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

eXperimental Infrastructures for the Future Internet

# D1.3: Federated Platform Architecture v1

Revision: 3.2

| | |
|---|---|
| Work package | WP 1 |
| Task | Task 1.3 |
| Due date | 31 December 2013 |
| Submission date | 12/02/2014 |
| Deliverable lead | University of Southampton IT Innovation Centre |
| Authors | Mike Surridge (IT Innovation), Berndt Bochow (Fraunhofer-Fokus), Silvio Cretti (CNET), Paul Grace (IT Innovation), Stefano Modafferi (IT Innovation), Brian Pickering (IT Innovation), Pascal Bisson (THALES), Maria Di Girolamo (ENG), Joaquim Iranzo (ATOS), Cyril Dangerville (THALES), Panos Trakadas (Synelixes) |
| Reviewers | Fernando Lopez Aguilar, Daniel Nehls |

| | |
|---|---|
| Abstract | This deliverable describes the current version of XIFI federated platform architecture starting from additional analysis of federation requirements elicited in work package 8, and discuss additional functionalities required to address a range of requirements and business models related to the XIFI federation itself and its sustainability beyond the project. |
| Keywords | Federation, Architecture, Functionalities, Requirements tracing |

**Document Revision History**

| Version | Date | Description of change | List of contributor(s) |
|---|---|---|---|
| 0.1 | 30.10.2013 | First draft (skeleton) | Mike Surridge (IT Innovation) |
| 0.8 | 12.11.2013 | Input of additional requirements of non-conventional infrastructures | Berndt Bochow (Fraunhofer-Fokus) |
| 1.0 | 12.12.2013 | Section 2 on federation models and initial introduction | Paul Grace (IT Innovation) |
| 1.1 | 14.12.2013 | Section 2 on technical requirements and Section 4 on the XIFI federation architecture | Silvio Cretti (CNET) |
| 1.2 | 16.12.2013 | Section 3 on SLA and resource discovery | Joaquin Iranzo (ATOS) |
| 1.3 | 18.12.2013 | Section 3 on usage monitoring | Panos (Synelexis) |
| 1.4 | 05.01.2014 | Section 3 on security | Pascal (THALES) |
| 1.5-1.9 | 05-01.2014 | Entire document revised and rearranged. | Stefano Modaferri (IT Innovation) |
| 2.0 | 11.01.2014 | New version released for WP internal review | Stefano Modaferri (IT Innovation) |
| 2.9 | 27.01.2014 | Significant rewrite addressing feedback from Project TM. | Stefano Modaferri (IT Innovation) |
| 3.0 | 29.01.2014 | Revision and consistency check | Silvio Cretti and Federico M. Facca (CREATE-NET) |
| 3.1 | 12.02.2014 | Revised after further discussion with the Project TM. | Mike Surridge (IT Innovation) |
| 3.2 | 13.02.2014 | Final partner revisions in response to first review | Brian Pickering (IT Innovation) |

**Disclaimer**



| 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 covers a logical view of the XIFI federated platform architecture. The architecture presented is based on the analysis of FI-WARE GEs and the different discussions that occurred with FI-WARE Architects to ensure alignments between the two projects.

The deliverable includes a review of the requirements related to federation (Section 2) and a description of the first complete version of the XIFI architecture (Section 3) showing the extent to which these requirements are addressed. This is followed (in Section 4) by an analysis of additional or enhanced functionalities that could be considered for development in Year 2 and beyond. This analysis is a report of work in progress, showing some potential future developments, but no decisions have yet been made regarding which (if any) will actually be implemented. This is still being discussed within the project and will be reported in D1.5.

The process of creating a federation involves technical, operational and business aspects, often more than the simple communication overhead, that are clear if the distributed nature of the solution is taken into account. Therefore a federation is not the simple sum of different, even identical, nodes and all the different viewpoints have to be considered in designing it. This deliverables takes account of these aspects, covering the different inputs collected so far from internal stakeholders (e.g. infrastructures and federation managers) and from external stakeholders (e.g. technology providers and developers).

**Section 2 starts by reviewing possible federation models** (shortly introduced in D1.1), based on the taxonomy developed by the FedSM project. Then it **reviews requirements** related to federation from three sources:

- technical requirements derived from the initial analysis of use case scenarios and use case projects carried out in WP1 and described in deliverables D1.1 and D1.2;
- a survey of XIFI nodes carried out more recently, to capture operational requirements and constraints related to federation management functionality;
- a review of previous federation efforts in the context of Future Internet developments from a socio-economic perspective, extracted from WP8 and described in D8.1.

Although the natural place for requirements elicitation and analysis is T1.2, some discussion has been included here concerning requirements captured after the analysis from D1.2 (the recent survey results and the socio-economic requirements from D8.1). Based on the analysis of requirements and models, the federation model chosen for XIFI is a hybrid federation model covering aspects of the "one-stop-shop" model (a common advertising channel for infrastructures where users choose which infrastructure they want to use), and the "integrator" model (the federator decides which infrastructure 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 XIFI and accessed in a uniform way under common terms by users (i.e. the "integratior" model can be applied), but non-conventional resources (e.g. sensor networks, LTE networks, etc.) are advertised by the federator and negotiated / accessed directly with / through the infrastructures.

Requirements and their analysis in XIFI are constantly evolving, thus deliverables can only provide a snapshot representing a precise point in time. For this reason, in line with the agile method followed in the project, since the project start we adopt a "liquid" solution to manage them. This deliverable briefly documents the approach that so far was not made transparent. Use case scenarios, requirements (both the high level ones described in this document and the more detailed ones associated to each architectural component) are tracked in an online tool (i.e. http://redmine.fi-xifi.eu/). A more complete description will be made available in D1.1b.

**Section 3** then **describes the current XIFI architecture**, which fully covers the technical requirements identified so far. This section explicitly relates each technical requirement to the architectural features that support it. These requirements encompass the features of a federated

 

system, and the operational approaches identified as potentially desirable from the up-to-date survey among the XIFI nodes. The architecture leverages on FI-WARE Cloud Hosting GEs in order to deploy a cloud management infrastructure on each node of XIFI but goes a bit further providing also a federation layer that guaranties a transparent access to all the nodes of the federation and supports high availability of the services offered. As this document describes, the rationale that drives the architecture development takes care of:

- providing an access to the distributed resources at an higher level regardless of the physical location or the specific infrastructure/node the resources belong to,
- avoiding single point of failure,
- supporting new nodes/infrastructures, willing to join the XIFI federation, easing and making as much automatic as possible the joining process.

Note that the current architecture doesn't support all of the requirements identified in Section 2, mainly because the socio-economic requirements were obtained by WP8 in parallel to the work described here, and have not yet been fully analysed from a technical perspective. This will be done in the next WP1 cycle and covered in detail in Deliverable D1.4. However, it is already clear that many of the socio-economic requirements are already met (e.g. uniformity of access, control over access rights, continuity of service, etc.). Some requirements appear to be beyond the scope of XIFI (e.g. the need to address aspects of the digital divide, which are independent of the XIFI architecture, and cannot be addressed with the resources available to XIFI). The rest may require some extensions of the current architecture and relevant FI-WARE Generic Enablers or other components, notably in the arrangements for federated security, usage monitoring, and SLA-based management.

**Section 4 describes existing and potential new federation functionalities**, in line with current XIFI federation needs and its sustainability beyond the project. It highlights options whereby new features could be implemented by using or extending of Generic Enablers in Year 2 of XIFI or beyond. This section does not present any final conclusions on which options should be implemented. It reports results from work in progress designed to provide input and inform the discussion of what should be attempted within the project consortium. The requirement for them will be further analysed and reported in D1.4 (M15) and their inclusion (or not) in the architecture documented in D1.5 (M18).

