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

eXperimental Infrastructures for the Future Internet

D1.5: Federated Platform Architecture v2

Revision: 1.2

Work package	WP 1
Task	Task 1.3
Due date	30 September 2014
Submission date	30 September 2014
Deliverable lead	University of Southampton IT Innovation Centre
Authors	Berndt Bochow (Fraunhofer-Fokus), Silvio Cretti (CNET), Stefano Modafferi (IT Innovation), Brian Pickering (IT Innovation), Maria Di Girolamo (ENG), Joaquim Iranzo (ATOS), Cyril Dangerville (THALES), Panos Trakadas (Synelixes), Fernando Lopez (TID), Eamon Power (Waterford), Cristian Cristellotti (TNetwork), Sergio Morant (ImagineLab), Giuseppe Cossu (CNET)
Reviewers	Fernando Lopez (TID), Daniel Nehls (TUB)

Abstract	This deliverable provides the second complete version of the XIFI federation architecture. It is the updated version of D1.3 [4] also exploiting the work carried out in WP1 and documented in D1.1.1 [1], D1.4 [5], Milestone MS1.3 [6] and in WP8 reported in D8.2 [8].
Keywords	Federation, Architecture, GE, Change Management Process,

Document Revision History

Version	Date	Description of change	List of contributor(s)
0.1	25.07.2014	First draft (skeleton)	Stefano Modafferi (IT Innovation)
0.2	10.09.2014	Version present in the wiki page imported as word document	A.A.V.V.
1.0	20.09.2014	First complete draft	Stefano Modafferi (It-Innovation
1.2	29.09.2014	Implementation of reviewers' comments	Stefano Modafferi (It-Innovation

Disclaimer

Copyright notice

Project co-funded by the European Commission in the 7th 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

This deliverable provides the second complete version of the XIFI federation architecture. It is the updated version of the deliverable D1.3 [4] and takes into account the new requirement list described in D1.4 [5] and accommodates it into the architectural framework. The new version of the architecture proves how now the architecture can be considered stable because few modifications were needed from previous version. However a change management process for updating the architecture is presented in order to control the impact of future modifications.

At this stage of the project some validations of the architecture have to be carried out. These validations should consider not only the requirement satisfaction but also:

- the coverage of the design principles elicited in D1.1.1 [1],
- the opinions of different stakeholders, like the five original infrastructure owners and the UC projects that have started to use the FIWARE Lab federation

As a result of these validations, the architectural framework proved to be well suited for supporting FIWARE Lab in accommodating large scale trials and addressing requirements issued by the relevant XIFI stakeholders. Nevertheless some improvements to the architecture or, better, to its real deployment on the different nodes of the federation could be needed in order to solve some issues present in the current deployment of the FIWARE Lab federation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

API	Application Programming Interface
AUP	Acceptable Usage Policy
DCRM	Data Centre Resource Management
FI-PPP	Future Internet Public-Private-Partnership
FMC	Fundamental Model Components
GE	Generic Enabler
GEi	Generic Enabler implementation
GUI	Graphical User Interface
IaaS	Infrastructure as a Service
IdM	Identity Management
IP	IPv4, IPv6 Internet Protocol (version 4, version 6)
NREN	National research and Education network
PaaS	Platform as a Service
SAML	Security Assertion Markup Language
SCIM	Simple Cloud Identity Management
SDC	Software Deployment and Configuration
SE	Socio-economic
SLA	Service Level Agreement
UI	User Interface
URI	Uniform Resource Indicator
USDL	Unified Service Description Language
VLAN	Virtual Local Area Network
VM	Virtual Machine
XACML	Extensible Access Control Markup Language

1 INTRODUCTION

1.1 Context, objective and scope of this deliverable

This deliverable provides the second complete version of the XIFI federation architecture. After summarizing the results of the milestone MS13 and the work on the identification of the right federation model for FIWARE Lab, this document describes the current architecture as an incremental version of what already presented in D1.3 [4]. The mapping between new requirement list (defined in D1.4 [5]) and architectural components highlights how requirements are satisfied and how new requirements have not impacted the whole architecture definition framework. Moreover the analysis of the FIWARE GEs used in the architecture proves how the XIFI architecture is founded on FIWARE and how some GEs has been modified in order to be suitable for XIFI.

In order to handle and control future possible modifications to the architecture, the change management process, adopted for upgrading the architecture, is explicitly described in this document.

The architecture is now stable and few changes occurred from M9, when D1.3 has been issued, and M18, when this document has been written. The validation of the design principles, occurred in Milestone MS1.3, has demonstrated the soundness of the architecture on that line. Further validation comes from the answers to a questionnaire submitted to the five original infrastructure owners and from the report of the actual experience of Use Case projects (e.g. FITMAN[2]) in using the FIWARE Lab federation.

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

- Developers and Technology Providers including UC project participants so they can verify how XIFI will satisfy their requirements and provide feedback if needed.
- Infrastructure owners and operators so they can understand the up to date design of the architecture that they have to support.
- XIFI architects and developers who need to have a clear view of the state of the art in federation architecture and of potential improvements in the features provided.
- XIFI federation operators so as to provide a clear picture of federated features and the type of operations supported by the XIFI federated architecture.

The document is divided into the following sections:

- **Section 1** (this section) introduces the context, the objectives and scope of the document.
- **Section 2** reports a summary of design principles and federation models.
- **Section 3** describes the second version of the federation architecture.
- **Section 4** provides a first validation of the architecture by means of a questionnaire, filled in by five infrastructure owners, and a report on the usage of FIWARE Lab by Use Case projects.
- **Section 5** provides a summary of conclusions.

[2]Fitman project: http://www.fitman-fi.eu/project-overview

2 DESIGN PRINCIPLES AND FEDERATION MODEL

XIFI Design Principles have been defined in D1.1.1 [1] and they have been validated in MS13 [6]. Federation models have been jointly analysed in WP8 and WP1 and the rationale behind the chosen one is reported in D1.3 [4]. No differences are present in design and models since the document and therefore they are reported below as they have been presented the first time. Nevertheless it is useful to recap the main concepts in the following sections.

2.1 Design Principles

XIFI architecture defines a community cloud platform and, as such, it should adhere to canonical cloud computing design principles [2] and heterogeneous cloud deployment best practises [3]. Table 1 reports the principles and their evaluation on if and how a given design principle has been applied to the XIFI architecture. This material is taken from MS13 [6].

	Principle	Description	Application of the principle to the current architecture
DP1	On-demand self-service [2]	"A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider".	The re-usage of the Cloud Chapter GEs, part of FIWARE, satisfies this principle, namely: Cloud Portal (or SelfService Interfaces) GE [13], PaaSManager GE [12], SDC Manager GE [11] and IaaS DCRM GE [10]. Moreover the XIFI Resource Catalogue & Recommender component [14] offers support to the user in order to select the services needed. Both End-Users and Technology providers have stressed the importance of this principle and its implementation: XIFI is therefore able to satisfy two complementary and sometimes conflicting points of view.
DP2	Broad network access [2]	"Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations)".	Services and GEs/SEs offered through FIWARE Lab are available via network and are accessible via TCP/IP and HTTP compatible standards. In particular RESTFul APIs are defined for each service. The FIWARE Lab GUI Portal is implemented so as to be accessible from any modern user device, ranging from mobile phones to workstations.

	Principle	Description	Application of the principle to the current architecture
DP3	Resource pooling [2]	"The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data centre). Examples of resources include storage, processing, memory, and network bandwidth".	In FIWARE Lab resources are pooled to serve multiple tenants at the same time. The physical resources are shared across virtual ones and assigned to active virtual resources. FIWARE Lab offers a federation of resources (mainly data centres but not only) located in different countries: each node is called "region" in the cloud terminology. The developers will be able to specify the node in which the resources are hosted. To allow for this, the XIFI architecture is designed to support federation of different nodes and services through the use of the Cloud Portal and the Security/Keystone Proxy [15].
DP4	Rapid elasticity [2]	"Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time".	Thanks to the Cloud Management Platform adopted (FIWARE Cloud Hosting GEs) and the data collected through the FIWARE Lab Monitoring System, XIFI is able to guarantee scalability and elasticity in both automatic and manual way.
DP5	Measured service [2]	"Cloud systems automatically control and optimize resource usage by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service".	FIWARE Lab Monitoring System implements this principle offering data collection from different sources, data aggregation and finally data visualization and reporting. Moreover FIWARE Lab offers SLA Management based on monitoring data.
DP6	User Centric [3]	Well-designed services and user interfaces are key to deliver best user experience that will facilitate adoption of cloud services.	All the services and user interfaces, developed in XIFI, are compliant and integrated into the FIWARE Lab federation offering an uniform access and behaviour.

	Principle	Description	Application of the principle to the current architecture
DP7	Simplicity [3]	The adoption of complex solutions may impact delivery time and the service quality. Dealing with heterogeneous environments may require complex solutions to automate some of the processes. Some processes may be kept to a manual mode in an initial phase for time–to-market, rather than designed with full functionality and for all IT services.	XIFI is trying to simplify the user experience automatizing the process of set up of a development environment (PaaS Manager) and the access to the different services (single and uniform access point). Where needed, XIFI architecture hides the heterogeneity of the different resources federated in FI-Lab (monitoring and identity management).
DP8	Reuse [3]	Cloud computing is nowadays a well-established field where a plethora of solutions are available off the shelves to be deployed to support the creation of cloud based infrastructures. Such solutions should be reused as much as possible to allow for a fast kick-start of cloud provisioning activities, unless there are strong reasons to develop a new solution.	XIFI is committed to use the FI-PPP technologies and in particular the FIWARE GEs. The concept of enablers addresses by definition, the reusability.
DP9	Service Dependability [16]	Dependability is a fundamental characteristic of cloud computing platforms. Service availability is one of the key attributes for a service to be defined dependable. Service availability requires dealing with issues such as no single point of failure and scalability/elasticity mentioned above. While some of these features are not dependant on the provisioning platform, but rather on the services developed on top of the platform, in the design of Cloud provisioning platform service availability should be careful planned also taking into consideration derived costs [3].	XIFI architecture pursues dependability and tries to avoid any single point of failure: the main and "critical" services like Identity Manager, Security Proxy, the Portal, etc. are provided in high availability.
DP10	Flexibility	In software engineering, the ability to adapt to evolving requirements is fundamental to allow for rapid prototyping and deployment. The more architecture is modular the more it is flexible and adaptable.	The XIFI architecture is defined in a modular way and the service paradigm is used thoroughly. Each component has a service interface used for external access and for communications among components.
DP11	Compatibility	Compatibility is the other side of "re-usage" coin, APIs should be as much as possible compatible with existing reference standards and de-facto standards to not constitute lock-in.	XIFI architecture implements adaptation layers where needed for making possible for legacy structure to become part of the XIFI federation. This is particularly emphasized in the Monitoring System.

