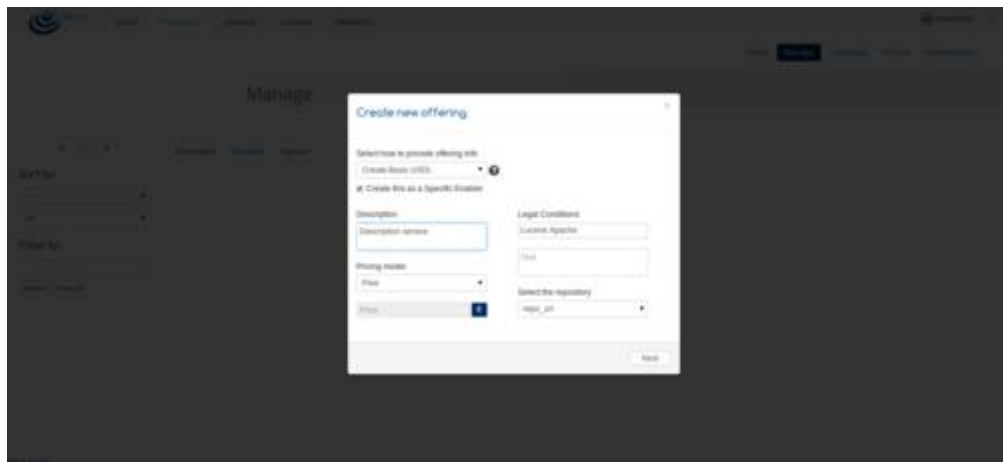
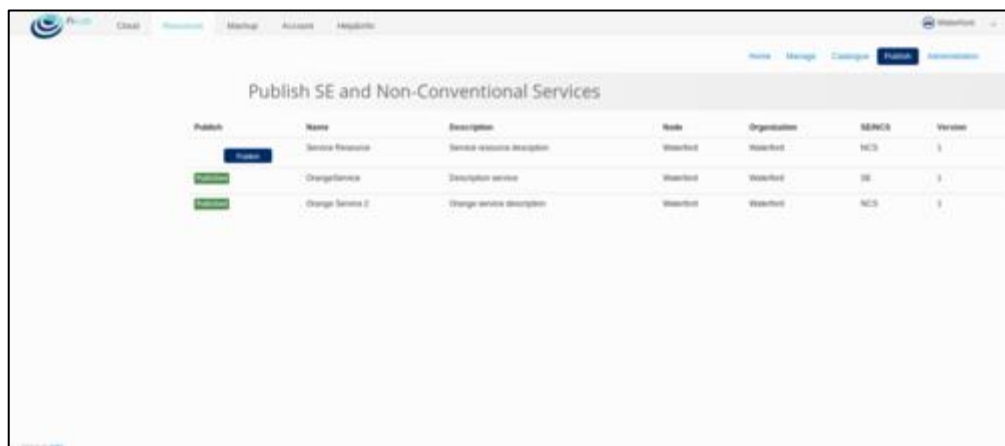


Figure 66: Resource creation, USDL selection part

- Validate the services. Resources needed to be published/validated before going live
- The federator role needs to validate that all the content are correct, before to confirm the publication in the catalogue.



Publish	Name	Description	Role	Organisation	SE/NC	Version
<input type="button" value="Publish"/>	Service Resource	Service resource description	Workflow	Workflow	SE	1
<input type="button" value="Publish"/>	ChangeService	Description service	Workflow	Workflow	SE	1
<input type="button" value="Publish"/>	Change Service 2	Change service description	Workflow	Workflow	NC	1

Figure 67: Resource publish page

- After the Federator has approved them, the services will be available to find and see the status in the XIFI Portal.

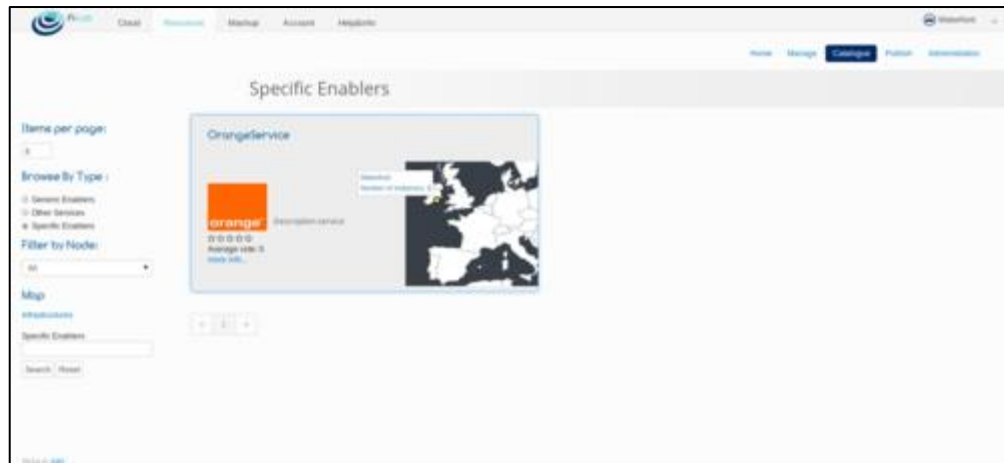


Figure 68: SE list

Afterwards, the FI Developers can find and use these services.

- List of Non-Conventional Services

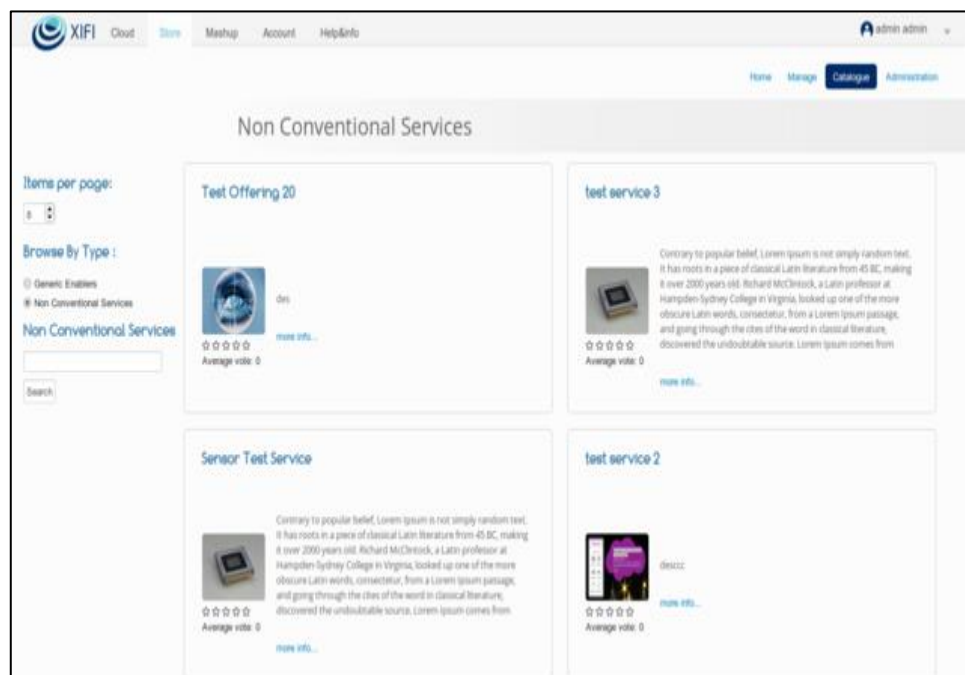


Figure 69: NCS list

- User can search through Non-Conventional Services by the given keyword:

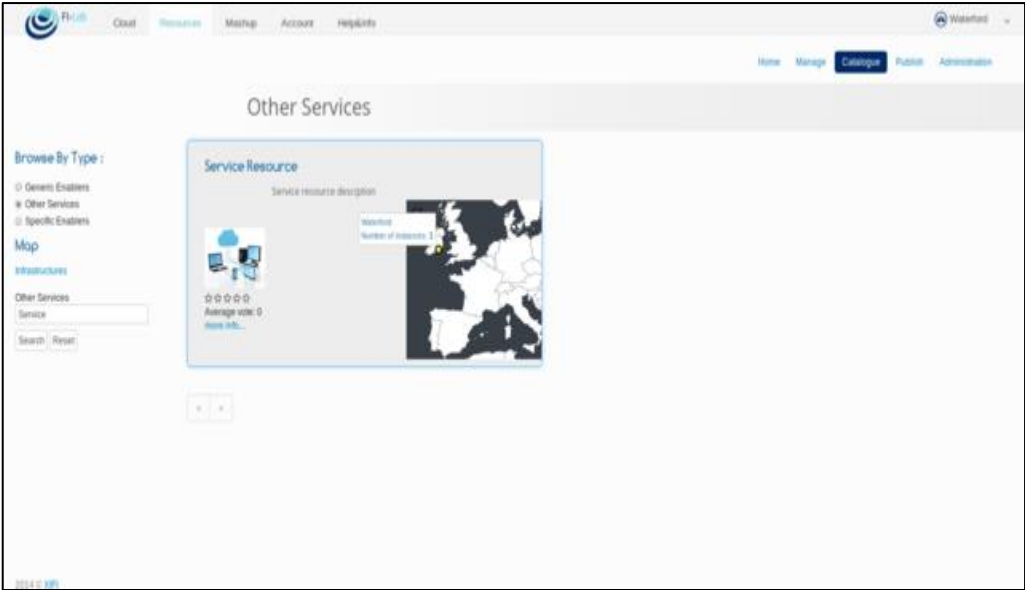


Figure 70: NCS search

6 CONCLUSIONS

This deliverable provides the second software version of XIFI Marketplace Portal described in the framework of WP4. XIFI Portal is conceived as a single entry point for accessing to services offered by the federation. It provides a sort of “glue” for integrating and aggregating the rest of the tools provided by WP4.

This document describes XIFI Portal itself and its relation to the following services:

- Resource Catalogue and Recommendation Tool
- Monitoring Dashboard
- SLA and Accounting Dashboard
- Security and Privacy Dashboard
- Interoperability Tools
- Federated Identity Management
- Infographics and Status Pages
- Cloud Portal
- Federation Manager GUI

In addition, this document provides a complete description of the modules that have been implemented concerning XIFI Marketplace aiming to provide the required functionality of the Marketplace.

Finally, the document provides an updated description of the policy recommendation service in the annex section, in order to describe the current status of the service that has been integrated in XIFI Marketplace.

Moreover, this document has gathered information from the XIFI wiki, which is the project collaborative tool, and the description of a component is not static and its specifications will be evolved and provided up to data through the wiki. During next months, component description will evolve to their final versions and can be used as reference manual for the showcases (WP6), training materials (WP7), and business approach (WP8). Many of these components have a direct relationship with the end user, so keeping updated content will be necessary in order to collaborate transversely (cross-WP).

The next steps will focus on keeping with the integration of the different component in the FIWARE Lab and FIWARE-Ops portals, providing a common web header for accessing to them.

REFERENCES

- [1] Subversion: <https://xifisvn.esl.eng.it>
- [2] D4.1b- Services and tools specification: <https://www.fi-xifi.eu/publications/deliverables.html>
- [3] D4.2- Baseline Tools v1: <https://www.fi-xifi.eu/publications/deliverables.html>
- [4] D4.4- Baseline Tools v2: <https://www.fi-xifi.eu/publications/deliverables.html>
- [5] D4.3- Marketplace implementation v2: <https://www.fi-xifi.eu/publications/deliverables.html>
- [6] Federated Identity Management component: https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Identity_Management_Generic_Enabler
- [7] FIWARE Lab : <http://lab.fi-ware.eu>
- [8] FIWARE project: <http://www.fi-ware.eu/>
- [9] FIWARE-Ops: <http://www.fi-ware.org/fiware-operations/>
- [10] XIFI – wiki: http://wiki.fi-xifi.eu/Public:Software_Components
- [11] XIFI Stakeholders: <https://www.fi-xifi.eu/about-xifi/stakeholders.html>
- [12] Cloud Portal: http://wiki.fi-xifi.eu/Public:Cloud_Portal
- [13] Federated Identity Management: http://wiki.fi-xifi.eu/Public:Federated_Identity_Management
- [14] FIWARE IdM GE: <https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.Security.IdentityManagement>
- [15] Monitoring Dashboard component: http://wiki.fi-xifi.eu/Public:Federation_Monitoring
- [16] SLA and Accounting Dashboard: http://wiki.fi-xifi.eu/Public:SLA_Manager
- [17] WS-Agreement specification: <http://ogf.org/documents/GFD.192.pdf>
- [18] Security and Privacy Dashboard: http://wiki.fi-xifi.eu/Public:Software_Components
- [19] Interoperability Tools: http://wiki.fi-xifi.eu/Public:Interoperability_Tool
- [20] Infographics and Status Pages: http://wiki.fi-xifi.eu/Public:Infographics_and_Status_Pages
- [21] Federation Manager: http://wiki.fi-xifi.eu/Public:Federation_Manager
- [22] Resource Catalogue and Recommendation Tool: http://wiki.fi-xifi.eu/Public:Resource_Catalogue%26Recommender
- [23] Federation Monitoring component: http://wiki.fi-xifi.eu/Public:Federation_Monitoring
- [24] Federation Monitoring API: <http://docs.federationmonitoring.apiary.io/>
- [25] DCA API: <http://docs.dca.apiary.io>

APPENDIX A: Policy Recommendation Service

Summary

The recommendation service allows experimenters to find suitable infrastructures within the federation that fulfil their specific privacy preferences. The service thus allows the identification of the most appropriate set of infrastructures for a concrete experiment. The Marketplace uses the privacy policy management tool (in conjunction with other XIFI tools) to allow experimenters/application providers to select the most suitable combination of infrastructures according to their privacy requirements following a guided process of recommendations.