# TABLE OF CONTENTS



 

 

# LIST OF FIGURES



 

# LIST OF TABLES



 

# ABBREVIATIONS

| | |
|---|---|
| **API** | Application Programming Interface |
| **DCRM** | Data Centre Resource Management |
| **DHCP** | Dynamic Host Configuration Protocol |
| **(D)DNS** | (Dynamic)Domain Name System |
| **DOW** | Description of Work |
| **EBM** | Exploitation and Business Modelling (Working Group) |
| **EC (EU)** | European Commission (European Union) |
| **FI-PPP** | Future Internet Public-Private-Partnership |
| **FMC** | Fundamental Model Components |
| **GE** | Generic Enabler |
| **GPU** | Graphical Processing Unit |
| **GRE** | Generic Routing Encapsulation |
| **GUI** | Graphical User Interface |
| **HTTP(S)** | Hypertext Transport Protocol (Secure) |
| **IaaS** | Infrastructure as a Service |
| **IdM** | Identity Management |
| **IEEE** | Institute of Electrical and Electronics Engineers |
| **IETF** | Internet Engineering Task For |
| **IoT** | Internet of Things |
| **IP** | IPv4, IPv6Internet Protocol (version 4, version 6) |
| **ISP** | Internet Service Provider |
| **JSON** | Javascript Object Notation |
| **KPI** | Key Performance Indicator |
| **L2** | Layer 2 |
| **L3** | Layer 3 |
| **LOS** | Line of Sight |
| **LTE** | Long Term Evolution |
| **MAC Id** | Media Access Control address Id |
| **NREN** | National research and Education network |
| **OAUTH** | Open Standard for Authorization |
| **OLA** | Operational Level Agreement |
| **OSI** | Open Systems Interconnection |
| **P2P** | Peer to Peer |
| **PaaS** | Platform as a Service |
| **PAP** | Policy Administration Point |

| | |
|---|---|
| **PDP** | Policy Decision Point |
| **PEP** | Policy Enforcement Point |
| **QoE** | Quality of Experience |
| **RAN** | Radio Access Network |
| **RF** | Radio Frequency |
| **RFC** | Request For Comments |
| **SaaS** | Software as a Service |
| **SAML** | Security Assertion Markup Language |
| **SBF** | Service Business Framework |
| **SCIM** | Simple Cloud Identity Management |
| **SDC** | Software Deployment and Configuration |
| **SE** | Socio-economic |
| **SLA** | Service Level Agreement |
| **SSO** | Single Sign On |
| **UAV** | Unmanned Airborne Vehicle |
| **UC** | Use Case |
| **UI** | User Interface |
| **URI** | Uniform Resource Indicator |
| **USDL** | Unified Service Description Language |
| **VANET** | Vehicular ad hoc network |
| **VLAN** | Virtual Local Area Network |
| **VM** | Virtual Machine |
| **WSN** | Wireless sensor, actor and actuator networks |
| **XACML** | Extensible Access Control Markup Language |

# 1 INTRODUCTION

## 1.1 Context, objective and scope of this deliverable

This deliverable provides the first complete version of the XIFI federated architecture. The objective of XIFI federation is to create a sustainable pan-European federation of Future Internet test infrastructures, by setting-up and operating a pan-European facility, updating and federating different types of infrastructures, for trials of applications based on the FI-PPP Generic Enablers. A key step and the focus of this report is the creation of a Community Cloud advanced by Future Internet (FI) facilities for the benefit of FI developers. The Community Cloud will be exposed through FI-Lab, the Future Internet Lab initiated by FI-WARE (http://lab.fi-ware.eu), thus allowing developers to access different resources around Europe. The architecture presented keeps into consideration the above objective and needs by internal and external stakeholders (e.g. infrastructure owners, technology providers, developers, etc.).

In order to do so, a roadmap has been followed that, starting from the analysis of the different possible federation models, considering the possible usage scenarios described in D1.1 and reviewing the technical requirements elicited in the previous activity in the WP, conduct to the definition of the XIFI federated architecture. Future possible enhancements and evolution of the current requirements and architecture are discussed in the final part of this document.

**Section 2** starts by reviewing possible federation models, based on the taxonomy developed by the FedSM project. It then reviews requirements related to federation from three sources:

- technical requirements derived from the initial analysis of use case scenarios and use case projects carried out in WP1 and described in deliverables D1.1 and D1.2;
- a survey of XIFI nodes carried out more recently, to capture operational requirements and constraints related to federation management functionality;
- a review of previous federation efforts in the context of Future Internet developments from a socio-economic perspective, extracted from WP8 and described in D8.1.

The socio-economic requirements have not been fully analysed at the time of writing and their impact (if any) on the final architecture should be evaluated in the coming months. Although the natural place for requirements elicitation and analysis is T1.2, operation preferences expressed in the survey, and socio-economic requirements from WP8 were captured after the completion of deliverable D1.2 from T1.2. An initial discussion of requirements is inserted to provide a more complete context for the subsequent presentation and discussion of the architecture and its possible future evolution. The conclusions of this analysis will be reported in the coming WP1 deliverables (D1.4 and D1.5).

To better allow readers to track use case scenarios, requirements (both the high level ones described in this document and the more detailed ones associated to each architectural component) and their evolution, we make public the online tool adopted to support the process (i.e. http://redmine.fi-xifi.eu/) A more complete description of the tool-based process will be included in D1.1b.

Based on the analysis of requirements and models, Section 2 finalizes a decision for the XIFI federation model.

**Section 3** describes the current XIFI architecture, showing how it supports the requirements. As said, the socio-economic requirements, elicited at the time of production of this work, should be better analysed from a technical perspective in order to understand what is applicable to XIFI. For this reason, their coverage by the current architecture is only partial.

Following the analysis in Section 2 and architecture status in Section 3, Section 4 has the objective of reviewing federation functionalities supported and identifying new potential federation functionalities that could be addressed by further evolution of the federated architecture within the life span of the project or beyond. The scope of the section is not to present a final decision on features to be added either provide a solution for their inclusion. It is designed to provide inputs to the discussion among

 

consortium partners for the upcoming year of activities. These discussions should be concluded by the end of Year 1, and the resulting decisions will be described in future deliverables D1.4 and D1.5.

## 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[2]:

- 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 and of potential improvements in the features provided.
- XIFI federation operators 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** analyses the current state of the art in federation and presents the requirements collected so far.
- **Section 3** describes the current architecture.
- **Section 4** describes federation features, discussing how they are related to existing Generic Enablers and which of the features might be considered as candidates for a future extension of the architecture.
- **Section 5** provides a summary of conclusions.

The survey used to capture operational considerations for federation members (nodes) is also included in this document as an appendix.

[2] The list refers to the Main stakeholders as defined in WP9.

# 2 FEDERATION REQUIREMENTS AND MODELS

## 2.1 Review of federation models

Federation is defined in the Oxford Dictionary as: "*an organization within which smaller divisions have some internal autonomy"*. D1.1 [1] introduced the core concepts of the XIFI federation, which is a federation of infrastructures offering resources to the developers of Future Internet applications, services and trials. Such infrastructures typically offer traditional storage and compute resources, and deploy instances of FI-WARE Generic Enablers (that provide re-usable services important for application development, e.g., data analytics, context-based message brokering, IoT gateways, and many more). Further infrastructures may also provide specialist and highly heterogeneous resources (e.g. sensors generating smart city data, WiFi testbeds, etc.). In the first instance, five traditional infrastructures have become the initial federation members; membership is to be extended by around ten new infrastructures in the XIFI open call and potentially from others willing to join.

An important consideration of any federation is how it will operate. The FedSM project (www.fedsm.eu) has created a framework in which the operation of a federation can be modelled. We will use this framework to explore and understand the different kinds of federation in relation to XIFI. That is, investigate to what extent a particular federation model provides value to the XIFI users and members and meets these stakeholders' requirements [2]. The FedSM framework was chosen for two reasons: i) simplicity: it promotes a viewpoint of clearly and simply identifying federation operation beyond technical descriptions; ii) relevance: the framework is developed alongside Grid infrastructure federations (EGI.eu and Helix Nebula) that are in many aspects similar to cloud-based federation as in the case of XIFI.

FedSM considers three actors in a federation:

(1) The **user** or consumer: anyone requiring the services or resources offered by the federation; in XIFI these are defined in D1.1 [1] as "*Future Internet Developers (intended as IT professionals involved in the development of FI applications): application developers that want to leverage on FI-PPP technology platforms to develop innovative applications for so called Future Internet scenarios (e.g. smart mobility, smart energy, smart healthcare, ...)*".