Table 1: XIFI principles

2.2 Federation model

Based on the analysis of requirements and models, the federation model for FIWARE Lab chosen among the ones defined in [7] and analysed in D1.3 and after that in D8.2 [8] is a hybrid federation model covering aspects of the "one-stop-shop" model (a common advertising channel for infrastructures where users choose which resource/service they want to use), and the "integrator" model (the federator decides which resource/service they will use). Whether the user or the federation decides which resources will be allocated depends on the type of resources. Conventional data resources are allocated by FIWARE Lab and accessed in a uniform way under common terms by users (i.e. the "integrator" model can be applied), but non-conventional resources (e.g. sensor networks, etc.) are advertised by the federator and negotiated/accessed directly with/through the infrastructures.

3 UPDATES ON FEDERATION ARCHITECTURE (D1.3)

In this section we provide an up to date version (ver. 2.0) of the federation architecture with respect to the one presented in D1.3 [4]. The changes requested are mainly derived by the new set of high level requirements defined in D1.4 [5] (for a description of the procedure that conducts to an update to the architecture, see Section 0). In fact the changes with respect to the previous version of the federation architecture are minimal and do not affect the overall architectural framework. So most part of the information present hereunder is the same already provided in D1.3. Nevertheless the few updates made to the architecture are highlighted and commented in the section 3.5.

The following FMC diagram provides the overall architecture diagram:

- the bottom part of the figure contains the architecture and the components deployed on each node of the FIWARE Lab (XIFI) federation (both master and slave)
- the upper part of it contains the architecture and components deployed only on the master nodes (for a definition of master and slave node, see D1.1.1 [1]).

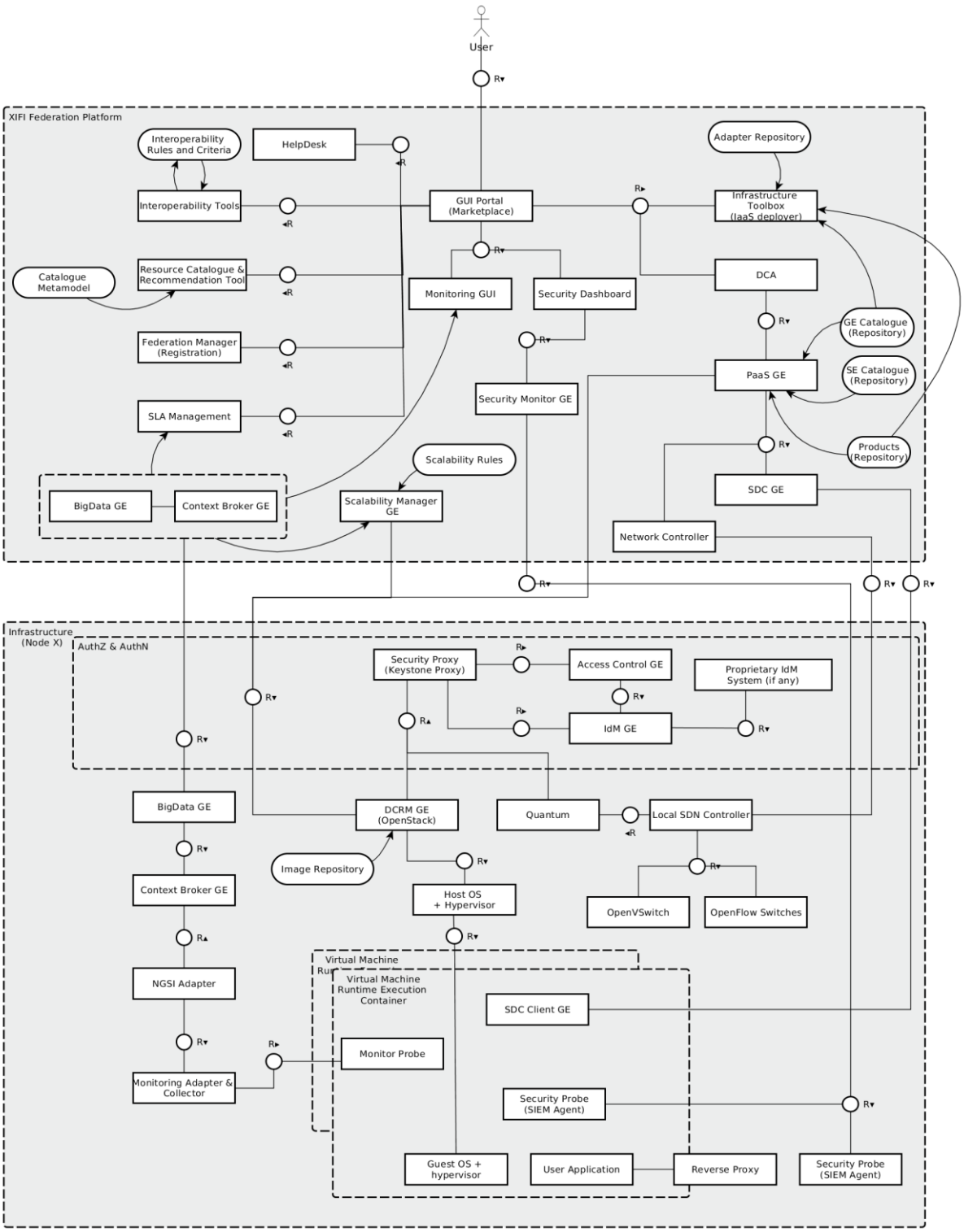

Figure 1: Overall Architecture ver. 2.0[3]

[3] Notes on the naming in the figure:

- In the last version of FIWARE "Scalability Manager GE" has changes its name in "Policy Manager GE".
- The current name of "Quantum" is "Neutron"

The two next sections describe the two parts of the architecture: as in the D1.3, the following description is limited to an overview of each component; more detailed description will be provided in the deliverables describing the requirements and design of each component or in the corresponding FIWARE documentation.

3.1 Architecture of a generic (slave or master) XIFI Node

Three main functional groups can be identified in the bottom part of the previous figure. For each functional group changes, occurred after D1.3 version, are signalled.

1. Components enabling cloud computing (slightly changed from the D1.3)
2. Components enabling monitoring functionalities (no changes from D1.3)
3. Components enabling security functionalities (slightly changed from the D1.3)

3.1.1 Components enabling cloud computing

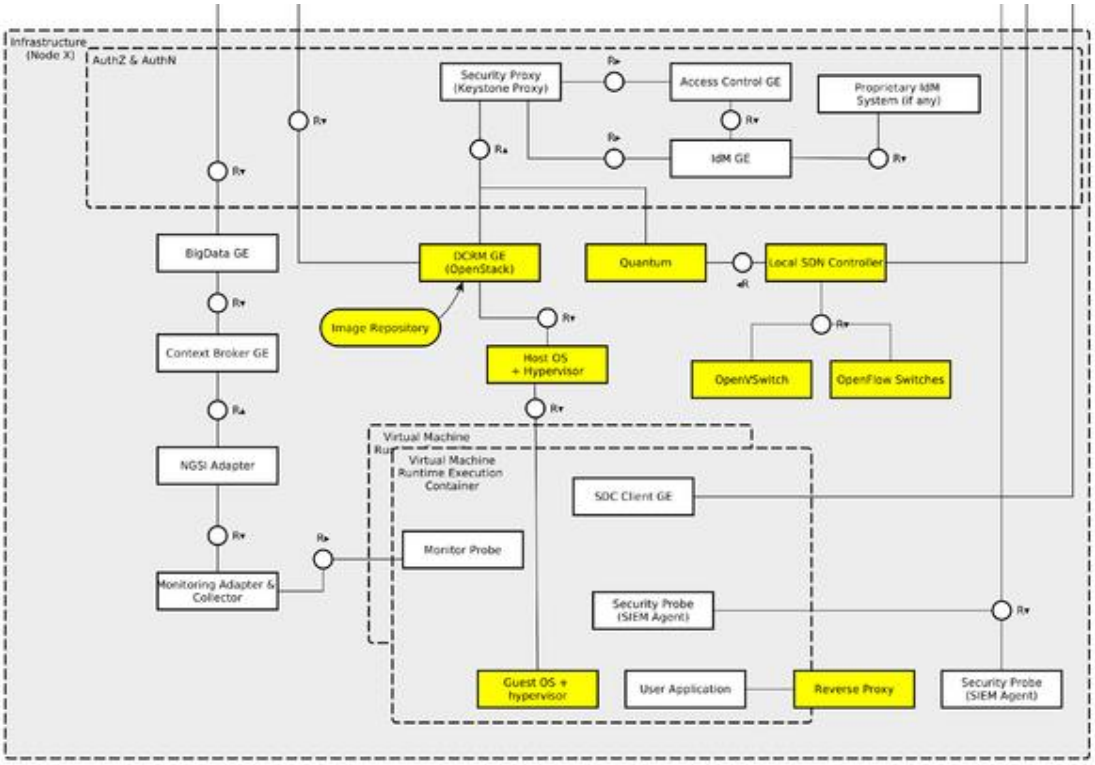

Figure 2: Components Enabling Cloud Computing ver. 2.0

These components (in yellow in the previous figure) enable the setup of a cloud computing environment based on OpenStack [9]: the *FIWARE DCRM GE*[10] wraps OpenStack and, together with *Neutron (Quantum)*, *Local SDN Controller* (connected to the master Network Controller component), *Open vSwitch*[4] and *OpenFlow Switches*[5] components, provides all the services requested to a IaaS Management System. Moreover each virtual machine will be equipped with the *FIWARE*

[4]OpenSwitch: http://openvswitch.org/

[5]http://archive.openflow.org/wp/documents/

SDC Manager GE client, connected to the *FIWARE SDC Manager GE* [11], present only in the master node, in order to deploy different products and GEis. The *Reverse Proxy* component helps to prevent the consumption of public IPs when accessing end user applications and services.