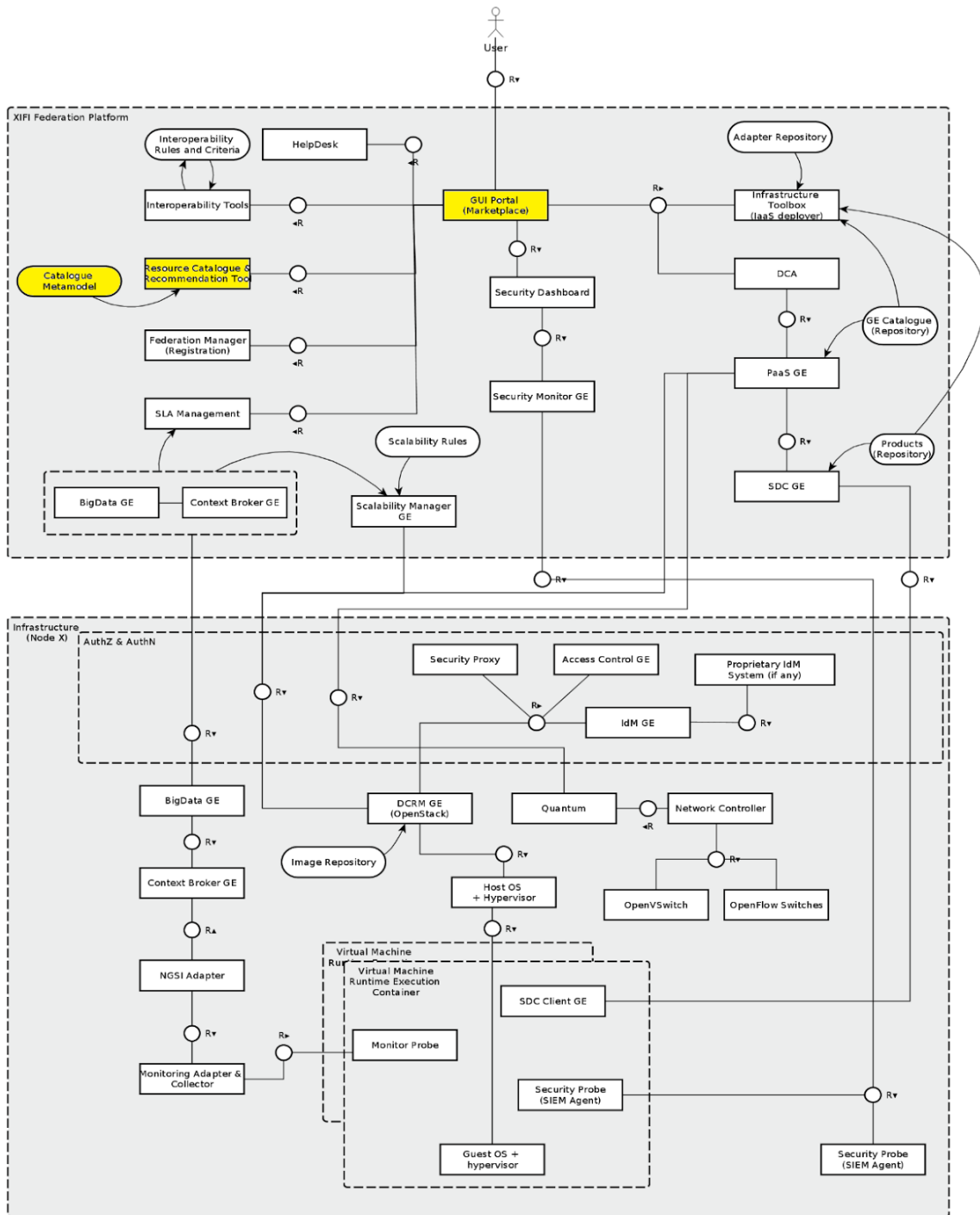


Figure 71: Privacy Recommendation service within the Xifi Federation Platform context

Reference Scenarios	UC2-Setup and use of development environment
Reference Stakeholders	As end user (Developers, End Users/Adopters; Sponsors/ Investors)

Planned OS license	LGPL version 3.0
---------------------------	------------------

A.1.1 Component Leader

Developer	Contact	Partner
Bassem Nasser	bnm@it-innovation.soton.ac.uk	IT-Innovation

A.1.2 Motivation

The XIFI federation brings together computational infrastructures (e.g. computational resources, networking resources, and storage resources) in order that they can be leveraged in combination by the developers of Future Internet technology. The central objective is to support industry stakeholders who require Future Internet resources to perform large-scale trials in order to evaluate and validate their technologies before they are transferred to market.

The technical compliance has been addressed by the interoperability tool in XIFI; however, the legal and ethical compliance is still an issue for experimenters and application providers mainly when their applications process data that is private (personal) or commercially sensitive. In such instances, the experimenter would currently be obliged to consider a hybrid cloud, with the private data used only within their private cloud. This may not, however, be practical or even desirable all the time and defeats the object of federated XIFI resource.

Before deploying their applications or their data onto the federation infrastructure, the experimenters and application providers will need to understand the different applicable privacy policies.

A XIFI privacy recommendation service allows these federation users to choose the nodes that match their privacy requirements.

This service requires the nodes' privacy policies to be available in machine readable format that can express a sophisticated set of terms and conditions associated with clear semantics for consistent processing.

The recommendation service can be applied in both scenarios:

- Privacy policy be associated with the whole node
- Privacy policy is associated with a specific service offered by the node.

A.1.3 User stories

Id	User Story Name	Actors	Description
1	Select privacy policies	Experimenter/ application developer/ application	The actor selects the different privacy policies to analyse. The privacy policies can be those of specific services or whole node.

		provider	
2	Specify privacy preference	Experimenter/ application developer/ application provider	The actor specifies their privacy preferences.
3	Find compatible policies	Experimenter/ application developer/ application provider	Actor runs the policy analysis and obtains a list of compatible policies

A.1.4 State of the art

A.1.4.1 Platform for Privacy Preferences (P3P)

Platform for Privacy Preferences (P3P) is a W3C specification that enables Web sites to express their privacy policy in a machine readable, standard XML format. The goal is to automate the user decision-making process based on the privacy practices of the website. The specification is supposed to be interpreted by user agents installed in the browser (browser plug-ins, or proxy servers) so that the user does not need to read the privacy policy of each site they visit.

In addition to the data-collection and data-usage practices, the specification provides a means of associating privacy policies with Web pages or sites, and cookies and a mechanism for transporting P3P policies over the web.

In the P3P context, any data that can be used reasonably by a data controller or any other person to identify an individual is considered to be identifiable data.

A policy reference file is used to locate the privacy policy documents associated with websites or resources at the same website. The policy reference file may be located in a predefined location or sent across via the html header or in an html “link” tag.

The main content of a privacy policy is the following:

- Policy identifier: this includes the name of the policy, URI of the natural language privacy statement, etc.
- Entity: identifies the legal entity making the representation of the privacy practices contained in the policy.
- Access: refers to the ability of the individual to view identified data and address questions or concerns to the service provider. Service providers must disclose one value for the access attribute.
- Disputes: this refers to one way the entity offers or acknowledges for a user to resolve disputes about the entity's privacy practices or alleged protocol violations (e.g. customer service, independent organization, court or applicable law). It should also include remedies that the entity offers to the identified dispute resolution procedures (e.g. correct the error, fine or compensation, legal redress).

- Policy statement: is a container that encapsulates the following information:
 - Consequence: a short summary or explanation of the data practices described in this policy statement that can be shown to a human user.
 - Purpose: purposes for data processing. This is mandatory when the collected data is “identifiable”.
 - Recipient: lists who is receiving the collected data. This can be the entity itself, legal entities following the same practices, public, unrelated third parties, etc.
 - Retention: the type of retention policy in effect. This can include, for instance, no retention, or retention for the stated purpose, legal requirement, business practices, etc.
 - Data: this includes information about the data to be transferred (e.g. computer information) or inferred (e.g. location information inferred from the IP address). The data types (e.g. business info, IP address, home info, etc) and the categories of the data (e.g. "financial" for Financial Information, "computer" for Computer Information, "demographic" for Demographic and Socioeconomic Data, etc) can also be included here.

A data schema and an XML policy schema are available to be used for defining privacy policies.

A.1.4.2 A P3P Preference Exchange Language (APPEL)

APPEL is a W3C working draft specification that complements P3P and allows users to express their privacy preferences in a set of preference-rules which allows the user agent to make automated or semi-automated decisions regarding the acceptability of machine-readable privacy policies from P3P enabled Web sites.

The preferences are encapsulated in a “ruleset” which includes a series of rules. Each rule contains conditions under which a behaviour (e.g. block, prompt) should be carried out.

APPEL allows users to express such preferences as:

- Requests for personal information that will be given out to 3rd parties should be blocked.
- The user does not mind revealing click-stream and user agent information to sites that collect no other information.
- The user is comfortable with giving out the first and last name, as long as it is for non-marketing purposes. The user requires assurances (i.e., dispute information) from both "PrivacyProtect" and "TrustUs" and to be explicitly prompted before actually accessing such a page.
- When interacting with her bank's Web site at <http://www.my-bank.com>, she accepts any data request as long as her data is not redistributed to other recipients.