(2) The **federation member (infrastructure)**: an individual or component joining the federation and thereby offering their services / resources to anyone using the federation; XIFI defines these in D1.1 as "*infrastructures offering capacity to host Future Internet applications and advanced hardware/services that can be used to support Future Internet application developers. As such Future Internet infrastructures are service hosting providers*" [*op.cit.*]

(3) The **federator**: the individual or component controlling and/or managing the result of federating individual members ; in particular whose goal is to provide value-added services related to the whole federation.

In order to utilise a service of the federation, these actors can interact in one of three ways (as depicted in Figure 1):

- *Certification.* The user interacts directly with the federation member (at no point is the federator involved). However, the federator may have certified the federation member such that the parties can interoperate.
- *Loose*. Part of the service interaction involves the user interacting with federation services before interacting directly with the federation member. For example, a user may discover available federation members and their contact information using a centralised federation discovery service, before invoking the service directly on the infrastructure.
- *Integrated*. All interaction is between the user and the federator. Using the same example, a user discovers an infrastructure through the centralised service but also invokes the service using a broker provided by the federation.

 

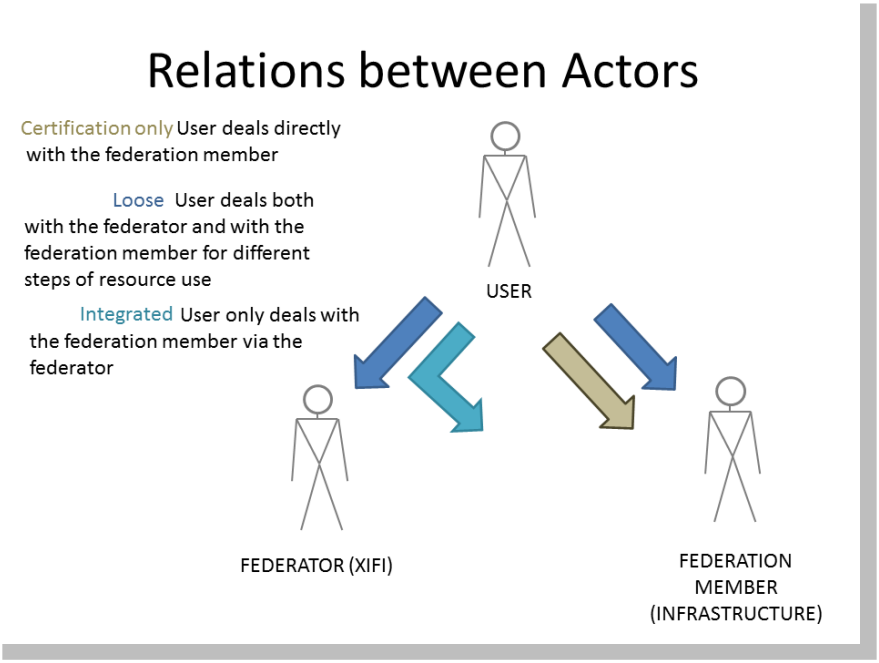


*Figure 1: Federation actor relation types*

FedSM provide five instances of the framework that illustrate how a federation can operate in different ways; they can be termed *federation models*. They provide an excellent tool to inform decisions about how the XIFI federation should operate in order to meet the requirements of its stakeholders, i.e. which services should be provided by the federation and which should be provided by the member? And how should the user interact with these services? We now present the models and then return to them after examining the expectations of the XIFI federation members (see 2.2.1) - this allows us to analyse which federation models best suit XIFI.

### 1. Invisible Co-ordinator

The federation is effectively a certification or validation authority. The federator defines membership rules, and checks compliance of members. The federation member (XIFI infrastructure) works to comply with the rules and seeks checks/certification from the federator. The user finds and engages with infrastructures via other channels, e.g. search engines, marketplaces not provided by the federation, etc. Certification authorities and franchises are real examples of this model in operation. Figure 2 illustrates how the actors interact. The users interact directly with the infrastructure (indeed they may have limited knowledge of the role of the federation). The infrastructures interact with the federator to ensure compliance e.g. they may verify that they implement a standard correctly or provide a service in a particular way.

 

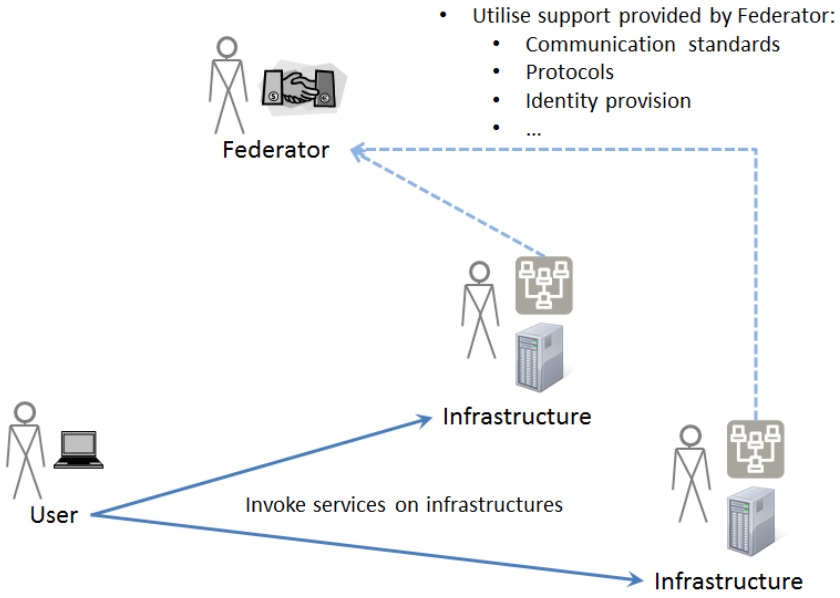


*Figure 2: The Invisible Co-ordination Federation Model*

## 2. Advisor

Where it is important that federation members can advertise their offerings and users can easily find them—the Advisor model goes beyond co-ordination and in this case the federation provides an initial port of call to find appropriate federation members. Figure 3highlights the relationships between the actors. The federator advises federation members (XIFI infrastructures) on how to promote their capabilities through federation. The federator advises users on where to find the capabilities they need. The users decides which federation members to engage with. After initial referral, interaction is between infrastructure and user. Real examples of this federation are government help desks, Amazon Partners, etc.

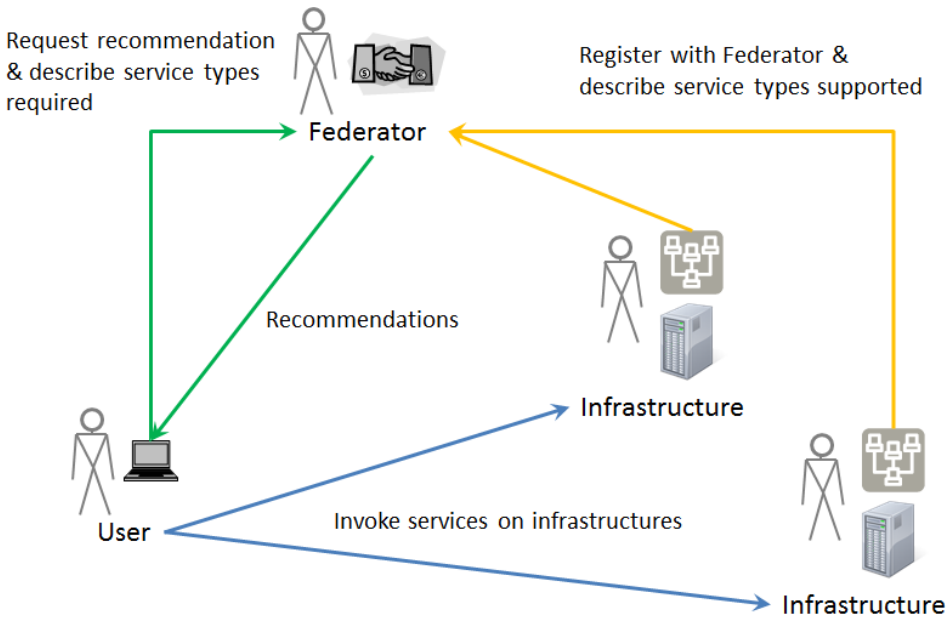


*Figure 3: The Advisor Federation Model*

### 3. Matchmaker

As opposed to an advertise and discovery model of federation, the Matchmaker model allows users to input their needs and the federation provides a matching service of federation members (XIFI infrastructures) with the best suited capabilities (as illustrated in Figure 4). Once again, the federator advises infrastructures on how to promote their capabilities through federation, and the infrastructures decide what capabilities to offer and the associated terms and conditions. The federator matches requests from users to capabilities / offerings from infrastructures, and hence it controls resource allocation by making reservations on behalf of the user. After reservation, it is up to the infrastructure to control the exploitation of resources and execution of services / applications. Brokers and http://kayak.com are real examples of this federation model.