3.1.2 Components enabling monitoring functionalities

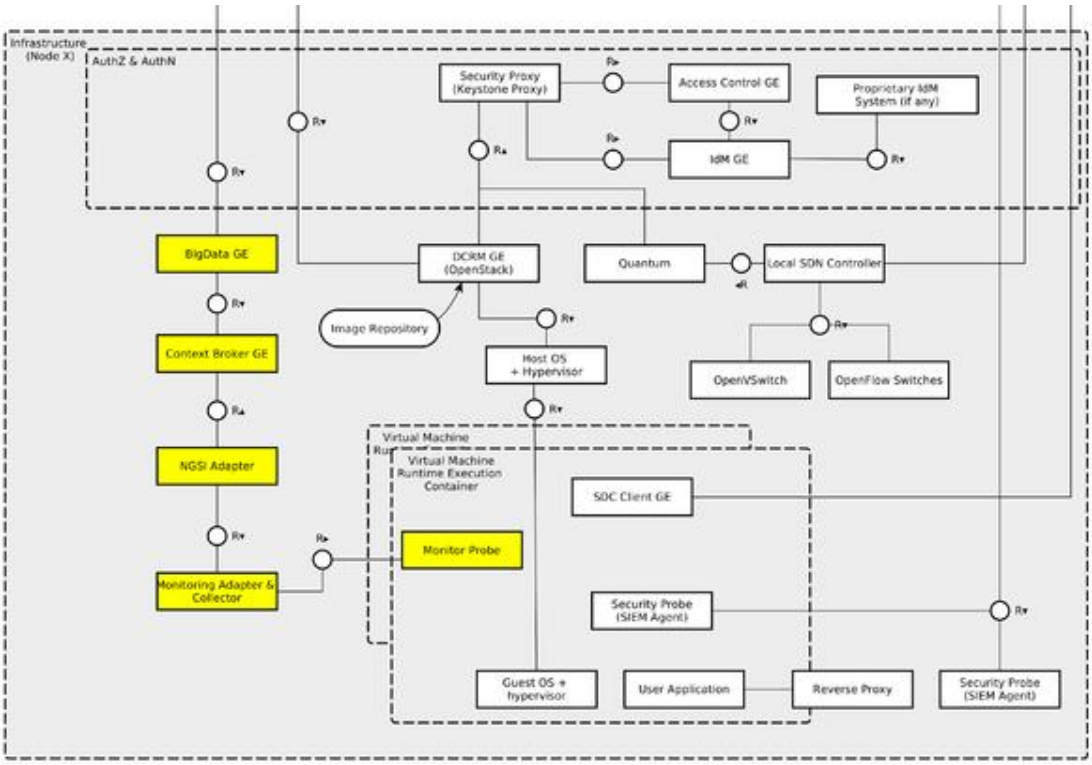

Figure 3: Components enabling Monitoring Functionalities ver. 2.0

These components (in yellow in the previous figure) enable the monitoring functionalities collecting data from physical devices, network devices, virtual machines and services. Data can be collected interfacing local *Monitoring Adapters & Collectors* tools (like for example Nagios[6]) that can be managed directly by the infrastructure owners. Data are then passed, through an *NGSI Adapter*, to the *FIWARE Context Broker GE* [28] and then to the *FIWARE Big Data GE* [29] where data can be elaborated, processed and stored. The local (i.e. installed on a slave node) Big Data GE communicates with an instance of the same GE in the master node in order to maintain aggregated data at the federation level. Having a local Monitoring System can also allow infrastructure owners to fine tune the data that can be published outside the infrastructure keeping "confidential" data private.

[6]Nagios: - http://www.nagios.org/

3.1.3 Component enabling security functionalities

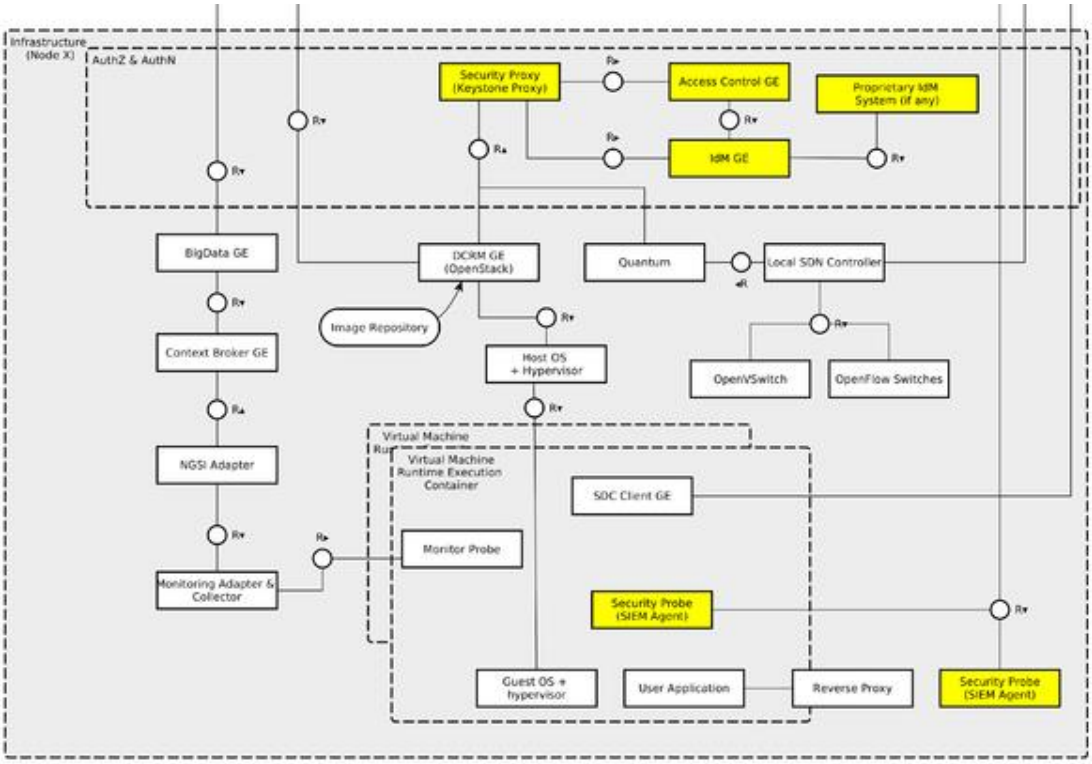

Figure 4: Components enabling Security Functionalities ver. 2.0

These components (in yellow in the previous figure) enable the security functionalities. The *FIWARE IdM GE* [17] together with the *Security Proxy (Keystone Proxy)* [15], and *Access Control GE* [23] provide both authentication and authorization services for each node. In the future we foresee the integration with *Proprietary IdM Systems* (i.e. existing systems installed on the infrastructure and managed by the infrastructure owner), if present on the nodes, using protocols like SAML and SCIM. In this way the infrastructure owner can keep control of the security and in particular of the identity management. *Security Probes (SIEM Agent)* are responsible to collect security monitoring data and send them to the master node (see following section).

The distribution of the security components (authentication and authorization) on each node of the FIWARE Lab federation will be provisioned in high availability so as to avoid any single point of failure.

3.2 Architecture of a XIFI Master Node

In addition to the components described in the previous section, a master node comprises the following functional groups:

1. User oriented services and tools (no changes from the D1.3)
2. Services and tools supporting the setup, deployment and operation of the Federation (slightly changed from the D1.3)
3. Federation Security tools (no changes from the D1.3)

3.2.1 User oriented services and tools

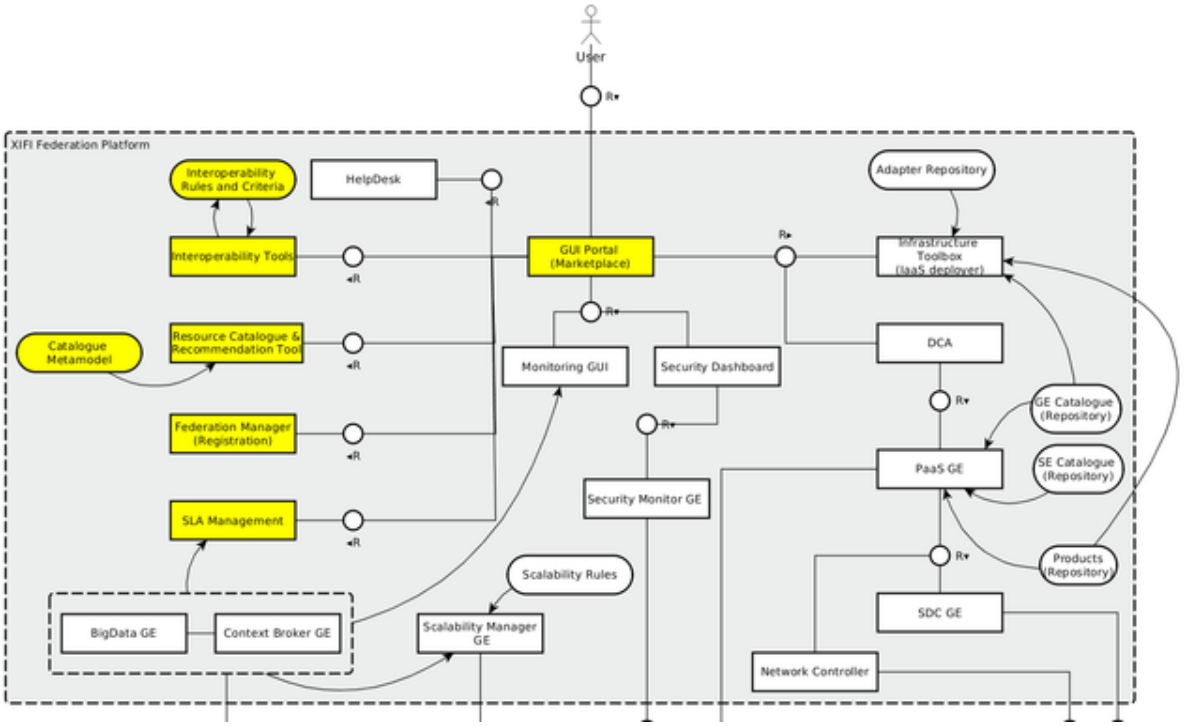

Figure 5: User Oriented Services and Tools ver 2.0

These services and tools (in yellow in the previous figure) implement a federation view of all the facilities offered by XIFI. *Resource Catalogue and Recommendation Tool* is oriented to find the right services offered by the federation; *Interoperability Tools* can verify the interoperability and compatibility of developed software with FIWARE GEs based on some rules; *SLA Management* handles the SLA negotiation; *Federation Manager* governs the registration of a new infrastructure to the FIWARE Lab (XIFI) federation. Finally *GUI Portal (Marketplace)* provides a single entry point (portal) and a graphical user interface for all these tools offering a sort of marketplace for all the services provided by XIFI.