Rule 3 can be encoded as:

```

<appel:RULE
behavior="request" prompt="yes"
  promptmsg="Service only collects your name
             for non-marketing purposes (assured)
             Do you want to continue?">
  <p3p:POLICY>
    <p3p:STATEMENT>
      <p3p:PURPOSE appel:connective="or-exact">
        <p3p:current/>
        <p3p:admin/>
        <p3p:customization/>
        <p3p:develop/>
      </p3p:PURPOSE>
      <p3p:DATA-GROUP appel:connective="or-exact">
        <p3p:DATA ref="#user.name.*" />
      </p3p:DATA-GROUP>
    </p3p:STATEMENT>
    <p3p:DISPUTES-GROUP>
      <p3p:DISPUTES service="http://www.privacyprotect.com"/>
      <p3p:DISPUTES service="http://www.trustus.org"/>
    </p3p:DISPUTES-GROUP>
  </p3p:POLICY>
</appel:RULE>
    
```

Figure 72 Privacy Recommendation service: Rule 3 encoded

A.1.5 Architecture design

The architecture should be flexible to address infrastructure privacy as well as service privacy. The following diagram sketches the interactions of the recommendation service:

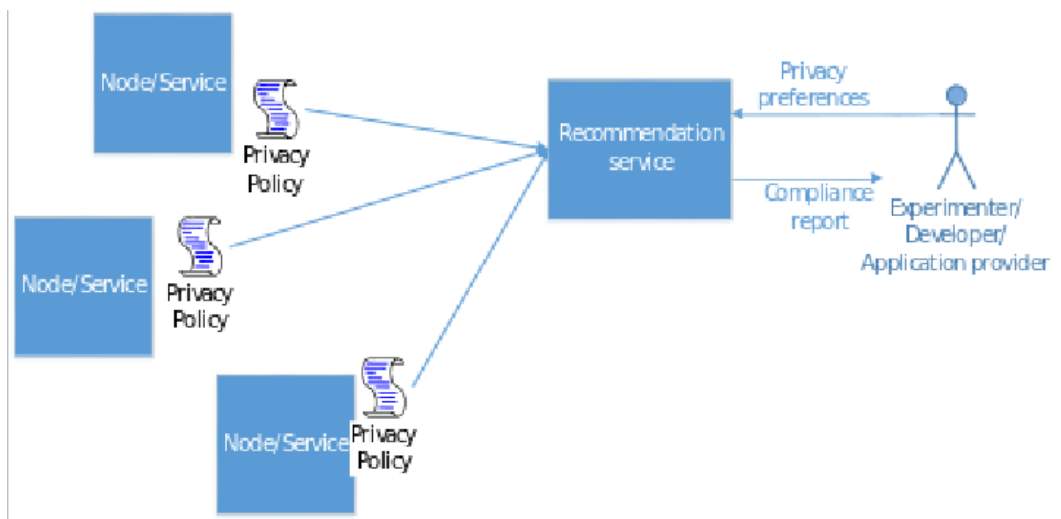


Figure 73 Recommendation service

Infrastructures publish a machine readable privacy policy that is accessible by the recommendation service. The experimenter provides their privacy preferences to the recommendation service which compares it with the different privacy policies (whether node or service policy). The recommendation service produces a compliance report detailing how each infrastructure covers the set of preferences.

The recommendation service building blocks are shown in the following figure:

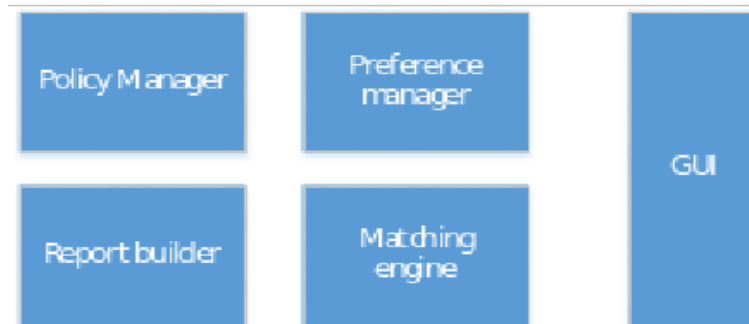


Figure 74 Recommendation service components

The recommendation service consists of a GUI that allows users to specify their privacy preferences. These preferences are managed by the “preference manager”. The “matching engine” uses the “policy manager” to collect the different privacy policies and compares them to the user’s privacy preferences. It then uses the “report builder” to provide the user with a report of the results to be displayed at the GUI.

The technology used in developing the privacy recommendation service is shown in red in the figure below.

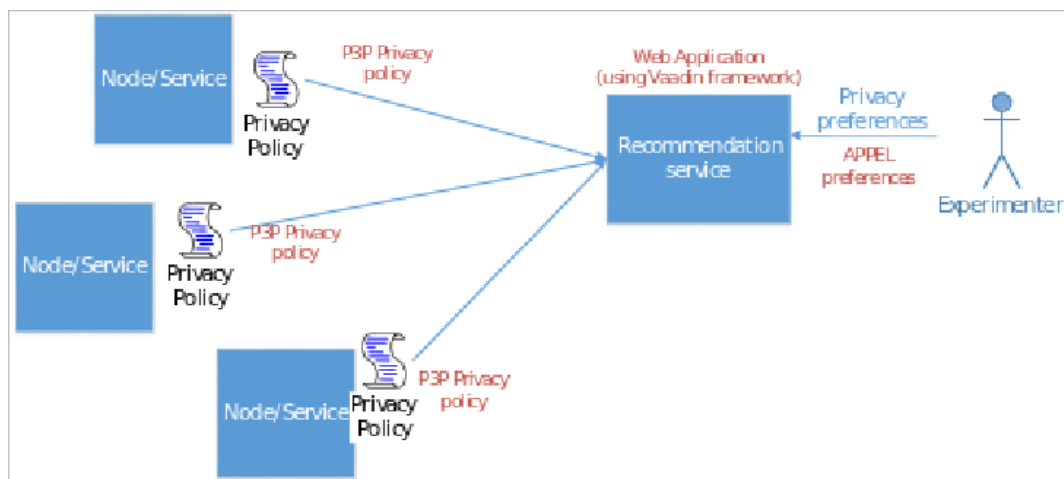


Figure 75 Recommendation service technology used

The privacy policies are encoded based on the W3C P3P recommendation for privacy policies. These are to be created by the different infrastructure providers for the node as whole or particular services. The location of these policies should be known (either a shared well-known location or a policy-reference file can be used at the root of the web server to indicate where the privacy policy is). In order to generate the privacy policies, there are online services/tools (e.g. <http://p3pedit.com/>) that provide services to specify P3P 1.0 and P3P 1.1 privacy policies. Moreover, the JRC Policy Workbench is an open source tool for creating a P3P policy and associating it to specific services hosted at the same website.

The Recommendation service is developed as a Web application using the Vaadin framework. It provides a GUI for the user to specify and evaluate their privacy preferences. These preferences are encoded according to the APPEL specification.

A.1.6 Release plan

Version ID	Milestone	User Stories
1.0	30/11/2014	1, 2,3

A.1.7 Test case

A.1.7.1 Test using pre-loaded policies

To test the service, you can use the already pre-loaded policies and rule sets. Note that the ruleset reference here is just a label. EU compliance for instance doesn't claim to cover all the EU regulations. The results should be as following:

Ruleset	Policy	Result
Low constraints	Policy1, Policy2	Accept, None of the prohibited characteristics have been found in this policy
High constraints	Policy1, Policy2	Reject, Personally identifiable info will be used beyond the stated purpose
EU Compliance	Policy1, Policy2	Reject, No compulsory marketing
Require address	Policy1, Policy2	Reject, Require identity and physical address of controller

A.1.7.2 Test using an uploaded policy

A natural language privacy policy is shown below:

This is our privacy policy for using our service. The service will collect some information when in use. This information includes the online connection related information such as the browser being used and information about the service usage (e.g. user clicks and paths taken when moving through the service). This information is stored indefinitely for Web site and system administration as well as research and development of the service.