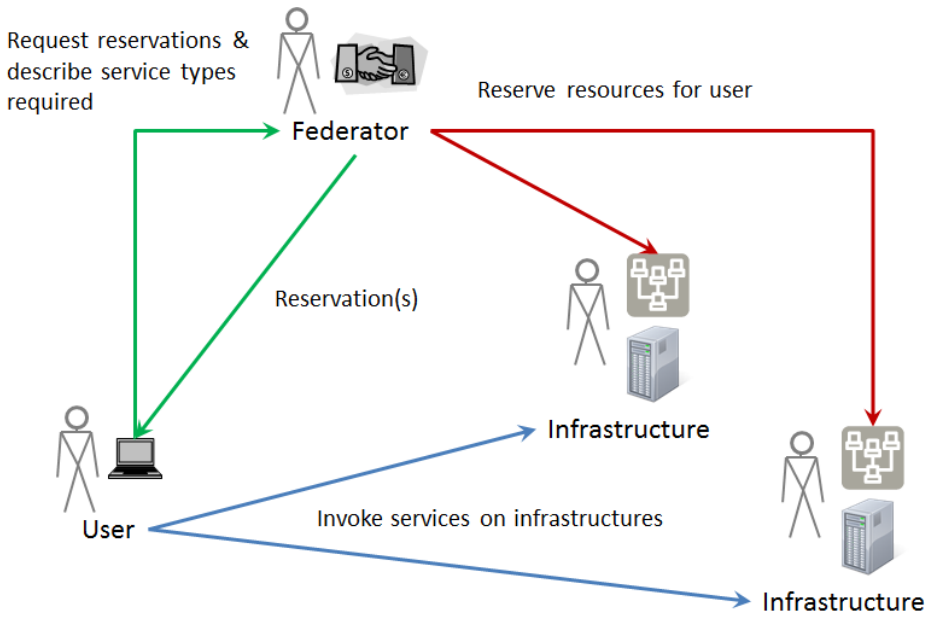


*Figure 4: The Matchmaker Federation Model*

### 4. One-Stop-Shop

A federation beyond the prior three models with additional support from the federation to monitor usage such that billing can be provided by the federation, i.e., the federation handles transactions. Once again, the federation provides a channel for the federation members (XIFI infrastructures) to advertise resources such that they can be discovered and utilised (illustrated in Figure 5). Users pay the federator who is the initial point of contact. Infrastructures bill one another for their contributions. Examples are airline code sharing, and online train booking.

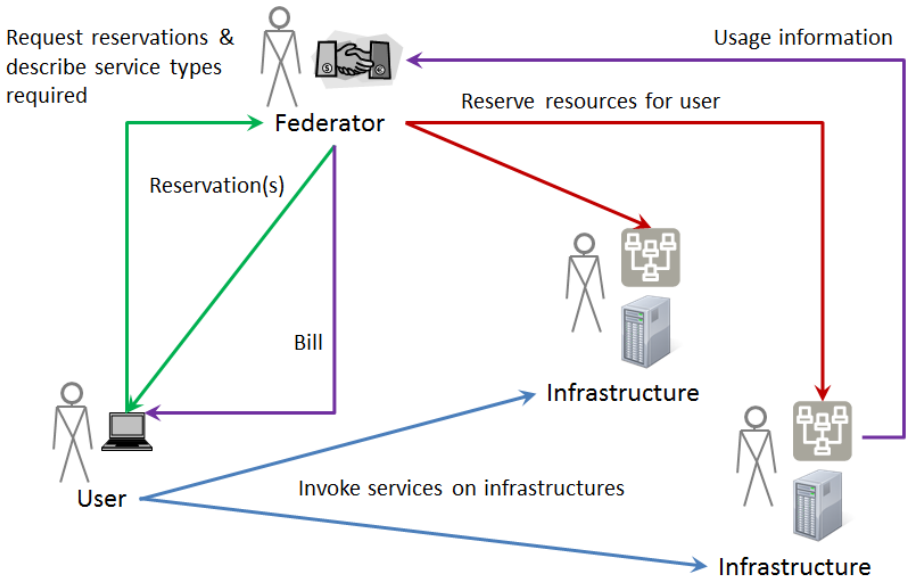


*Figure 5: The One-Stop Shop Federation Model*

## 5. Integrator

The federation takes responsibility for all interaction with the user; for example they are essentially a prime contractor handling all engagement. The user interacts with the federation in order to search for services, have them reserved, and then invoked at the appropriate time. All payment is made to the federator who manages the accounting of and payment to federation members (see Figure 6).

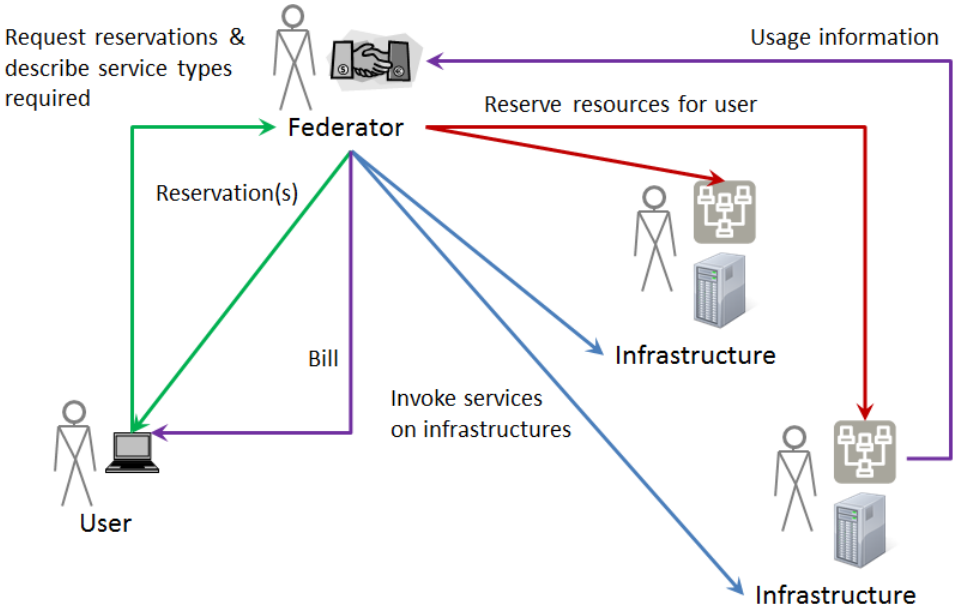


*Figure 6: The Integrator Model*

In practical terms, we may distinguish these models in terms of two extremes: i) a certification role only and ii) complete and tight integration. There is also a middle path: the loose federation. The key thing to learn from these models is that for each service provided by XIFI we must consider whether it is provided by the federator to the user, or whether it is provided directly by the infrastructure.

For now, given the responses from the initial federation members (the five nodes and GEANT), as well as the target to produce a community cloud based on FI-WARE technology, the most appropriate federation model would seem to lie between **one-stop-shop** and **integrator**. This will be discussed

 

again in Section 2.2.1.

## 2.2 Federation technical requirements

In this subsection we present a recap of the information related to the technical requirement specifications collected. This aims at helping to map the generic use case scenarios (present in D1.1) to the requirements (present in D1.2) and finally to the architecture definition (draft in D1.1 and first complete version in this document) so as to sort out and align the contributions present in the two previous WP1 deliverables (D1.1 and D1.2).