3.2.2 Services and tools supporting the setup, deployment and operation of the Federation

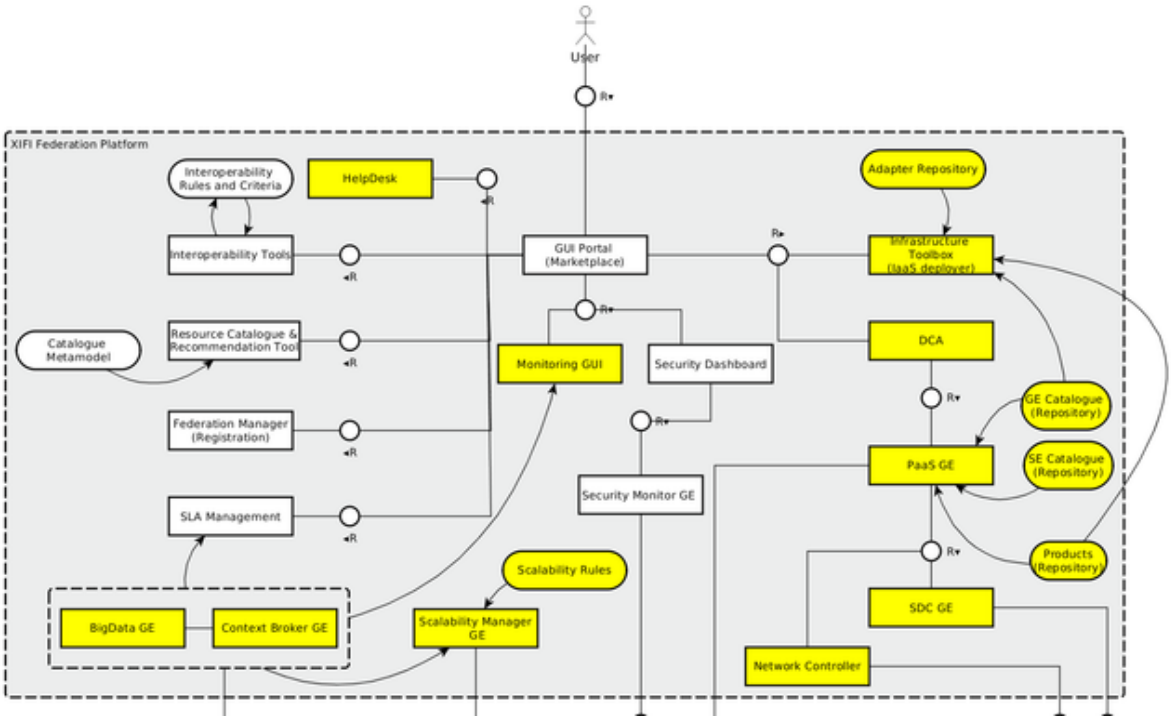

Figure 6: Service and tools supporting the Federation ver. 2.0

This set of tools (in yellow in the previous figure) offers all the functionalities to deploy the software needed to install a FIWARE Lab (XIFI) node starting from the bare metal and to operate a node during its activities.

The *Infrastructure Toolbox* aims at providing an automated installation of the IaaS Management System (OpenStack, DCRM and the Local SDN Controller), the components enabling monitoring functionalities and the component enabling security functionalities. *DCA (Deployment and Configuration Adapter)*, *FIWARE PaaS Manager GE* [12], *SDC Manager GE* and *Network Controller* provide functionalities for deployment of the GEs, SEs and third parties products on the different nodes of the federation and to set up the network connectivity among different VMs. In particular DCA enhances the functionalities offered by the PaaS Manager GE providing multi-node deployment, check of resource availability before the deployment process starts and a persistent configuration management database of the deployed GEs that can be consumed by other XIFI components like the "user oriented services and tools" (see previous section) and the Monitoring system components.

The *FIWARE Policy Manager (Scalability Manager)* implements elasticity and scalability rules. It implements a set of rules which are defined by the administrator of the tenant and allow to take decisions based on the monitoring data collected from the XIFI Monitoring System or any other information source.

The *FIWARE Big Data GE* [29] and *FIWARE Context Broker GE* [28] offer monitoring functionalities at the federation layer: they have been depicted also here (not only in the generic node architecture) in order to highlight that on the master nodes (at federation level) the monitoring data will be stored and aggregated following different perspectives (e.g. average on time, average on resources belonging to a node, etc.).

The *Monitoring GUI*, integrated into the *GUI Portal* component, provides a user interface on the monitoring data.

Finally, *Help Desk* is a problem tracking system that implements a workflow defined for processing user requests and providing user support (this is offered in collaboration with the FI-PPP program using a commercial tool – Jira[7]).

3.2.3 Federation Security tools

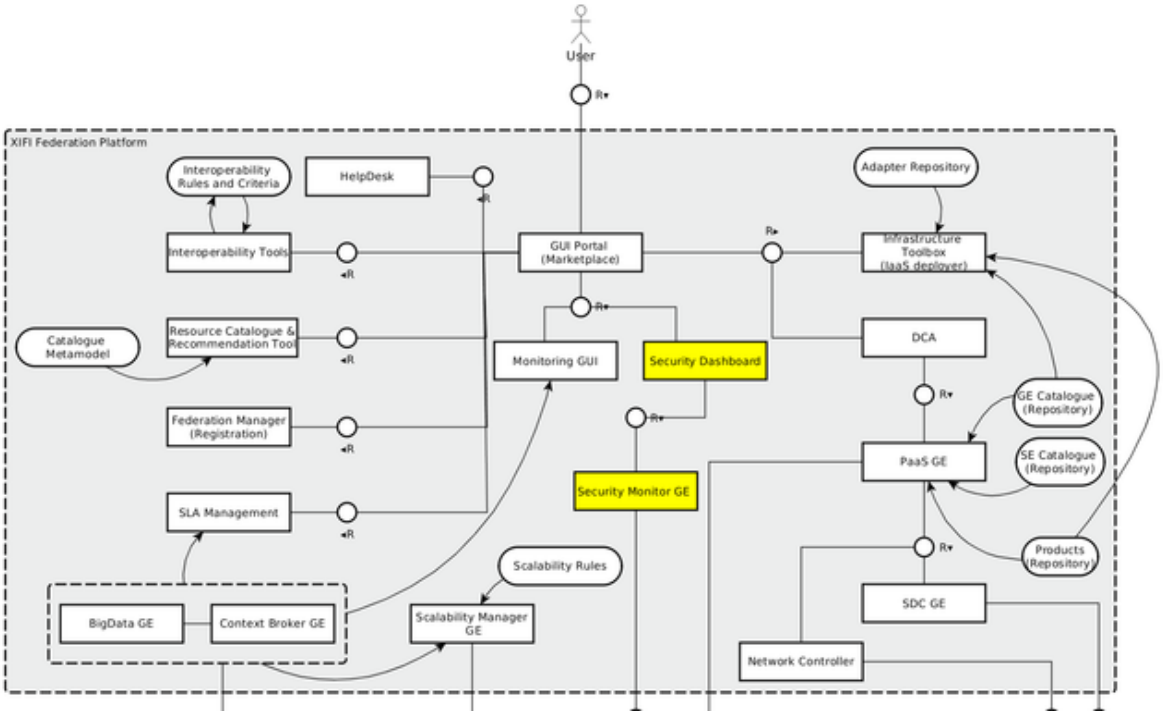

Figure 7: Federation Security tools ver. 2.0

The security system (in yellow in the previous figure), part of the master node, comprises the *FIWARE Security Monitoring GE* [25] that gathers security monitoring data from the remote probes and from proprietary security systems (if any) and the *FIWARE Security Dashboard* [26], integrated into the *GUI Portal* component, that provides a graphical user interface to show security monitoring data and alerts users in the case of security problems.

3.3 Node internetworking

Nothing has been changed with respect to what reported in section 3.3 of D1.3.

3.4 Summary of architectural decisions and rationale

Nothing has been changed with respect to what reported in section 3.4 of D1.3.

[7] https://www.atlassian.com/software/jira

3.5 Changes from previous version of the architecture (D1.3)

Modifications made to this version of the federation architecture with respect to the previous version described in D1.3 are minimal and do not affect the overall architectural framework. In the following diagram the modifications are highlighted in yellow.

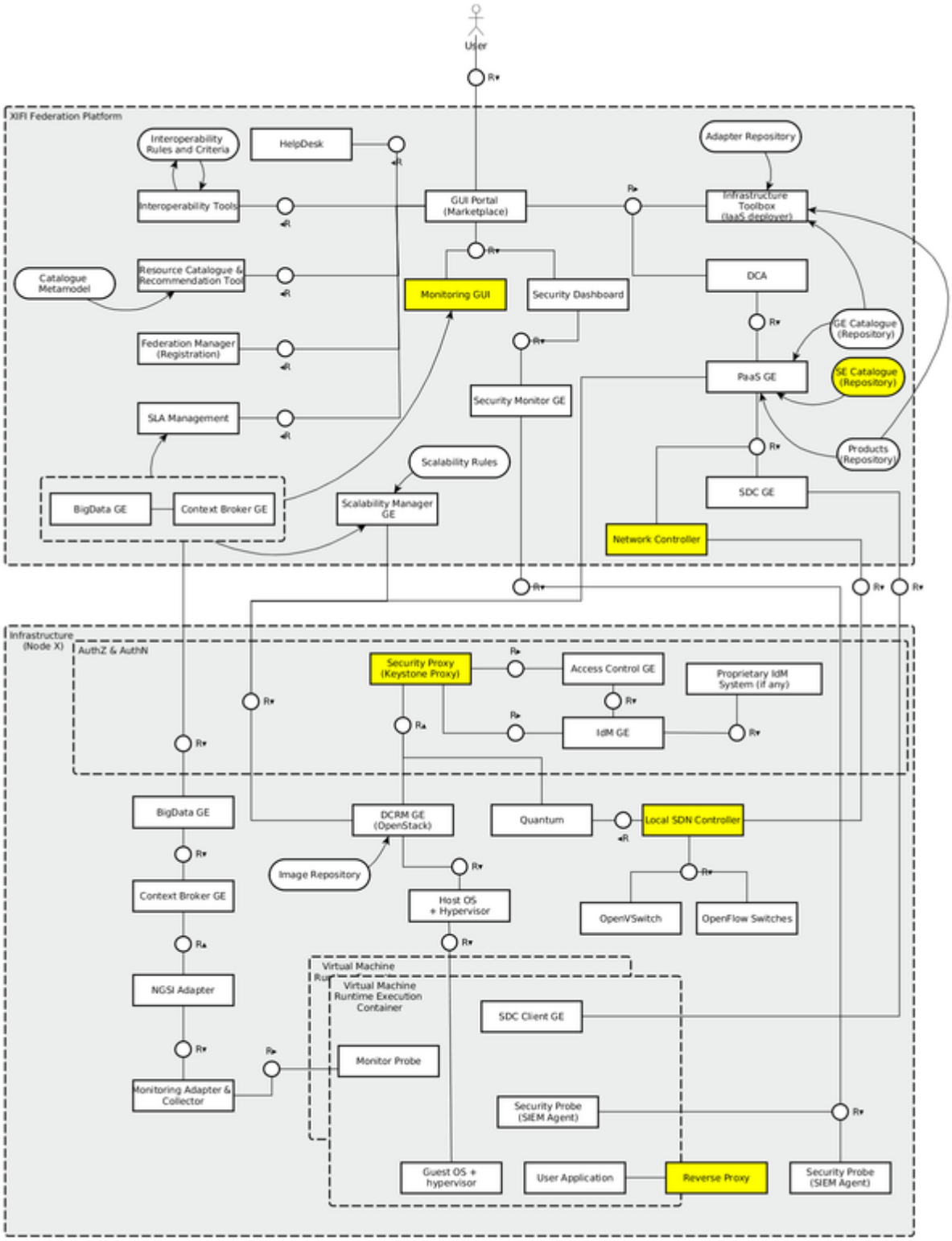

Figure 8: Modification to previous version of the architecture (ver. 1.0 vs ver. 2.0)