We also uses cookies (small files on your computer) to store your preferences and navigation information. We access this information each time you visit our service.

You can opt-in to allow your demographic data (postcode) to be used for further analysis. This analysis doesn't identify you and the record will be associated with a pseudonym identifier.

The following XML shows the P3P policy:

```

<POLICY xmlns="http://www.w3.org/2000/12/p3pv1"
  discuri="http://www.provider.com/ourprivacypolicy.html"
  opturi="http://www.provider.com/optin.html">
  <ENTITY>
    <DATA-GROUP>
      <DATA ref="#business.name">Provider</DATA>
      <DATA ref="#business.contact-info.postal.street">street</DATA>
      <DATA ref="#business.contact-info.postal.city">city</DATA>
      <DATA ref="#business.contact-info.postal.stateprov">state</DATA>
      <DATA ref="#business.contact-info.postal.postalcode">postalcode</DATA>
      <DATA ref="#business.contact-info.postal.country">Country</DATA>
      <DATA ref="#business.contact-info.online.email">email</DATA>
      <DATA ref="#business.contact-info.telecom.telephone.intcode">44</DATA>
      <DATA ref="#business.contact-info.telecom.telephone.loccode">2380</DATA>
      <DATA ref="#business.contact-info.telecom.telephone.number">445566</DATA>
    </DATA-GROUP>
  </ENTITY>
  <ACCESS><nonident/></ACCESS>
  <STATEMENT>
    <PURPOSE><admin/><develop/></PURPOSE>
    <RECIPIENT><ours/></RECIPIENT>
    <RETENTION><stated-purpose/></RETENTION>
    <DATA-GROUP>
      <DATA ref="#dynamic.clickstream.server">
        <CATEGORIES><online/></CATEGORIES>
      </DATA>
      <DATA ref="#dynamic.http.useragent">
        <CATEGORIES><online/></CATEGORIES>
      </DATA>
      <DATA ref="#dynamic.cookies">
        <CATEGORIES><preference/><navigation/></CATEGORIES>
      </DATA>
    </DATA-GROUP>
  </STATEMENT>
  <STATEMENT>
    <PURPOSE><pseudo-analysis required="opt-in"/></PURPOSE>
    <RECIPIENT><other-recipient/></RECIPIENT>
    <RETENTION><indefinitely/></RETENTION>
    <DATA-GROUP>
      <DATA ref="#user.home-info.postal.postalcode">
        <CATEGORIES><demographic/></CATEGORIES>
      </DATA>
    </DATA-GROUP>
  </STATEMENT>
</POLICY>

```

Figure 76: Privacy Recommendation service: P3P policy example

- Copy the P3P XML policy into test.xml and save on your machine

- Add new policy via the button “+”
- Select “browse” and then select the file test.xml and click upload
- The policy will appear in the policies list
- Go to the preferences page and evaluate the rule sets again.
- The results of the newly added policy are as following.

Ruleset	Policy	Result
Low constraints	test	Accept, None of the prohibited characteristics have been found in this policy
High constraints	test	Accept, The default rule has fired i.e. none of the other rules led to policy rejection.
EU Compliance	test	Reject, Blocked because you cannot access all your data after submitting it
Require address	test	Reject, Require identity and physical address of controller

A.1.8 Installation manual

A.1.8.1 Pre-requisite software:

- Java Development Kit (JDK 7 and later) <http://www.oracle.com/technetwork/java/javase/overview/index.html>
- Maven (Version 2 and later) <http://maven.apache.org/download.cgi>
- Apache Tomcat server (8 or later) <http://tomcat.apache.org/>

A.1.8.2 Installation steps

The recommendation service is provided as a Web application jar file. It can be deployed into containers like Tomcat by simply putting the jar file in Tomcat/Webapps or via the Tomcat administrator interface.

A configuration file allows the administrator to tweak the service functionalities. The parameters to be filled are:


Configuration parameter	Description
require_authentication=false/true	This configures whether the OAuth 2.0 authentication should be used to authenticate users. Set to <i>true</i> to authenticate users and <i>false</i> to skip the authentication phase. If

	authentication is required then the service should be registered at the OAuth IdM and OAuth credentials obtained (to be provided in the parameters below).
idm_url=https://account.lab.fi-ware.org:443	This is the base url of the OAuth 2.0 Xifi IdM
authorize_url = /oauth2/authorize	This configures the authorisation url endpoint.
token_url=/oauth2/token	This configures the access token endpoint
userdata_url=/user	This indicates the user data endpoint. This will be used to pull information about the authenticated user.
verifier_name=code	This is the name of the parameters used by the OAuth server when provided and authorisation code.
scope=	This is the scope of oauth delegation. Fill if needed by the selected IdM.
callback_url=http://domain:8080/privacyservice-1.0/	This is the callback url which should refer to your privacy service installation.
client_id=...	The id provided by the OAuth IdM upon registering the privacy service
client_secret=	The secret provided by the OAuth IdM upon registering the privacy service

A.1.9 User manual

A.1.9.1 Authentication

The user must be authenticated in order to access the service. This is currently done via OAuth 2.0 XIFI FIWARE Lab IdM. The user should be registered at <https://account.lab.fi-ware.org/>. The Privacy recommendation service will redirect the user to the IdM in order to authenticate. Thanks to OAuth, user credentials are not communicated to the privacy recommendation service. However the privacy recommendation service will get the results of the authentication process. It will then let the user in if the authentication was a success.



Privacy Recommendation Service

This is the XIFI Privacy Recommendation Service.
This service allows you to test whether [P3P](#) privacy policies match your preferences. The preferences are specified according to [APPEL](#). Please Login to proceed.

Login with Xifi IDM

Figure 77: Authentication page

Note that the user may access the service via the marketplace. In this case the user is forwarded to the privacy recommendation service along with an access token (via a post html message). The service will check that it is valid by getting the user information from the IdM. If the user is valid then they will be able to access this service.

A.1.9.2 Policy loading

The user can load P3P policies in preparation for their evaluation. The “policies” tab shows the current list of policies ready for evaluation against the preferences. The user can add more to the list using the “+” button. The dialog appears giving two options for adding a policy:

- By providing the location of the policy online. A URL of the policy can be provided e.g. <http://domain.com/policy.xml>
- Upload the policy file from disk.

In both cases a user-defined “name” for the policy is required.