The following figure represents the process flow from use case scenarios to requirements and architecture (where detailed requirements for each component are provided) highlighting where (which deliverable) the specific topic has been introduced and described. It is important to note that requirements passed a refinement process from their version in D1.1 and D1.2.


*Figure 7: From Use Case Scenarios to Architecture*

Since the requirements are evolving, the best way to handle them and keep up to date the mapping of scenarios to requirements and to the architecture (detailed requirements for each component) is not a static document like an official deliverable. For this reason we decided to use an online tracking tool like Redmine. At: http://redmine.create-net.org/projects/xifi-requirements/issues the reader can find the latest version of requirements and their mapping. Of course stable and consistent snapshots will be taken from the Redmine tool and provided in the next WP1 deliverables.

In order to keep the development process of XIFI consistent with respect to the scenarios and requirements elicited, we adopted a simple process that allows us:

- to keep always aligned and up to date the information gathered;
- support the tracking and mapping among use case scenarios, high level requirements and architectural components;
- keep under control the coverage and the coherence of what has been developed with respect to the high level requirements.

As depicted in Figure 7, the flow of information starts from the use case scenarios based on expectations gathered from the XIFI stakeholders through interviews and surveys (see D1.1 for more details on this). The use case scenarios have then been converted into a set of high-level requirements that provide the basis for the development of the XIFI federated platform. From the high level requirements, the XIFI architecture has been defined where high level components or subsystems and their relationships have been identified. Then each component/subsystem, present in the architecture, has been assigned to a partner in order provide a detailed analysis (detailed user stories/requirements have been defined), a detailed design (splitting the original architectural components in more elementary ones) and an implementation. These last activities are delegated to the single component's owners in the relative work packages (WP2/WP3/WP4).

In order to support this process, we adopted a tracking tool so as to help in keeping the tracking and mapping among scenarios, high-level requirements and detailed user stories/requirements. As already said, the tool selected is Redmine. It is configured in the following way:

 

1. Two Redmine projects called XIFI-Requirements and XIFI-Components have been created.
2. In the XIFI-Requirements project the scenarios (with attribute *Category* set to *Scenarios*) and the high level requirements (with attribute *Category* set to *Requirements*) are described.
3. In the XIFI-Components project the user stories are described for each component (with attribute *Category* set to the component name).
4. Then, leveraging on the possibility to link different items present in Redmine, the mapping between scenarios and high level requirements and the mapping between high level requirements and component user stories are provided.

In the following, we summarize the results of the re-submission of a survey to the infrastructure owners, in order to gather expectations about the XIFI federation. Based upon this, the scenarios and the requirement specifications a mapping table will be presented highlighting how each use case scenario is mapped onto the corresponding requirements.

### 2.2.1 Expectations from infrastructures concerning federation operation

Expectations were collected from XIFI infrastructure owners (federation members) right at the beginning of the project and the results were shown in D1.1. After clarifying the structure of the XIFI federation and specifying how it was intended to work, it has become clear that the infrastructure owners now have a much better understanding of XIFI, and some of their early inputs needed revision. A new and updated survey was therefore carried out. The new results are shown in this sub-section.

| Function Name | Description |
|---|---|
| Identity Provision | Issuing and managing identities and attributes to users. |
| Authentication | Perform authentication of a user identity. |
| Authorisation | Authorising whether a user has access to a particular resource at a particular time. |
| Access Control | Enforce security policies and control access to local resources. |
| Resource Discovery | Maintain a registry of resources that can be searched by users. |
| Resource Allocation | 1. Allocating my resources to match user requirements (i.e. a match-making service).<br>2. Informing users that my resources match their requirements, but leaving them to contact me to get resources allocated (i.e. a recommendation service) |
| Resource Monitoring | Monitoring the availability of resources. |
| Usage Monitoring | Monitoring usage of my resources and generating accounting data (based on SLA terms) for users coming through XIFI |
| Exception Tracking | Handling of errors and bugs in the provision of resources to users; the reporting of those exceptions and the tracking of the handling of the exception until resolution. |
| SLA Negotiation | Users and providers agree on a level of service that will be provided. |
| Billing | How users pay for the resources that they use. |

*Table 1: List of federation functions in XIFI*

Table 1 lists the functions[3] that are provided (or potentially provided) by the federation to its users. These provide the basis of a questionnaire posed to the original federation members (infrastructures) registering their requirements concerning how they wish to operate. That is, there may be some functions they absolutely want the federation to manage, and some that they do not. There may also be many where they do not care (i.e. they have a solution, but there is no overriding requirement for it to be used instead of a federation service). The survey is attached in completion in the appendix and the results are shown in Table 2.

[3]This list is very common and includes the typical features provided by a federation. The interest reader can be referred to the FedSM deliverables (http://www.fedsm.eu).

| XIFI node (member) | | Has no existing solution: should be provided centrally by the federation | Have our own solution but prefer to use a federated solution | Prefer our own solution but will accept a federated solution | Should be provided and controlled by the member. |
|---|---|---|---|---|---|
| **Berlin** | | Exception Tracking (except for non-conventional) | Usage Monitoring | Authorisation<br>Resource Discovery (for conventional)<br>Resource Allocation<br>Resource Monitoring<br>SLA Negotiation | Access Control<br>Resource Discovery (for non-conventional)<br>Resource Monitoring (for certain resources) |
| **Brittany** | | Resource Allocation<br>Usage Monitoring<br>Billing | Identity Management<br>Authentication<br>Authorisation<br>Access Control<br>Resource Discovery<br>Exception Tracking | Resource Monitoring<br>SLA Negotiation | |
| **Irish Node** | **HEAnet** | Access Control<br>Resource Discovery<br>Resource Allocation<br>SLA Negotiation | | Identity Management<br>Authentication<br>Authorisation<br>Exception Tracking | Resource Monitoring<br>Usage Monitoring |
| | **WIT** | Identity Management<br>Authentication<br>Authorisation<br>Access Control<br>Resource discovery<br>Resource allocation<br>SLA Negotiation<br>Billing | Monitoring<br>Usage Monitoring<br>Exception Tracking | | |
| **Sevilla/ Malaga** | | | Identity management<br>Authentication<br>Allocation<br>Monitoring | SLA Negotiation<br>Billing<br>Security | |
| **Trento** | | Identity Management<br>Authentication<br>Authorisation<br>Access Control<br>Resource discovery<br>Resource Allocation<br>SLA Negotiation<br>Exception Tracking<br>Billing | Resource Monitoring<br>Usage Monitoring | | |

*Table 2: Responsibility model of federation functions*

From Table 2 it is clear that:

- Infrastructures are willing to allow the federator to provide common centralised services for many functions, e.g. Resource discovery. This differs from the initial survey response presented in D1.1 where less co-ordinated models were preferred. This is likely due to a greater understanding of the actors and relationships within a federation offered by XIFI.
- The best fit to infrastructure expectations falls between the **one-stop-shop** and the **integrator** federation model depending on the resource/service considered: the federator acts as an

integrator for the “conventional” data centre services (computational capacity) but acts more as a broker for the non-conventional ones (sensor networks, LTE networks etc.) and users interact with the infrastructure directly when wishing to use such types of resources.

- The federation however must be flexible for heterogeneous infrastructures that are not simply data centres but offer non-conventional resources too. In certain cases operations may need a direct interaction between user and member. The Federator should facilitate this interaction.

### 2.2.2 Use case scenarios summary

D1.1 defined the following high-level scenarios:

1. UC-1 - Joining the federation: an infrastructure owner wants to join the XIFI Federation.
2. UC-2 - Setup and usage of a development environment: a developer (UC project) wants to use the services offered by the XIFI platform in order to develop and test innovative applications.
3. UC-3 - XIFI Services & tools for private cloud setup: a developer (UC project) creates and uses his private cloud using the XIFI services and tools but the infrastructure is not federated with XIFI.
4. UC-4 - User support: an end-user of the XIFI Federation requests support and help.
5. UC-5 - Network and Data Centre operations: monitoring and operating an infrastructure as part of the XIFI federation.