As already mentioned in the previous sections, the architecture underwent only few modifications in this version:

1. **Components enabling cloud computing**: here the network controller subsystem has been better described taking into account the changes in the requirements (REQ-4.x - see D1.4) and taking advantage of the work made in task T3.1. The old *Network Controller* component has been split into a local component (called *Local SDN Controller*) and a "master" (i.e. installed only on the master node) *Network Controller* that orchestrates the different local controllers in order to offer inter-node connectivity through the MD-VPN and support to QoS. Moreover a *Reverse Proxy* component has been explicitly added in order to overcome the problem of lacking of public IP addresses.
2. **Component enabling security functionalities**: here just the name of the *Security Proxy* component has been changed in *Security Proxy (Keystone Proxy)* in order to clarify that this component is an implementation, offered by FIWARE, of the OpenStack Keystone component.
3. **Service and tools supporting the setup, deployment and operation of the Federation**: this set of components has been modified under three perspectives:
    - Introduction of the new "master" *Network Controller* (see first point in this list)
    - A *Specific Enabler (SE) Catalogue* has been added in order to take in account the possible integration of FIWARE Lab with UC Projects Catalogues as requested by REQ-2.2 (see D1.4).
    - Explicitly added the *Monitoring GUI* component, integrated into the *GUI Portal* component, which offers a user interface on the monitoring data.

3.6 Mapping of requirements on architectural components

This section provided an update of the mapping among use case scenarios, high level requirements and architectural components. For the list of Use Case Scenario, see D1.1.1; for the list of high level requirements see D1.4. Up to date mapping can be found on XIFI Redmine[8] together with a mapping of the high level requirements on detailed ones (for each component).

Use Case Scenario	Requirement	Architectural Component	Comment
UC-1	REQ-1, REQ-2, REQ-2.1, REQ-2.2	Infrastructure Toolbox, Federation Manager, Resource Catalogue&Recommender	OpenStack, DCRM GE, Network Controller subsystem, Monitoring subsystem and Security components will be deployed and configured by the Infrastructure Toolbox component. Federation Manager governs the registration process. Resource Catalogue is used for registering conventional and non-conventional services.

[8] XIFI redmine: redmine.fi-XIFI.eu

Use Case Scenario	Requirement	Architectural Component	Comment
UC-2	REQ-3, REQ-3.1, REQ-3.2, REQ-3.3, REQ-3.4, REQ-4, REQ-4.1, REQ-4.2, REQ-5, REQ-6, REQ-7, REQ-8, REQ-12	GUI Portal, Resource Catalogue&Recommender, Interoperability Tools, DCA, Paas Manager GE, SDC Manager GE, DCRM GE, Network Controller subsystem, Policy Manager GE, IdMGE, Access Control GE, Security Proxy (Keystone Proxy), Reverse Proxy	Through the usage of GUI Portal, Resource Catalogue& Recommender, IdM GE, Access Control GE, DCA and the PaaS and SDC Manager GE, a developer can set up his development platform and deploy it on the selected regions. Moreover it can be appropriately configured in terms of network connectivity and scalability/elasticity. In case of special or non-conventional services, XIFI does not provide at the moment any specific tool, but in the future could be feasible to integrate in the XIFI platform tools provided by infrastructure owners offering non-conventional services management.
UC-3	REQ-13, REQ-14	Infrastructure Toolbox and, if requested, Federation Manager, IdM GE, Monitoring Adapter and Collector, Context Broker GE, Big Data GE etc.	A user, using the Infrastructure Toolbox can install over the bare metal the basic cloud infrastructure for his private cloud. Moreover, if needed, other services like IdM GE (and other security components) and Monitoring subsystem can be installed. The user can decide the level of integration between his/her infrastructure and the rest of XIFI federation.
UC-4	REQ-10, REQ-10.1	HelpDesk	User can submit a support request through the Help Desk component.
UC-5	REQ-9, REQ-9.1, REQ-9.2, REQ-9.3, REQ-9.4, REQ-9.5, REQ-11, REQ-15	Context Broker GE, Big Data GE, SLA Management, Monitoring Adapter & Collector, Security Monitoring GE, Security Dashboard, Network Controller subsystem, DCRM GE, Security Proxy (Keystone Proxy)	Monitoring data is collected by Monitoring Adapter & Collector, sent to Context Broker GE, elaborated and stored via Big Data GE and finally provided to SLA Management and Monitoring GUI (integrated into the GUI Portal). Security Monitoring is made through Security probes, Security Monitoring GE and Security Dashboard. Through Network Controller subsystem, DCRM GE and Security Proxy, administrators can configure their infrastructures in terms of network, quotas, security rules etc.

Table 2: Mapping of requirements on architectural components

3.7 Change Management Process

This section explains how the updating of the architecture is handled in XIFI project.

With this version the architecture has reached a quite stable status and in the future we do not foresee significant changes to it. The work made on requirements has provided a set that is stable now (see section "Conclusions" in D1.4) and consequently also the architecture should remain stable in the next months of the project. Moreover the architectural framework and the rationale of the architectural choices made in the past has proved to be solid and allowed to integrate in the first version of the architecture (defined before M9) almost all the requirements elicited in the second iteration (from M9 to M15 of the project).

Nevertheless some changes could be requested again in the future. For this reason the *change management* process followed to update the architecture is made explicit hereunder in order to inform all the stakeholders of the procedure that governs architectural changes. This section highlights the way XIFI is handling the change management process inside its lifecycle.

The following figure presents a classical change management software system graph where, at a certain point in time, a stable release is forked into a new branch.

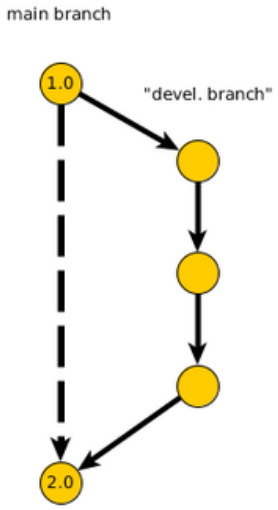

Figure 9: Change management system

We adopted a similar approach to manage the updates to the XIFI architecture. The process that conducts to a change in the architecture is depicted in the following diagram.

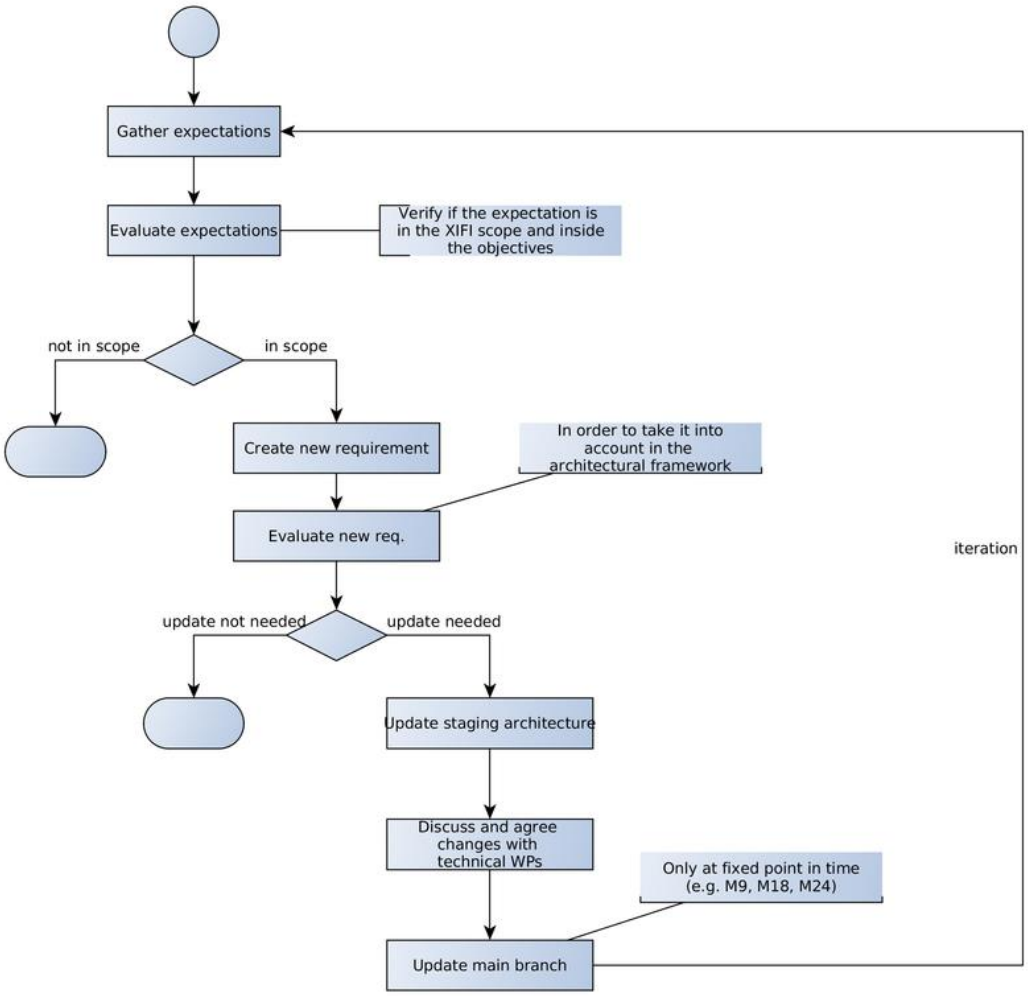

Figure 10: Change management workflow

1. After a new stable release is issued (generally corresponding to the dates of the official architectural deliverables in WP1), the "development" of a new version of the architecture continues internally in WP1 analysing the input (expectations, requirements etc.) coming from the different stakeholders (internal and external ones).
2. Requirement phase (in T1.2): expectations are analysed and, if approved (i.e. if they are conform to the vision and objectives of XIFI), they are converted into high level requirements.
3. Architecture phase (in T1.3): new requirements are analysed and, if approved (here the importance for the stakeholders, the impact of each requirement on the architecture and possible issues on the backward compatibility are taken into consideration), converted into a change to the architecture in the "staging area".
4. The change is discussed and agreed with the other technical work packages and it is then promoted as an "official" change to the architecture.
5. At specific point in time (i.e. when a milestone or a deliverable is scheduled) the added changes are merged into a new stable version of the architecture.
6. The iteration starts again from the beginning.