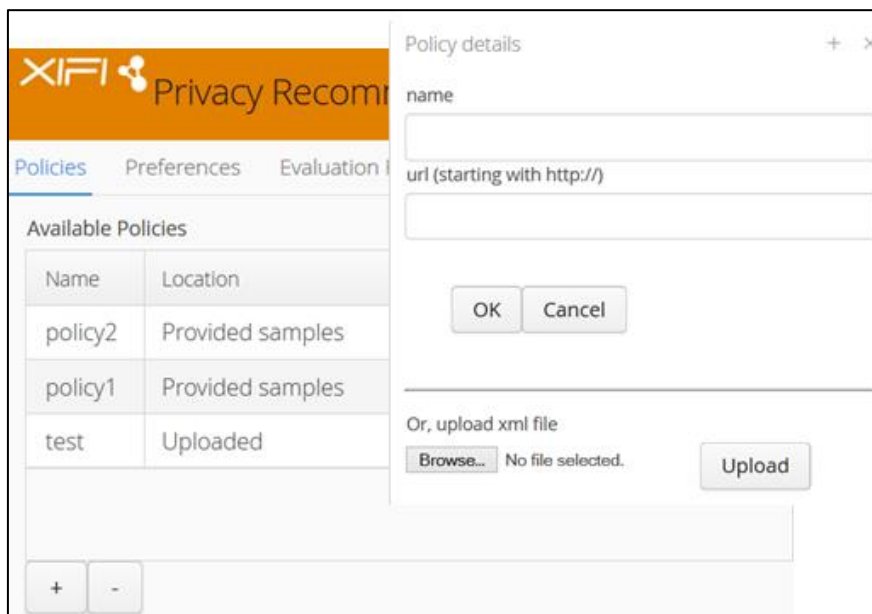


Figure 78: Policies view

The service has already a policy sample loaded. The “Location” column shows the origin of the policy. According to the way the policy was loaded, this will show one of the following:

- The policy url if the policy has been fetched from online location
- “Uploaded” if the policy file upload option was used
- “Provided samples” if this policy was loaded from the service packaged samples.

A.1.9.3 Preferences specification

A.1.9.3.1 Understanding the layout

The preferences tab shows the APPEL rule sets specified by the user. For convenience, sample rule sets are displayed and can be used and edited directly by the user. Each rule set has a user-defined name includes a set of rules which are displayed in the “Rule” column.

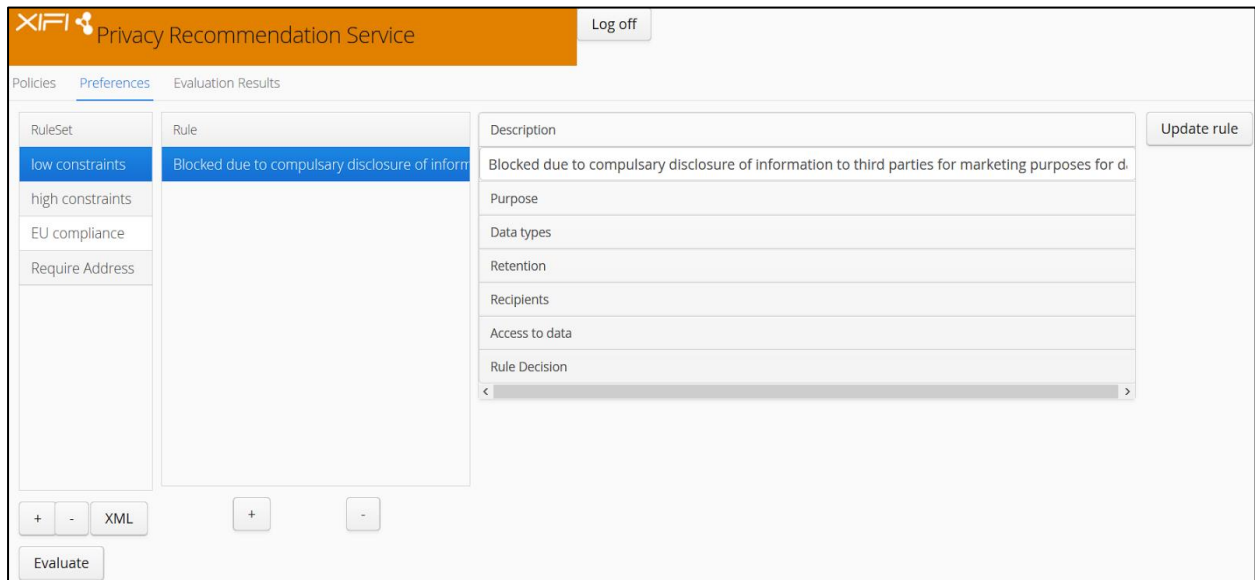


Figure 79: Preferences layout

Rule sets can be added to the list via the “+” button underneath. The following dialog is then displayed providing two options to collect the rule set information:

- A user defined name of the Rule set which creates an empty rule set and adds it to the list. The user can then manually add rules to the rule set.
- Upload rule set xml file. This will upload the rule set details including any rules specified there.

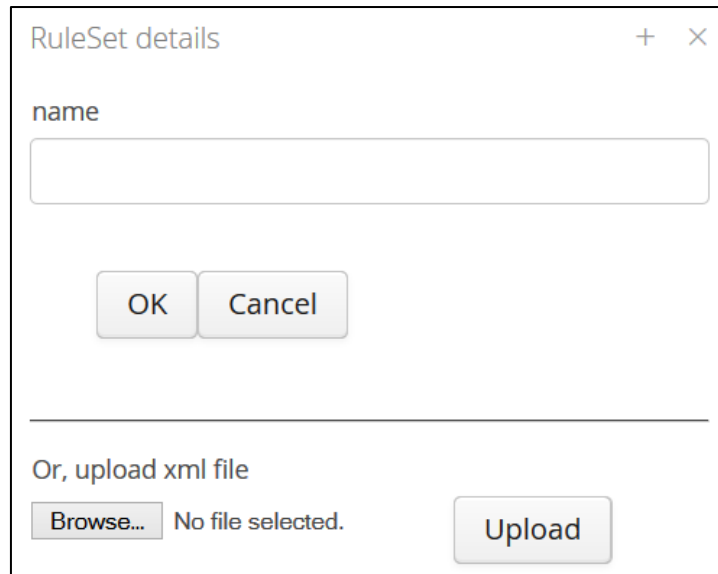


Figure 80: Add RuleSet

In a similar way with the “-“ button the user can remove the selected Rule set from the list.

The “Xml” button allows visualizing the rule set in xml form. The xml can be copied and saved to be uploaded later on if needed.

A.1.9.3.2 Editing rules

Rules can be added into a rule set via the “+” button underneath the rules column. A new empty rule is created and can be edited via the collapsed categories on the right side. Under each category a set of information can be provided. The rule is then updated with this information once the “update rule” button is clicked (for instance the description in the rule list will be updated according to the description details provided by the user).

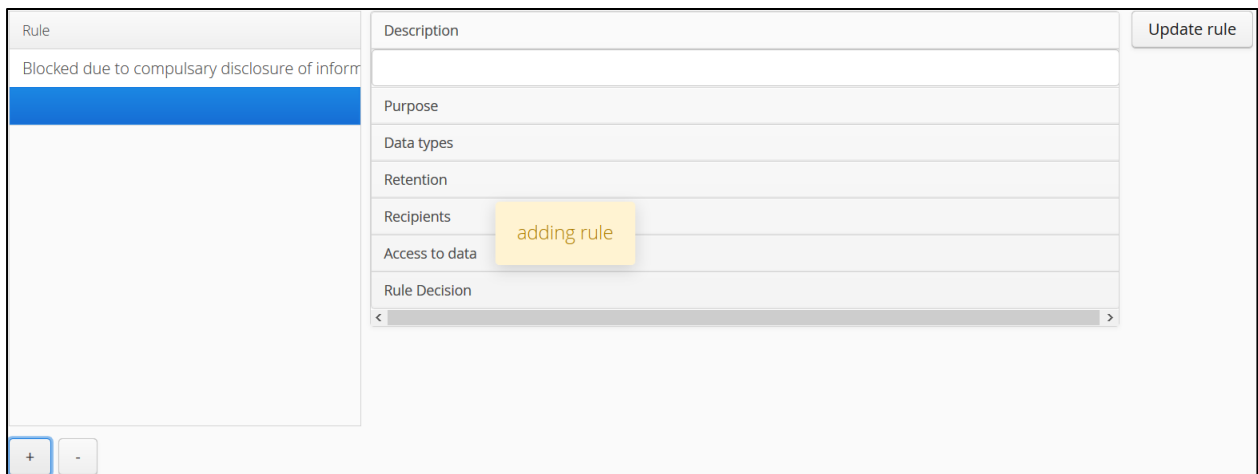


Figure 81: Add rules

A rule allows the user to specify what they want to happen if the policy matches a set of conditions. For instance the user can specify a rule that says “if the purpose of collecting data is marketing, then reject the policy” or “if there is no retention period of the collected data then accept the policy”. These