For a detailed description of these scenarios, please refer to the D1.1 deliverable.

### 2.2.3 Requirement specifications

From the refinement of requirements presented in D1.2 and from the re-submission of the previous survey (see 2.2.1), the following high-level requirements have been identified. XIFI uses an agile approach, so the requirements and their analysis are constantly changing. For this report, a snapshot was taken on 31 January 2014. (At any point in time, the latest version of these requirements can be found on XIFI Redmine – see 2.2).

1. REQ-1 - Tools to set-up a new infrastructure - Provide tools to support the minimal set up of new infrastructures joining the XIFI federation starting from bare metal to an up & running XIFI node where software enabling cloud computing, monitoring and security functionalities has been (almost) automatically installed.
2. REQ-2 - Registration process for joining the federation - Define the registration and validation process and tools for an infrastructure willing to join the federation. The process should include registration of services (e.g. OpenStack) and resources (e.g. super-user identities, servers).
3. REQ-3 - Set up of development environments - XIFI should provide a marketplace where the users can search and reserve resources and services, offered by the XIFI federated nodes, in order to create their own development environments. Services and resources can be provisioned through a SaaS or PaaS model transparently with respect to the node they belong to.
    - REQ-3.1 - Non-conventional services - In the XIFI marketplace also "non-conventional" services (like sensor network, LTE network etc.) can be advertised. Their reservation and the negotiation of their usage could be also conducted "manually" contacting directly the infrastructure owner.
    - REQ-3.2 – Support scalability. Scalability and elasticity of the development environments should be supported.
4. REQ-4 - Network configuration - Support network configuration in order to provide the requested network resources (e.g. bandwidth, cross-infrastructure connectivity) using technologies like software defined networking
5. REQ-5 - Regions Availability - Allow services and resources to be located in different regions so as to:

 

- Guarantee location constraints: content and data can be subject to geographical and legal restriction about where they can be stored and processed.
- Support low network latency.

6. REQ-6 - No Single Point of Failure - In order to avoid a single point of failure, XIFI should provide services as high availability: federation level XIFI services and tools (like Cloud Portal, Resource Catalogue, Help Desk, and Security System) should be provisioned in high availability.
7. REQ-7 - Backup and Recovery - XIFI should support indirectly backup and recovery services through data replication and snapshot storage. This means that XIFI allows data replication and snapshot storage on its community cloud but it is responsibility of the developer to arrange his/her services/applications so as to take advantage of these features. Only in specific circumstances and on-demand a traditional backup service can be arranged.
8. REQ-8 - Security - Security and Privacy services like authentication and authorization, single sign on, identity federation, data protection, data privacy, network security, security monitoring and auditing should be provided.
9. REQ-9 - Monitoring - XIFI should provide monitoring of network devices, servers (both physical and virtual) and services in order to gather data for Fault Management and Performance Management. Monitoring data should be collected either directly by XIFI or gathered from an infrastructure proprietary monitoring system.
    - REQ-9.1 - Federation Monitoring - Monitoring data should be aggregated and shown to the end user (having grants to see it) at the federation level through the use of a graphical user interface or a dashboard.
    - REQ-9.2 - Security Monitoring - XIFI should monitor possible security attacks or threats in order to be able to react promptly.
    - REQ-9.3 – Logging and Accounting - User activities should be logged.
10. REQ-10 - Help Desk and Tutoring - Facilities should be provided for handling requests coming from end users, experiencing issues with the XIFI platform, and/or from infrastructure owners that want to join the XIFI federation. Tutoring material and other initiatives like webinars should be provided.
11. REQ-11 - System Administration and Configuration Management - XIFI should provide tools for handling operations of the infrastructures federated like starting up and shutting down the relevant federation services, putting an infrastructure under maintenance for a period of time, etc.
12. REQ-12 - Special hardware - hardware different from that associated with a classical data centre (like GPU, sensor networks, mobile networks, large storage) should be provided. For a better description of this requirement see section 2.4.
13. REQ-13 - Private Clouds - XIFI should provide support to build a private cloud: a user willing to create his own private cloud should be able to use some XIFI services (see REQ-1) in order to set it up. Moreover other XIFI services, like monitoring or security, can be used to manage this private cloud.
14. REQ-14 - Special facilities: XIFI should provide support for integration with facilities and features offered by infrastructures related to specific domains (like smart hospital, smart factory, smart logistics, and smart media). The integration can be pursued at different levels (ranging from completely private cloud - see REQ-13 - to interconnected infrastructures where only some XIFI services are exercised to – where feasible - completely federated infrastructures that become part of XIFI federation).

### 2.2.4 From scenarios to high-level requirements

Hereunder a mapping table between use case scenarios and requirement specifications is provided.

| Use Case Scenario | Requirement |
|---|---|

| | |
|---|---|
| UC-1 | REQ-1, REQ-2, REQ-6 |
| UC-2 | REQ3, REQ-4, REQ-5, REQ-6, REQ-7, REQ-8, REQ-12 |
| UC-3 | REQ-13, REQ-14 |
| UC-4 | REQ-10 |
| UC-5 | REQ-9, REQ-11 |

*Table 3: Use Case Scenario mapped to requirements.*

The different scenarios and requirements are mapped to architecture components in Section 3. For each component, the single requirements and scenarios are then broken down into user stories that respect atomicity, unambiguity, concreteness, completeness and consistency criteria. These are documented in the respective accompanying deliverables (or in the XIFI redmine).

## 2.3 Federation business requirements

The first socio-economic requirements analysis is documented in Deliverable D8.1. D8.1 is scheduled for M09, i.e. it is simultaneous with this report, so it was not possible to fully consider its socio-economic requirements and map them to technical requirements for the architecture. The XIFI workplan envisages this analysis being undertaken during the update of requirements in D1.4, which will then be taken into account in the refined Federation Architecture presented in Deliverable D1.5 (an update of this report) at M18. However, it makes sense to at least bear in mind the socio-economic requirements even at this stage.

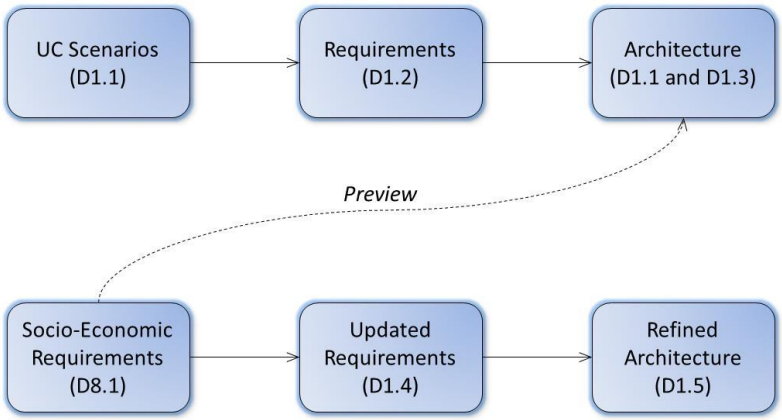


*Figure 8: Introducing Socio-Economic Requirements*

Triggered by early input from WP8, a survey of the founding XIFI nodes was conducted to examine how socio-economic control issues affected the way federated functions should be implemented and operated. Section 2.3.1 describes the results of this survey and provides a very preliminary analysis of the socio-economic requirements from D8.1, indicating how they might amend or extend the existing technical requirements (see 2.2.3). Figure 8 summarizes this process.