3.8 Usage of GEs for managing federation

The following table shows information about the Generic Enablers used in the set up of the FIWARE Lab (XIFI) federation, highlighting their purpose and role in XIFI and possible changes made (or requested by XIFI project to FIWARE) in order to satisfy the needs of the federation.

GE Name	Purpose	Used in XIFI Architecture
IdM GE [17]	The purposes are mainly three: • supporting the distribution in High Availability of IdM GE • access to federated resources • support Identity Federation standards	This GE is used in XIFI with some changes: • The High Availability support is implemented through proper deployment architecture that allows multiple IdM instances to share the same database cluster and the same virtual IP. The database cluster, compatible with the KeyrockIdMGEi [18] architecture is based on Galera, a wsrep API implementation for MySQL. The Virtual IP service is implemented using *keepalived*, a load balancing and high-availability IP service for Linux. • The support for Identity Federation is achieved by integrating in the IdM the SAML 2.0[9] support to act both as a Service Provider and an Identity Provider. Implementation is based on the integration of Shibboleth[10] modules.
Respositoy GE [19]	The purposes are mainly the following: • Publish the description of various aspects of the service according to a uniform description language (USDL[11]). • Manage the complete resource lifecycle and store its resources in arbitrary formats. • Provide a consistent uniform API to USDL service descriptions and associated media files for applications of the business framework. • Allow to refer to entities of the service description via the resource URL. So, it is already well prepared to allow the distribution of service descriptions all over the Internet.	This GE has been used directly, in the XIFI Architecture, without introducing any changes. • It is used for storing and describing both the specific enablers (SE) and non-conventional services. It allows defining technical specification as well as business specification. • The description follows the standard USDL, which is a uniform description language to describe services. • It exposes a RESTFul API in order to manage the repository lifecycle of the XIFI resources. • It allows defining resources and introducing ontologies of different domains.

[9] SAML 2.0: https://www.oasis-open.org/standards#samlv2.0

[10] Shiboleth: https://shibboleth.net/

[11] USDL: http://www.internet-of-services.com/index.php?id=264

GE Name	Purpose	Used in XIFI Architecture
Wstore GE [20]	The purposes are mainly the following: • Manage the resources/services lifecycle. It supports the publication of new offerings, manages offering payment, and provides access to all purchased services. • Manage the complete business lifecycle (future business requirements about the billing, payment...) • Manage the rating and recommendation of the services. • Integrated support for the Repository GE [19], which stores the services descriptions, MarketPlace GE [21] and Revenue Sharing GEs [22]. .	This GE is used in XIFI introducing some changes and competitive advantages: • It can operate in a federated environment, including functionalities as the visualization of the different nodes where the same services have been deployed. • Integrated with the FIWARE Catalogue containing the GE descriptions. • Definition and offering of non-conventional services and Specific Enablers. • Searching of heterogeneous resources/services (GE/SE and non-conventional services) in a transparent way through the different federated nodes. • Provide details of the status of the services instances and their evolution as the monitoring data and the number of instances per node.
Access Control GE [23]	The purpose is to manage and enforce XACML-standard-compliant[12] attribute-based authorization on access to security-sensitive resources (services or data handled by them); "attribute-based" means based on attributes such as the requester's attributes, requested action, attributes of the requested resource itself, and possibly environment attributes.	The GE is used in XIFI in various ways: • By the Security Proxy [24] for requesting authorization decisions, to control access to federated resources, in particular cloud resources. • By Use Case projects where such advanced attribute-based access control is required to protect their custom applications and data, preferably when providing RESTful APIs. • By the Security Monitoring GE [25] with help of Security Probes, and as a result, by the Security Dashboard to display access control logs for accounting and detecting non-compliant behaviour. The GE implementation has been enhanced for XIFI to provide multi-node High Availability.
Security Monitoring GE [25]	The purpose of this GE is to collect security events happening on security-sensitive components of the federation, thanks to the security probes installed in the slave nodes, and to apply correlation rules to these events in order to detect violations of the compliance to the security policy; in case of violations, it will raise alerts and notify the security operators through the Security Dashboard [26] integrated in XIFI Portal.	This GE is used in XIFI to monitor security events happening on the critical components, with the help of Security Probes deployed there and sending normalized events to the centralized component of the GE in the master nodes. The GE implementation has been customized for XIFI in order to integrate the user management with the IdM GE and to include the accountability events generated throughout the federation by the Access Control GE.

[12]XACML-standard-compliant: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

GE Name	Purpose	Used in XIFI Architecture
DCRM GE [10]	This GE is part of the FIWARE Cloud Hosting Chapter and offers some enhanced functionalities over OpenStack.	This GE is used in XIFI for provisioning VMs, storage and network resources. It is used "as is" without any changes in XIFI.
Paas Manager GE [12]	This GE is part of the FIWARE Cloud Hosting Chapter and offers PaaS Manager functionalities.	This GE is used in XIFI for deploying software stacks that build application environments based on blueprint templates. It has been enhanced in XIFI adding multi-region support.
SDC Manager GE [11]	This GE is part of the FIWARE Cloud Hosting Chapter and offers automated deployment of software products	This GE is used in XIFI for providing automatic deployment, through recipes, of software products. Some improvements have been developed in order to manage the auto provision of software independently of the version and type of the operating system.
Cloud Portal GE [27]	This GE is part of the FIWARE Cloud Hosting Chapter and offers a graphical user interface on the IaaS management layer (OpenStack)	This GE is used in XIFI for providing user access to the IaaS management functionalities. It has been changed in XIFI in order to add multi-region support and the possibility to register new regions.
Context Broker GE [28]	This GE offers Publish/Subscribe functionalities based on NGSI standard.	It has been used in XIFI in order to handle subscription of different monitoring data sources and for publishing this data to the interested components. It is used "as is" without any changes.
Big Data GE [29]	This GE offer big data analytics support and is based on Hadoop and HDFS.	This GE has been used in XIFI in order to analyse and elaborate the monitoring data collected from the different sources. Thanks to this GE, the raw data collected is aggregated using different operators (average, max, min) on different dimensions (time vs space/location of the data). This GE has been modified in XIFI in order to integrate the latest stable Hadoop version (at the time of the development).

Table 3: Generic Enablers usage in managing XIFI federation

4 PRELIMINARY VALIDATION OF XIFI FEDERATED ARCHITECTURE

4.1 Infrastructure Owner experience

As preliminary validation of the XIFI architecture, a questionnaire has been proposed to the 5 original infrastructure owners. The questionnaire was the following:

1. Is the architecture as defined giving you everything you need to cover federation functionalities?
2. What are the main positive points introduced by the XIFI architecture from your point of view?
3. Main points to be investigated in future version of the architecture.
4. Were there any surprises?

Below the answers are reported:

4.1.1 Lannion node

1. Is the architecture as defined giving you everything you need to cover federation functionalities?

The actual architecture definition provides the basic environment to create the FIWARE Lab federation allowing having a single entry point for user to access the federation resources.

2. What are the main positive points introduced by the XIFI architecture from your point of view?

Several key points allowing an infrastructure owner to monitor the infrastructure within the federation have been identified (since the beginning).

- Monitoring of the cloud components, the virtual machines and the network status
- Monitoring of SLA
- Security Monitoring

3. Main points to be investigated in future version of the architecture

- It would be nice to improve the way the GEi repository (where the GEi images are located) is maintained across the nodes. At the time of the writing GEis are available on a server and must be uploaded manually on each region, and there is not consistency checking of the repository. Even further, is not clear what GEs are optional/mandatory on the regions
- Having the possibility to share flavours across the node could be interesting. A good idea may be to create a central catalogue in which a flavour can be selected and pushed to a region

4. Were there any surprises?

Not all tools used in FIWARE Lab allow the management of roles and in particular the role of region administrator. This is critical in order to ensure the capacity of the infrastructure owner to administrate their infrastructure and grant the access control to its resources.

4.1.2 Spain node

1. Is the architecture as defined giving you everything you need to cover federation functionalities?

Currently, the architecture allows us to provide all functionalities related to the deployment of resources (e.g. servers) in different regions. It also allows to distribute the resources usage between different regions depending of whatever requirements that we might find in our business (legal, technical, security or so on). In consequence these functionalities introduce lots of opportunities for a business deployment.

2. What are the main positive points introduced by the XIFI architecture from your point of view?

The more important aspect introduced by XIFI architecture is the possibility to deal with the federation. Another positive aspect is the management of data security in a federation environment.

3. Main points to be investigated in future version of the architecture

Currently, we have defined a catalogue of the GEIs that can be used in the node. It provides very useful information about the component together with where it is installed. Nevertheless, the lack of a repository of those components introduces a serious problem when we try to deploy a new node or we want to update a previous version of the components. In consequence, we need lots of time to proceed with these operations. The association of the catalogue with a binary repository will provide an integrated place from which we can obtain the different components in an easy way. This functionality will also include the development of the installation, un-installation and updating scripts, mainly using chef or puppet, of those components.

Besides, some investigation should be done in order to know the way to update the OpenStack. Currently, it is not easy to update from a previous version to another. In consequence, it is a hand-made operation and it usually involves lots of time of the cloud administrators. Any application developed in this line will help us to provide a solution that could be extended beyond the life of the project.

Despite the introduction of the federation functionalities, currently, an integrated monitoring of available resources including regions health status is not provided. This information could be useful in order to know where we want to deploy specific resources. Currently, we have developed a monitoring component which is integrated with the Cloud Portal, but more information about the overall health of the region not only if it is up and running but also the percentage of memory, cpu used or bandwidth latency (already provided) with an evolution in time, could help us to decide where to use resources.