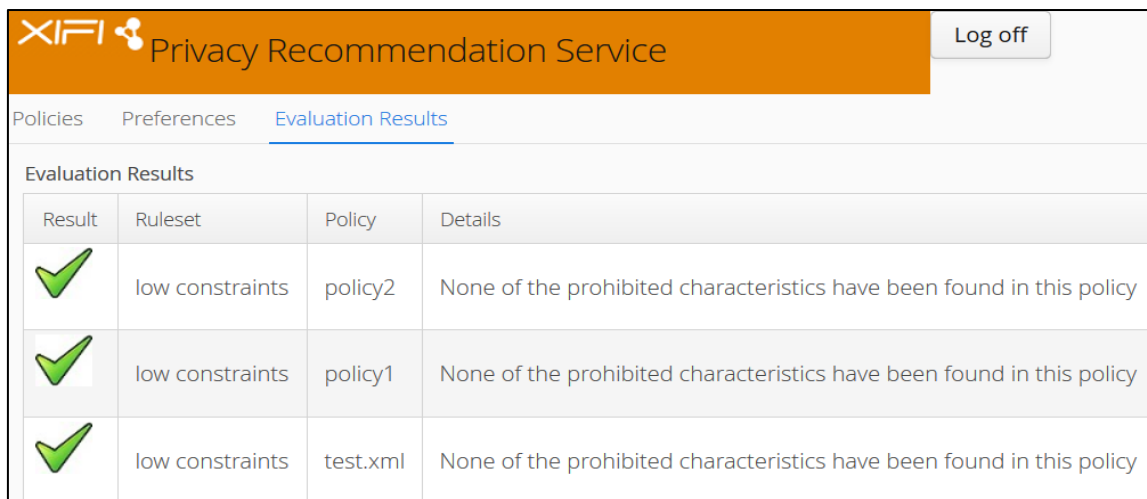
conditions are encoded under the rule details. These include:

- **Description:** description of the rule. This is displayed under the rule column once the update rule button is clicked.
- **Purpose:** This allows specifying the policy purpose conditions for this rule (i.e. for what purpose the data is collected).
- **Data types:** this allows the specification of the data types to be considered under this rule. A list of data types defined by P3P is provided for the user to select from.
- **Retention:** this specifies the retention period of the collected (if any) data.
- **Recipients:** this specifies the recipients of the user collected data.
- **Access to data:** this specifies what retained data the user has access to.
- **Rule decision:** this specifies the rule decision (i.e. accept or reject the policy) if the specified conditions in the categories match the policy.

For convenience, under each category there is a “help” link that explains the meaning of the terms under that category.

A.1.9.4 Rule Set Evaluation

APPEL 1.0 relies on ordered rules. APPEL rules are intended to express preferences over P3P policies.






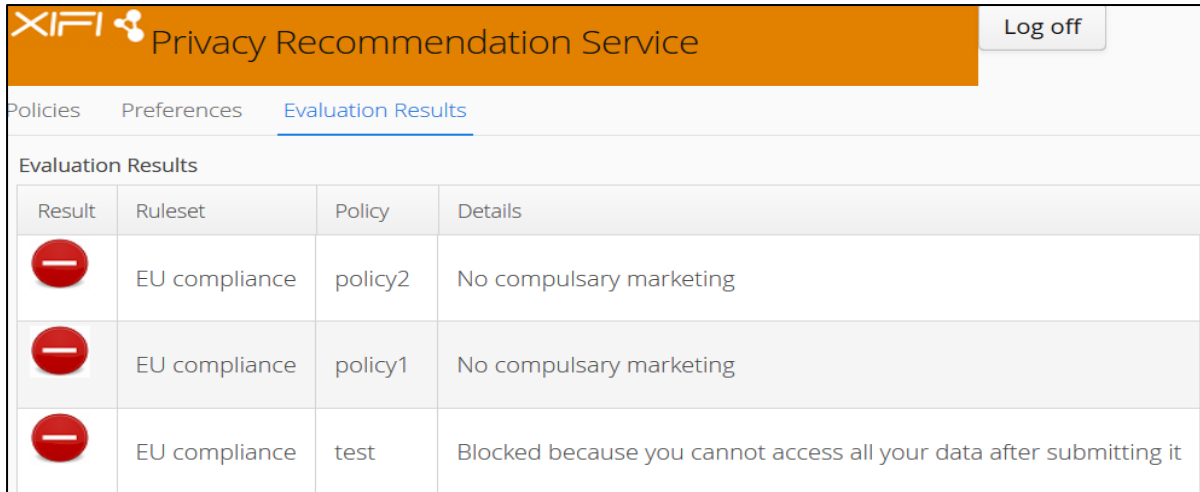
XIFI Privacy Recommendation Service Log off			
Policies Preferences <u>Evaluation Results</u>			
Evaluation Results			
Result	Ruleset	Policy	Details
	low constraints	policy2	None of the prohibited characteristics have been found in this policy
	low constraints	policy1	None of the prohibited characteristics have been found in this policy
	low constraints	test.xml	None of the prohibited characteristics have been found in this policy

Figure 82: Evaluation results- Policies accepted

To evaluate a rule set, the user has to select the rule set in the Rule Set column and click the “evaluate” button underneath (see Figure 6 Preferences layout). The service will then evaluate the selected rule set against all the policies listed in the policies tab. Once the evaluation is done, the user is automatically redirected to the Evaluation Results tab where the results are shown.






XIFI Privacy Recommendation Service			
Log off			
Policies Preferences Evaluation Results			
Evaluation Results			
Result	Ruleset	Policy	Details
	EU compliance	policy2	No compulsory marketing
	EU compliance	policy1	No compulsory marketing
	EU compliance	test	Blocked because you cannot access all your data after submitting it

Figure 83: Evaluation results – Policies rejected

The results include the final decision of accepting or rejecting the policy according to the Rule Set rules. It includes the name of the rule set and policy being evaluated. Moreover, it displays the details of why the decision was made. In this example none of the undesired conditions specified in the rules was met. If a rule causes the evaluation of the policy to fail (i.e. policy rejected) then the description of that rule is shown here. This requires that the rule contains meaningful description that allows the results to be understood.

P3P policy

The following is an example taken from the P3P specification that shows how an English-language privacy policy can be encoded as a P3P policy. The example is provided in both English and as a more formal description using P3P element and attribute names.

CatalogExample's Privacy Policy for Browsers

At CatalogExample, we care about your privacy. When you come to our site to look for an item, we will only use this information to improve our site and will not store it with information we could use to identify you.

CatalogExample, Inc. is a licensee of the PrivacySealExample Program. The PrivacySealExample Program ensures your privacy by holding Web site licensees to high privacy standards and confirming with independent auditors that these information practices are being followed.

Questions regarding this statement should be directed to:

CatalogExample

4000 Lincoln Ave.

Birmingham, MI 48009 USA

email: catalog@example.com

Telephone 248-EXAMPLE (248-392-6753)

If we have not responded to your inquiry or your inquiry has not been satisfactorily addressed, you can contact PrivacySealExample at <http://www.privacyseal.example.org>.

CatalogExample will correct all errors or wrongful actions arising in connection with the privacy policy.