### 2.3.1 Socio-economic requirements impacting the federation operation

Deliverable D8.1 provides an extensive review of socio-economic factors and analysis that are relevant to the FI-PPP. Drawing on the *Digital Agenda for Europe*, *Horizon 2020*, four EU co-ordination and research projects (the SMART reports, FI3P, SESERV and TAFI), two related federations (the EIT ICT Labs and ENoLL) and results from the EBM Working Group, a set of seventeen socio-economic requirements has been identified [3]. These are summarised in Table 4

 

below. Note that these socio-economic requirements have not yet been analysed technically from a software engineering perspective. Here we provide an initial assessment of how they related to the previously identified use cases and technical requirements.

| Socio-Economic Requirement | Area | Comment |
|---|---|---|
| **SE_Req1**: *Users want a common look and feel when they access XIFI resources from anywhere at any time, to be able to personalise their entry point, but receive the same levels of service irrespective of where they store data and/or run experiments.* | Usability | This relates to user experience and the desire to make things easy and intuitive for the user(s). |
| **SE_Req2**: *Resources should be accessible from any geographic location, signing in once and gaining access everywhere. The same operational standards should be maintained irrespective of access device or location.* | Single System Image, Usability, Security | Users want to be able to access and exploit resources from anywhere |
| **SE_Req3**: *To maintain trust, users should be kept informed of any and all activities, including faults and potential failures, when using the facilities* | Trust, Operations | There should be a degree of transparency about all operational states; users often accept failures if they are kept informed, and see evidence of resolution |
| **SE_Req4**: *Users want the same high-speed access irrespective of access location; they should be served with the same levels of QoS and QoE at all times* | Usability, Operations | Like telephone and Internet subscribers, users have an expectation that services will run with the same level of performance irrespective of where the service is located. |
| **SE_Req5**: *The Community around XIFI should include all relevant actors and stakeholders, including the XIFI federation and federation members themselves; all should be encouraged to share knowledge and support. Community building and maintenance is very significant.* | Community | Community and participation are important non-functional aspects of any service or system. This may run in parallel and independently of the system or service; or alternatively be supported by the system or service with bulletin boards etc. |
| **SE_Req6**: *Users will inevitably carry out experiments across multiple domains. They* want *to collaborate and share experience (see SE_Req5), but they will also use resource as fits them.* | Utility | As above: this is about supporting the community of users which grows up around the system / service. |
| **SE_Req7**: *XIFI should keep stakeholders informed of overall progress, using some form of appropriate, targeted continuous self-validation* | Self-assessment | This relates to iterative self-monitoring: does the system do what it should be doing? But extends into sharing such information with those most interested in the system. |
| **SE_Req8**: *XIFI should be able to connect to other resources on demand and on an* ad hoc *basis as dictated by the requirements of users. This involves technical as well as commercial relationships to be enabled.* | Inter-operability | This is a requirement for dynamic connection of other resources / services / data sources. |
| **SE_Req9**: *In protecting personal as well as experimental data, XIFI needs to provide appropriate controls and auditability.* | Security | Auditability is a key factor in generating and maintaining trust: users want to know and be shown what's going on. |
| **SE_Req10**: *The direct and broader communities around XIFI need careful management: understanding who they are, what motivates them and how best to communicate with them is essential for XIFI's success.* | Ecosystem | This relates more especially to support for the wider community beyond direct XIFI users. |
| **SE_Req11**: *The participation of indirect actors and/or stakeholders should be considered (cf. SE_Req5).* | Ecosystem | As above. |

| Socio-Economic Requirement | Area | Comment |
|---|---|---|
| **SE_Req12**: *Network traffic between and within sites should be monitored, and be able to be managed. Users may want to know and/or manage and control network traffic.* | Operations | This relates to a user's desire to be kept informed of what is happening and how the system is operating. |
| **SE_Req13**: *XIFI should allow for up and down scaling of resource (ie planned configuration change) as well as the temporary, dynamic allocation of resource during execution (to cater for unforeseen requirement increase or decrease). Users may not be able to size their system requirements accurately, but would still expect them to run.* | Utility | Users expect some leeway in resource handling during execution. This includes planned (scaling up) as well as unplanned (elasticity) increase in resource which should ideally be handled dynamically. |
| **SE_Req14**: *XIFI Users should be able to run innovative and environmentally relevant experiments, connecting to 3rd party services if necessary (see SE_Req8), as well as expecting federation resources to be managed sustainably and efficiently.* | Utility | This is part of the overall objectives of the FI-PPP and the EU initiatives which led to its launch: the FI-PPP should support, facilitate and promote innovative uses of technology. |
| **SE_Req15**: *As part of the federation as an environmentally sustainable entity (see SE_Req14), all resources and facilities should be sharable across all users under appropriate terms.* | Community (Utility) | All IT systems should consider at some level environmental impact and sustainability. |
| **SE_Req16**: *XIFI should provide failover and recovery capabilities. Users expect continued operation as well as data integrity*[4]. | Security | Data and all information related to the work being done by users should be secure and recoverable in the case of failure. |
| **SE_Req17**: *XIFI needs to encourage participation, and complementing the knowledge sharing of the community (see SE_Req5) by providing training and context-specific help functionality. Users want support in preparation as well as during the use of the federated facilities.* | Usability | Another non-functional feature in support of the community, but also in aid of potential users to support and encourage them to use XIFI. |

*Table 4: Socio-Economic Requirements from D8.1*

For each requirement, a broad "area" of interest has been listed in the second column:

- *Community:* anything pertaining to the support of interaction and dialogue between members of a community (e.g. user groups);
- *Ecosystem:* anything relating to the broader Future Internet and FI-PPP commercial and operational environment;
- *Interoperability:* those issues which relate to the federation's ability to access and connect with other types of resource (e.g. sensor networks) and data (e.g. SNS feeds);
- *Operations:* items affecting how the federation can be run on a day to day basis;
- *Security:* areas which relate to ensuring the integrity and confidentiality of information and data, as well as the continued operation (e.g. in the face of attack or other outages);
- *Self-assessment*: this relates to the perceived need for the XIFI federation to monitor and report on its own performance etc.
- *Single System Image:* user desire to operate with a single entity, even though underneath this may equate to multiple physical locations and resources;

[4]This would be part of an SLA with the user; SLA management would have operational as well as commercial implications, of course.

- *Trust:* the desire by users to be able to count on a service/technology especially in terms of security and integrity;
- *Usability:* how easy the user finds to exploit a system;
- *Utility:* how useful the system is: is it fit for purpose?

Requirements in Table 4 have been stated in broadly non-technical terms, representing what potential users from the FI ecosystem might expect from the federation, without considering their relevance or applicability to XIFI. They are related to a wide range of technical and non-technical issues, including but not limited to the federation architecture used in XIFI. In the forthcoming Deliverable D1.4, these socio-economic requirements will be considered further and mapped onto existing and new technical requirements for XIFI. Here, we are only concerned with possible implications for the federation approach and architecture.

**SE_Req1**: relates to the uniformity of user experiences. XIFI does not deal directly with application end-users, but only with those developing and operating such applications on XIFI. Application developers and operators who choose to access resources through XIFI should have a uniform experience for resources offered by the different member of the federation. This SE requirement relates to UC-2 as described in Section 2.2.3, for setting up a development environment or a cloud hosting infrastructure. It also very related to the federation model selected.

The one-stop portal interface and the associated marketplace (technical requirement REQ-3) provide a uniform way to access to federation services and federation member resources. Thus, this requirement poses no immediate additional challenges to the federation architecture.

**SE_Req2**: relates to the uniformity and fairness of access, including sign-on and operational aspects, regardless of access location. This affects application service providers and developers engaged in UC-2 and relates to SE_Req1.

Location-independent access to services and resources is covered by different technical requirements. REQ-3 covers the transparent access to any resources through its declinations in the one-stop portal interface and the associated marketplace, and REQ-8 includes single-sign-on features. REQ-5 supports the possibility to deploy same services in different geographical areas. REQ-6 deals with the fact that access to the service should be possible even in case of partial disruption or unavailability of the system. The requirement may be further covered by the planned expansion of the Federation in Year 2.

**SE_Req3**: relates to the maintenance of user trust through transparency of reporting. This clearly relates mainly to scenario UC-5, though the information being reported may also relate to UC-2, UC-3 and possibly even UC-1.

Achieving ‘trust’ always presents challenges when mapped to technical requirements. This is because measures to promote trust through increased transparency may also degrade trustworthiness (i.e. security) through inappropriate disclosure of vulnerabilities (known or unknown to the operators). Too much transparency may also undermine business confidentiality by leaking information between potentially competing stakeholders, e.g. application developers and operators may want more transparency than XIFI infrastructure owners and operators, yet all need to trust XIFI if it is to be successful. Existing technical requirements such as REQ-9 and REQ-11 cover the basics, but more analysis is needed to determine what should be made transparent to which XIFI stakeholders. For example, when should resource requests (successful or otherwise) at each node be logged and made available to users? This analysis is deferred to D1.4.