4. Were there any surprises?

The utilization of the monitoring components related to the notification of the infographics status of the different nodes is focused in a standard installation of OpenStack. This is not the case of the Spain node, in which we have the different OpenStack services deployed in different physical machines. It was a serious problem that we had to resolve by developing our own probes for some of the OpenStack services. Currently the distribution of the OpenStack services is more or less standard but the situation of Spain node could be repeated in the future and it should be taken into consideration in order to resolve future issues.

4.1.3 Berlin node

1. Is the architecture as defined giving you everything you need to cover federation functionality?

From the architectural perspective, XIFI implements a one stop shop model federation through a central authority (i.e. the master node). User authentication and identities are managed by a single master node with all the well-known positive and negative aspects this choice carries with it. For the Berlin node some missing functionality caused node-local issues, in particular regarding the loss of information and control about a node's users and about local access control and surveillance of "non-conventional" (mostly expensive and therefore shared) services.

These issues are addressed by the project and solutions may be available further towards the end of the project. In consequence, the Berlin node had to provide additional functionality on the node level:

- To ensure node privacy in federation monitoring, due to legal and business implications, data has to be locally inspected and potentially filtered prior to sending it to the presentation and use. This required to set-up a secondary node monitoring.
- The XIFI architecture makes available all sets of GEs as defined by FIWARE through a replicated repository available across individual nodes but does not support or "market" customized versions -- they are considered "SEs" in general. For example, the Berlin node had to provide a customized S3C GE to access the remote wireless infrastructure.
- Non-conventional resources are considered integral part of the offer of XIFI, but their nature is considered local. There are therefore few dedicated services to them (mainly advertisement in the marketplace) and their management including SLA and access control is therefore left to the node requiring the Berlin node to handle a simple secondary identity management.

2. What are the main positive points introduced by the XIFI architecture from your point of view?

The node set-up has been greatly simplified by the ITBox. When considering this tool as a "machine-readable architecture description", it is the most convenient way to set-up a compliant node configuration. Nevertheless it is recognized that an ITBox based node set-up usually is not covering all node requirements. It rather is the "least common denominator" providing a convenient starting point for node customization keeping compliance with the federation in the process.

3. What are the main points to be investigated in future version of the architecture?

The Berlin node suggests providing support for nodes to locally manage authentication and user identities, either for the node or for particular resources of the node.

For this an approach will be needed to maintain direct contact with the user in case the user opts to use node-specific services, which may include general consideration of a node-based "customer relationship management". From earlier experience with federated testbeds, it is well known, that node-local resources (in particular "non-conventional" resources) need direct contact between user and node to bilaterally clarify details of the request to support and to negotiate on configurations, usage periods and support needed. Current experience seems to disclose that these considerations may apply to any type of scarce resources such as public IP addresses of a node made available to the user.

Support of service level agreements on "non-conventional" resources and services should also be added to future architecture. It would be helpful to see some architectural support for interfaces between XIFI and "non-conventional" policies and measurements (in particular suitable data formats).

As for the GE repository the suggested enhancement is to update the architecture allowing a central repository, a node local repository cache and a node local (secondary) repository keeping customized GEs and non-public SEs for convenience. This would better integrate with a generic software deployment and maintenance strategy as well as with the integrated development chain, first deploying to a node for testing and then publishing through the master node towards all nodes.

4. Were there any surprises?

Unfortunately, and mostly for non-technical reasons (i.e. legal one), the current deployment is different from the reference architecture and this caused many of the issues previously mentioned.

4.1.4 Waterford node

1. Is the architecture as defined giving you everything you need to cover federation functionalities?

The identity management functionality architecture approach is valid and fulfils our needs from a security and integration point of view. We look forward to the third release of the identity management components with support for SAML which will allow us to integrate existing communities at our site. We also look forward to fine-grained capability management integrated with different role levels at federation level.

The federation monitoring solutions gather independent data from each site. This data is both pulled periodically and pushed when requested. We currently look forward to using the aggregated data in order to observe federation-wide technical issues which can be resolved by the community of operations people.

The approach to resource discovery is in line with leading edge work in the state of the art. As an infrastructure owner, we would like to be able to add resources to the federation as needed with a small amount of effort. The mix of centralized and distributed resource discovery appears to meet these needs.

For usage monitoring, the topics of resource usage life-cycle and traceability are important as they affect our operations on a daily basis. As our infrastructure is connected via our NREN (HEAnet), we must comply with their Acceptable Usage Policy (AUP). Resource Usage can provide us with some insight if there are potential issues in how resources are being used and traceability can help us take corrective action in situations where there has been a violation of the AUP. In a perfect world there would be no violations of the AUP.

2. What are the main positive points introduced by the XIFI architecture from your point of view?

The introduction of systems such as the ITBox and software repository helps easier maintenance of the running systems. The Big Data GE, Context Broker and resulting Monitoring dashboard can play an important role in running the infrastructure.

3. Main points to be investigated in future version of the architecture

In summary, resource usage from a central point of view is important for our operations team. The possibility of per tenant or per user accounting is also more important from an operations perspective proving its worth in assisting in security and preventing legal issues. The assertion of its lower importance due to only being used for billing (which XIFI is not doing yet) may not be as accurate as originally thought.

The distribution of updates to the software repository for operation teams and the GE repository may become important as the scale of the federation and its services grows. The software repository and GE repository may share a lot of characteristics. One could suggest using an eventual consistency model for updates to the content of both repositories and geo-location filter for access to a more local replica while still maintaining an apparently centralized service in both cases. The impact would be additional resource requirements at nodes that are willing to participate with the benefits being an even spread of network resources being used for distributing updates to both the catalogue and the repository.

4. Were there any surprises?

The change management on the architecture and resulting subsystem release management could be improved. At present there are many threads of communication around the updates to the architecture and resulting subsystems. It may be beneficial to coordinate this communication in a similar fashion to how Linux distributions are released. This means that not only is there scope to release updates during a given distro support period but the operations person can be confident that the changed subsystems have been tested against the other subsystem components.

4.1.5 Trento node

1. Is the architecture as defined giving you everything you need to cover federation functionalities?

The actual architecture definition provides the basic environment to create the FIWARE Lab federation

2. What are the main positive points introduced by the XIFI architecture from your point of view?

- All the monitoring tools are a good to maintain the node up and running.
- The installation of the OpenStack has been simplified by ITBox.

3. Main points to be investigated in future version of the architecture

For an infrastructure owner is important to have a general overview of the node resources, but at the moment is not so easy to get that in particular on the web interface.

4. Were there any surprise?

Currently we are using OpenStack Grizzly, and that version has many bugs. A lot of time was spent to fix the cluster.

4.1.6 Questionnaire analysis

Analysing infrastructure owners answers it is immediately clear across them a consistent view of the FIWARE Lab federation. Everyone sees the basic functionality well covered by XIFI and sees already existing and potential benefit of being part of the federation. ITBox has got a general consensus as a very useful tool for managing the deployment phase.

Looking at future development it is also possible to identify a common direction towards an improved management of GEi repository and improved process for software updating. Some other aspects like accounting and some advanced monitoring functionalities are also present as nice to have in future releases. A fine tuning between what is managed at central or local level (e.g. user management or

monitoring aspects) is also welcome by several owners. Last but not least more emphasis on the management of non-conventional services is expected.

Some of the comments are about the architecture but most part of them are already addressed by the architecture and are more related to improving exiting components inside the architecture and/or to improving the current deployment. WP1 will take care of these comments either considering them for the future development of the architecture (if not yet covered) or passing them to the relevant component owners so they can evaluate how to address them or finally to push for a deployment that is compliant with the federation architecture.

Since the XIFI project is not too far from its end, this feedback could be useful also for projects or initiatives that will guarantee the future of FIWARE Lab (e.g. FI-CORE).

4.2 End users and Use Case experiences in using the federated architecture

This section reports the experience of some UC project, like FITMAN[13], in using the FIWARE Lab federation. Some issues derived from specific aspects including the maturity of the XIFI federation. Tackling them, XIFI proved to be able to support experimental trials. The experience and lessons learnt gained in interacting with FITMAN will be applied to future interactions with other Use Case projects.

4.2.1 Current status of deployment of the architecture from an end-user point of view

The FIWARE Lab federated services are available to the end users since April 2014. At the moment of writing the federation is composed by six nodes: Spain, Trento, Lannion, Berlin, Waterford and Prague (available since June 2014). Using the FIWARE Lab Cloud Portal[14] it is possible to access the federation infrastructure: the federated services span from virtual machine deployment, to cloud storage, to set-up of application environment, to the monitoring of the application environment. These services are offered through the following user oriented tools:

- Cloud Portal [27]
- Monitoring [30]
- Resource Catalogue & Recommender [14]
- PaaS Manager [12]

all accessible through the FIWARE Lab GUI Portal.

In order to use the XIFI facilities, end user has to sign up through the FIWARE Lab Account Portal[15] and to accept the Terms and Conditions[16]. The FIWARE Lab Portal is the main access point to FIWARE Lab federation and offers the possibility to configure organizations in order to manage and group together users for specific projects (in this way different users can act as a single tenant). Moreover it is possible to benefit from the so-called IaaS layer: deploying virtual machines (VMs), including storage, and defining the security features and networking of a virtualized environment. These services are offered to the end users through the FIWARE Lab Cloud Portal leveraging on the OpenStack cloud management platform. Thanks to the federated nature of XIFI, it is also possible to select the node (the so-called "region"), where to deploy an environment.