What We Collect and Why:

When you browse through our site we collect:

- the basic information about your computer and connection to make sure that we can get you the proper information and for security purposes.
- aggregate information on what pages consumers access or visit to improve our site.

Data retention:

We purge every two weeks the browsing information that we collect.

The formal description is shown below, using the P3P element and attribute names [with the section of the spec that was used cited in brackets for easy reference]:

Disclosure URI: <http://www.catalog.example.com/PrivacyPracticeBrowsing.html>

[[3.2.2 Policy](#)]

Entity: CatalogExample

4000 Lincoln Ave.

Birmingham, MI 48009

USA

catalog@example.com

+1 (248) 392-6753

[[3.2.5 Entity](#)]

Access to Identifiable Information: None

[[3.2.6 Access](#)]

Disputes:

resolution type: independent

service: <http://www.privacyseal.example.org>

description: PrivacySealExample

[[3.2.7 Disputes](#)]

Remedies: we'll correct any harm done wrong

[[3.2.8 Remedies](#)]

We collect:

dynamic.clickstream

dynamic.http

[[5.5 Base data schema](#)]

For purpose: Web site and system administration, research and development

[[3.3.5 Purpose](#)]

Recipients: Only ourselves and our agents

[[3.3.6 Recipients](#)]

Retention: As long as appropriate for the stated purposes

[[3.3.7 Retention](#)]

(Note also that the site's human-readable privacy policy MUST mention that data is purged every two weeks, or provide a link to this information.)

The XML encoding of the policy:

```

<POLICIES xmlns="http://www.w3.org/2002/01/P3Pv1"
  xmlns:p3p11="http://www.w3.org/2006/01/P3Pv11">
  <POLICY name="forBrowsers"
    discuri="http://www.catalog.example.com/PrivacyPracticeBrowsing.html"
    xml:lang="en">
    <ENTITY>
      <EXTENSION>
        <p3p11:data-group>
          <p3p11:datatype>
            <p3p11:business>
              <p3p11:orgname>CatalogExample</p3p11:orgname>
              <p3p11:contact-info>
                <p3p11:postal>
                  <p3p11:street>4000 Lincoln Ave.</p3p11:street>
                  <p3p11:city>Birmingham</p3p11:city>
                  <p3p11:state>MI</p3p11:state>
                  <p3p11:postalcode>48009</p3p11:postalcode>
                  <p3p11:country>USA</p3p11:country>
                </p3p11:postal>
                <p3p11:online>
                  <p3p11:email>catalog@example.co.uk</p3p11:email>
                </p3p11:online>
                <p3p11:telecom>
                  <p3p11:telephone>
                    <p3p11:intcode>1</p3p11:intcode>
                    <p3p11:loccode>248</p3p11:loccode>
                    <p3p11:number>3926753</p3p11:number>
                  </p3p11:telephone>
                </p3p11:telecom>
              </p3p11:contact-info>
            </p3p11:business>
          </p3p11:datatype>
        </p3p11:data-group>
      </EXTENSION>
      <DATA-GROUP>
        <DATA ref="#business.name">CatalogExample</DATA>
        <DATA ref="#business.contact-info.postal.street">4000 Lincoln Ave.</DATA>
        <DATA ref="#business.contact-info.postal.city">Birmingham</DATA>
        <DATA ref="#business.contact-info.postal.stateprov">MI</DATA>
        <DATA ref="#business.contact-info.postal.postalcode">48009</DATA>
        <DATA ref="#business.contact-info.postal.country">USA</DATA>
        <DATA ref="#business.contact-info.online.email">catalog@example.com</DATA>
        <DATA ref="#business.contact-info.telecom.telephone.intcode">1</DATA>
        <DATA ref="#business.contact-info.telecom.telephone.loccode">248</DATA>
      </DATA-GROUP>
    </ENTITY>
  </POLICY>
</POLICIES>

```

```

    <DATA ref="#business.contact-info.telecom.telephone.number">3926753</DATA>
  </DATA-GROUP>
</ENTITY>
<ACCESS><nonident/></ACCESS>
<DISPUTES-GROUP>
  <DISPUTES resolution-type="independent"
    service="http://www.PrivacySeal.example.org"
    short-description="PrivacySeal.example.org">
    <IMG src="http://www.PrivacySeal.example.org/Logo.gif" alt="PrivacySeal's logo"/>
    <REMEDIES>
      <correct/>
    </REMEDIES>
  </DISPUTES>
</DISPUTES-GROUP>
<STATEMENT>
  <PURPOSE>
    <admin/>
    <develop/>
  </PURPOSE>
  <RECIPIENT>
    <ours/>
  </RECIPIENT>
  <RETENTION>
    <stated-purpose/>
  </RETENTION>
  <!-- Note also that the site's human-readable
  privacy policy MUST mention that data
  is purged every two weeks, or provide a
  link to this information. -->
  <EXTENSION>
    <p3p11:data-group>
      <p3p11:datatype>
        <p3p11:dynamic>
          <p3p11:clickstream/>
          <p3p11:http/>
        </p3p11:dynamic>
      </p3p11:datatype>
    </p3p11:data-group>
  </EXTENSION>
  <DATA-GROUP>
    <DATA ref="#dynamic.clickstream"/>
    <DATA ref="#dynamic.http"/>
  </DATA-GROUP>
</STATEMENT>
</POLICY>
</POLICIES>

```

APPEL evaluation algorithm

The evaluator engine is provided with various pieces of evidence and a rule set in addition to the P3P base data schema and any custom data schemas referenced in the evidence. Evidence includes URI of the service (to be accessed, given the background of the tool) and a single P3P policy from the service. The rule set is composed of rules. Each rule has behaviour, prompt and promptmsg attributes. The evaluator returns the behaviour (as specified by those attributes) of the rule that fired on the basis of the evidence (URI and policy).

Behaviour

The behaviour options are block, request and limited.

A user-agent application should interpret the behaviour outputs as follows:

- Request: the provided evidence is acceptable. The user is allowed to access the service.
- Limited: the provided evidence is not fully acceptable. The user is allowed to access the service with limitations (all but absolutely necessary request header should be suppressed)
- Block: the provided evidence is not acceptable. The service shouldn't be accessed.

Given that we are not using this specification to automatically check user requests and act upon them, we can limit the behaviour in our context to Request or Block. The Block behaviour is interpreted as the policy doesn't match the preferences whereas the Request behaviour is interpreted as the policy satisfies the preferences.

Prompt

Applications should interpret the prompt attribute as follows:

- Prompt="no": the behaviour should be performed without soliciting input from the user
- Prompt="yes": the user should be prompted for a decision whether the behaviour triggered by the rule should be performed.

In our context, we are not proceeding with any automated action on behalf of the user. In any case, the provided tool will just let the user know the result of the evaluation. This attribute is ignored by our service.

Rule processing

A rule is evaluated to true if all of its enclosed expressions are satisfied. An expression is satisfied if any of the available evidence satisfies it. Each rule in the rule set is evaluated in the order it appears in. Once a rule evaluates to true, the corresponding behaviour is returned and rule evaluation ends. However evaluation should continue with the rest of the rules in order to have a comprehensive list why a particular behaviour got triggered (e.g. have a full view of why block was triggered which may be because of more than one preference rule). Evaluation should continue to find all rules with similar behaviour and prompt attribute values and return the aggregated result. This feature can be added to enhance the current implementation.

Rule sets should be written so that there is always a rule that will fire. A rule evaluator should return an error if it is called without a rule set, empty rule set or if no rule fires. This is usually overcome with a default rule at the end which encodes the default behaviour required (e.g. block rule at the end

to catch any otherwise cases).

```
<appel:RULE behavior="request" description="The default rule has fired">  
<appel:OTHERWISE/>  
</appel:RULE>  
</appel:RULESET>
```