[13]Fitman project: http://www.fitman-fi.eu/project-overview

[14]FI-Lab Portal (available at https://cloud.lab.FI-WARE.org)

[15]https://account.lab.FI-WARE.org/users/sign_up

[16]http://forge.FI-WARE.org/plugins/mediawiki/wiki/FI-WARE/index.php/FI-LAB_Terms_and_Conditions

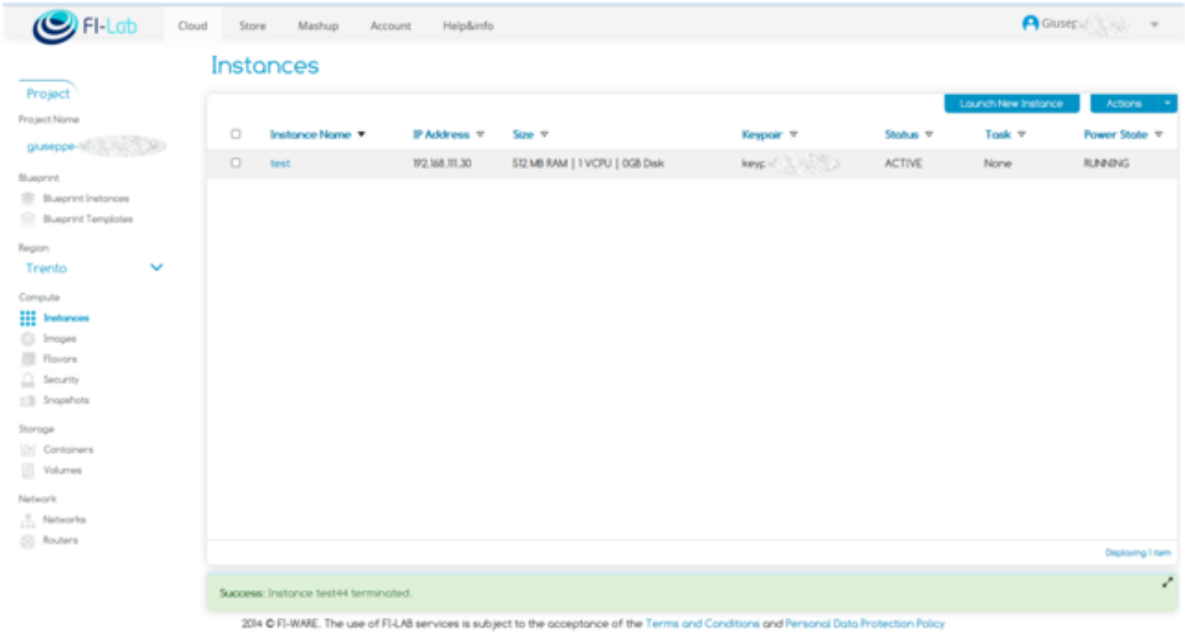

Figure 11: FIWARE Lab Cloud Portal snapshot

End users can also take advantage of the related API (instead of using the FIWARE Lab Cloud Portal) to directly manage the available cloud resources.

Each tenant has a quota of cloud resources, such as:

- instances: 3 (number of instances allowed per tenant)
- floating IPs: 1 (number of floating IPs allowed per tenant)
- cores: 6 (number of instance cores allowed per tenant)
- volumes: 10 (number of volumes allowed per tenant)
- gigabytes: 1000 (number of volume gigabytes allowed per tenant)
- ram: 2048 (megabytes of instance ram allowed per tenant)

Along with the cloud computing IaaS environment, end user can leverage on the other specific XIFI services:

- Users can monitor the status of running VMs, such as available disk, CPU usage and RAM usage.
- By means of the Recommender & Resource Catalogue users are able to select and "evaluate" the Generic Enablers (GEs) provided in the FIWARE Catalogue.
- Through the PaaS Manager, users can deploy multi-tier application environment.

FI-PPP Use Cases projects, like FITMAN (but also FINESCE), have used these services so far with positive results: they were able to deploy their environments on FIWARE Lab, to play with them and to expose them to their customers. Nevertheless it must be mentioned that these projects requested the usage of an ad-hoc quota, so the Use Cases projects asked to the Infrastructure Owners to improve the resources assigned through the FIWARE Lab Help Desk tool: for instance FITMAN UC needed 4 virtual machines with 6 VCPUs, 6144MB of Memory and one public IP.

4.2.2 Evaluation of the current architecture from and end-user point of view

The current XIFI architecture fulfils the end-user requirements but in some cases the current available and implemented functionalities and services are not yet completed and can be improved. There are

some limitations that have been raised during the last months. Some of these limitations are related to the FIWARE Lab Cloud Portal and can be easily solved: for instance users are not able to deploy their own images or download the image files through the API, but they have to ask to Infrastructure Owner to perform these actions. Such kind of issues is well known and a solution will be available as soon as they will be integrated into the Cloud Portal.

The main limitation, raised by end-users so far, is related to the availability of a reasonable number to public IPs. Considering as an example the mentioned FITMAN Use Case project, it is using only one public IP: this means that only one instance can be accessed via SSH at a given point in time. The XIFI architecture foresees the usage of a "reverse proxy" for accessing via HTTP the user services and for limiting in this way the request of public IPs. For other protocols, like SSH, XIFI should provide other solutions (proxies) in order to guarantee the availability of public IPs. An alternative could be to adopt IPv6.

Another aspect to take into account is the integration of services through the whole federation. Some advanced features are in roadmap but not yet completely available such as the multi-tier deployment on multiple regions (PaaS Manager) and QoS support (Network Controller). So at the moment these features cannot be evaluated from the end-user perspective.

5 CONCLUSIONS

The second version of the XIFI architecture has been presented in this deliverable together with a description of the evolution from the previous version. As highlighted in this document, changes to the architecture are limited and not relevant for the whole framework: this means that the current version of the architecture can be considered stable. Nevertheless a change management process for updating the architecture has also been presented in order to control the impact of future modifications.

A validation process of the architecture has been carried out on different perspectives:

- Check of the application of the XIFI design principles defined in D1.1.1 (taking advantage of the results of milestone MS13)
- Interview of five infrastructure owners on their view of the architecture
- Report on the experience gained in interacting with the UC projects.

The current architecture has therefore proved to be suitable to support large scale trials and to address the requirements issued by the relevant XIFI stakeholders. Nevertheless some improvements to the architecture or, better, to its current deployment on the different nodes of the federation are requested by both infrastructure owners and UC projects. In particular some issues like:

- Handling public IP scarcity with some sort of proxies or with the adoption of IPv6,
- Strengthening the availability and the "security" of the federation providing a distributed deployment of key components like Security Proxy (Keystone Proxy) and IdM,
- Stabilising some functionalities and better automatising others, like PaaS Manager and software repositories

should be addressed in the future months. But all this cannot be done without the help and collaboration with external stakeholder like the FIWARE/FI-CORE project.

REFERENCES

[1] XIFI Deliverable (2013) D1.1.1 :XIFI Core Concepts, Requirements and Architecture Draft.*Available at:* https://bscw.fi-XIFI.eu/pub/bscw.cgi/d60668/XIFI-D1.1b-XIFI_core_concepts_requirements_and_architecture_draft.pdf

[2] National Institute of Science and Technology, The NIST Definition of Cloud Computing. Retrieved 30 July 2013 http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[3] HP, Seven design principles for building a converged cloud. Retrieved 30 July 2013. http://www8.hp.com/h20195/v2/GetDocument.aspx?docname=4AA4-6524ENW

[4] XIFI Deliverable (2013) D1.3: Federated Platform Architecture v1. *Available at:* https://bscw.fi-XIFI.eu/pub/bscw.cgi/d58827/XIFI-D1.3-Federated_Platform_Architecture_v1.pdf

[5] XIFI Deliverable (2014) D1.4: Analysis of UC, Infrastructures and Enablers v2. *Available at:* https://bscw.fi-XIFI.eu/pub/bscw.cgi/d82316/XIFI-D1.4-Analysis_of_UC_infrastructures_and_Enablers_v2.pdf

[6] XIFI Deliverable (2014) Milestone MS13: Evaluation of the XIFI Federated Architecture design principles.

[7] FedSM Consortium, (2012) D3.1: Business models for Federated e-Infrastructures. Retrieved 30 July 2013. http://www.fedsm.eu/sites/default/files/FedSM-D3.1-Business_models-v1.0.pdf

[8] XIFI Deliverable (2014) D8.2: Socio-economic factors and business models for XIFI federation members v1.

[9] OpenStack: http://www.openstack.org/

[10] FIWARE Consortium, Data Center Resource Management (DCRM). https://forge.FIWARE.eu/plugins/mediawiki/wiki/FIWARE/index.php/FIWARE.OpenSpecification.Cloud.DCRM

[11] FIWARE Consortium, Software Deployment and Configuration GE.https://forge.FIWARE.eu/plugins/mediawiki/wiki/FIWARE/index.php/FIWARE.OpenSpecification.Cloud.SDC

[12] FIWARE Consortium, Paas Manager GE. https://forge.FIWARE.eu/plugins/mediawiki/wiki/FIWARE/index.php/FIWARE.OpenSpecification.Cloud.PaaS

[13] FIWARE Consortium, SelfServiceInterfaces GE. https://forge.FIWARE.eu/plugins/mediawiki/wiki/FIWARE/index.php/FIWARE.OpenSpecification.Cloud.SelfServiceInterfaces

[14] XIFI Resource Catalogue and Recommender: http://wiki.fi-XIFI.eu/Public:Resource_Catalogue%26Recommender

[15] XIFI Security/Keystone Proxy: http://wiki.fi-XIFI.eu/Public:Security_Proxy

[16] A. Avizienis, J.C. Laprie, B. Randell, and C. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions On Dependable And Secure Computing, Vol. 1, No. 1, January-March 2004

[17] FIWARE Consortium, IdM GE. http://catalogue.FIWARE.org/enablers/identity-management-keyrock

[18] FIWARE Consortium, Keyrock Plugin. https://forge.FIWARE.org/plugins/mediawiki/wiki/FIWARE/index.php/Identity_Management_-_KeyRock_-_User_and_Programmers_Guide

[19] FIWARE Consortium, Repository GE. http://catalogue.FIWARE.org/enablers/repository-sap-ri

[20] FIWARE Consortium, WStore GE. http://catalogue.FIWARE.org/enablers/store-wstore

[21] FIWARE Consortium, MarketPlace GE. http://catalogue.FIWARE.eu/enablers/marketplace-sap-ri

[22] FIWARE Consortium, RevenueSharing GE. http://catalogue.FIWARE.org/enablers/revenue-settlement-and-sharing-system

[23] FIWARE Consortium, Access Control GE. http://catalogue.FIWARE.org/enablers/access-control-tha-implementation

[24] XIFI Consortium, Security Proxy GE. http://wiki.fi-XIFI.eu/Public:Security_Proxy

[25] XIFI Consortium, Security Monitoring GE. http://wiki.fi-XIFI.eu/XIFI:Wp4:Security_Monitoring_GE

[26] XIFI Consortium, Security Dashboard GE. http://wiki.fi-XIFI.eu/Public:Security_Dashboard

[27] FIWARE Consortium, Cloud Portal GE. http://catalogue.FIWARE.org/enablers/self-service-interfaces-cloud-portal-upm

[28] FIWARE Consortium, Context Broker GE.http://catalogue.FIWARE.org/enablers/publishsubscribe-context-broker-orion-context-broker

[29] FIWARE Consortium, Big Data GE. http://catalogue.FIWARE.org/enablers/bigdata-analysis-cosmos

[30] XIFI Federation Monitoring: http://wiki.fi-XIFI.eu/Public:Federation_Monitoring