**SE_Req4**: states an expectation that users will get uniformly high performance access regardless of location. Variations in access network performance normally arise because the return on investment in network capacity varies (e.g. it may be lower in sparsely populated areas), so less capacity is installed in some regions than others. This is one of the issues that lead to the Digital Divide. XIFI can help by ensuring federated resources are available in multiple regions (REQ-5), and allowing application developers and operators to select the best resources (REQ-3) close to their users. However, XIFI cannot ensure through architectural choices that high performance will be universally available. This requirement can therefore only be addressed by non-technical means, e.g. by regulating the providers

 

of access networks forcing uniformly high performance, or by subsidising investment in regions where returns are less attractive.

**SE_Req5**: this relates to community building. This also goes beyond the scope of the federated architecture platform and includes other different activities in XIFI. It relates to scenarios such as UC-1 (becoming part of the federation is joining the community), UC-3 (using the resources of the federation is becoming part of the community) and UC-4 (services provided to the community). However, these should be addressed mainly by other activities in the project such as the federation office, the training organization, the dissemination activities and the showcase demonstrator. Some support for community building could be included in the architecture by extending requirements such as REQ-10 (help desk and tutorials) in future.

**SE_Req6**: relates to the composition of resources across domains, and also to the interaction between different stakeholders (e.g. developers, application users, technology providers). This relates to scenarios UC-2, for setting up and using XIFI resources. It may in some circumstances also affect application users, e.g. if an application spans resources from multiple XIFI domains.

This requirement is addressed by existing technical requirements REQ-3 (provisioning services and resource), REQ-4 (e.g. in terms of setting up cross-infrastructure network connectivity), REQ-5 (e.g. in term of availability of different "regions") and REQ-8. It is possible that some extensions to REQ-8 regarding access control management policies may be appropriate. This will be analysed further in D1.4.

**SE_Req7**: is concerned with tracking the progress of XIFI in meeting socio-economic needs. This is not something that can be mapped to technical requirements, although it may require some types of monitoring data, e.g. numbers of users, etc., from scenario UC-5 (node operations). This aspect is already covered by REQ-9.

**SE_Req8**: is concerned with the ability of XIFI to dynamically connect users and resources from different domains, potentially including some resources outside of XIFI's direct control. This relates to scenarios UC-1, UC-2 and UC-3, and requirements REQ-3, REQ-13 and REQ-14. XIFI already addresses dynamic allocation and configuration of federated resources, as this is covered by the self-provisioning approach incorporated into scenarios UC-2. The same is true for configuration of XIFI services and tools using external resources in private clouds under scenario UC-3.

Agile configuration may also have implications for the way some federated functions are implemented. For example, in the FITMAN project, smart factory owners would like to run applications that use a smart factory along with XIFI-hosted data storage and analytics enablers. One way to facilitate this would be for XIFI to respect user identities issued by the user's own home domain, which they presumably use to access non-XIFI resources. This is covered by technical requirement REQ-8.

The only issue is if federated resources are provided by an infrastructure owner with constraints on how they are used, or under what terms. Currently this is addressed only for non-conventional services under REQ-3.1, access to which requires interaction with the infrastructure owner. One way to handle this is to include new specific requirements allowing infrastructure owners to specify their terms of access during UC-1, e.g. via an Operating Level Agreement (OLA) between the node and the XIFI. The requirements for UC-2 (and UC-3) could then also be extended to take at least some of these terms into account in the self-service provisioning. This would imply some extensions of REQ-2 and REQ-3, and modification of REQ-14 for aspects covered by the extended REQ-3.

**SE_Req9**: is concerned with privacy protection. This is relevant to UC-2 (and eventually UC-3) for setting up an environment to develop and operate an application, and UC-5 concerning auditing to verify whether privacy has been respected. It is already quite well addressed through REQ-8 and REQ-9.2 and 9.3, and (where appropriate) REQ-13. It is also necessary to balance this socio-economic requirement for privacy against SE_Req3 concerning transparency (see above). The need to verify privacy would normally outweigh concerns about the confidentiality of access logs (requests including refusals), but this doesn't mean all logs should be published.

 

**SE_Req10**: concerns the understanding, management and support of XIFI-related communities, which is related to scenario UC-1, UC-3 and UC-4. Like SE_Req5, this is not directly related to the federation architecture. However, depending on the approach used by the consortium, it may lead to new requirements for community services that could have implications for the architecture. Further discussion with non-technical XIFI WPs about their approach is therefore needed.

**SE_Req11**: relates to the role of indirect participants in sustaining stakeholder ecosystems. This may have implications for the technical architecture, but these cannot be analysed until the indirect participants have been identified and scenarios defined. This will depend on the business models used by the XIFI federator and by infrastructure operators and their users.

It is recommended that D1.4 explore initial feedbacks by application developers adopting XIFI and that WP8 and WP9 should investigate whether any partner's exploitation plans involve the delivery of technical services (including information) to indirect stakeholders. Until this is done, no action is required in WP1.

**SE_Req12**: concerns the need to monitor and manage network traffic within the federation. This is relevant to scenario UC-2 and UC-5. There are limits to the grain size at which traffic can be monitored and managed. To this extent it is addressed by technical requirement REQ-4 which deals with network management, and REQ-9 which deals with resource monitoring.

**SE_Req13**: relates to the need for scalability in response to variable user demand. This relates to scenarios UC-2 and UC-5. It is already largely addressed by technical requirement REQ-3.2 and REQ-5.

**SE_Req14**: concerns the need for flexible interconnection of XIFI and 3rd party resources (which is also covered by SE_Req8 above), and efficient and sustainable resource management within the federation. For the most part efficiency and sustainability is best addressed by the operating practices of individual nodes and does not impact the federation architecture. However, it may make sense to introduce federated efficiency/sustainability features, e.g. energy monitoring and management. An analysis on these aspects should be conducted in D1.4.

**SE_Req15**: relates to the sustainable sharing of resources between domains. Like SE_Req14, no energy-monitoring and management requirements have been considered in the technical discussion. An analysis on these aspects should be conducted in D1.4.

**SE_Req16**: concerns the ability of XIFI to ensure business continuity for its users, through its failover and recovery capabilities.

This is an important socio-economic requirement, already largely addressed by technical requirement REQ-6 (no single points of failure for key services) and REQ-7 (concerning data backup). This arises during UC-2 or UC-3 (setting up a development environment or operational cloud).

**SE_Req17**: concerns the provision of training, which relates to scenario UC-4 (user support), and in that context it is already covered by technical requirement REQ-10. However, this requirement does not only relate to the architecture of XIFI, and should be addressed also by other activities in the project such as the federation office, the training organization, the dissemination activities and the showcase demonstrators.

A more comprehensive analysis, starting from the UC scenarios will be documented in D1.4. The above preliminary analysis provides an initial indication of what has to be investigated:

| Socio Economic Requirements | UC Scenarios | Technical Requirements | To be investigated |
|---|---|---|---|
| SE_Req1 | UC-2 | REQ-3 | |
| SE_Req2 | UC-2 | REQ-3, REQ-5, REQ-6, REQ-8 | |
| SE_Req3 | UC-1, UC-2, UC-3, UC-5 | REQ-9, REQ-11 | Analyse transparency issues, taking account of possible conflicting stakeholder needs. |

| | | | |
|---|---|---|---|
| SE_Req4 | UC-2 | REQ-3, REQ-5 | Liaise with non-technical WPs over possible non-technical approaches. |
| SE_Req5 | UC-1, UC-3, UC-4 | REQ-10 | Consider extensions if required by non-technical WPs to support community building. |
| SE_Req6 | UC-2 | REQ-3, REQ-4, REQ-5, REQ-8 | Consider possible extensions of REQ-8 regarding access control policy management. |
| SE_Req7 | UC-5 | REQ-9 | |
| SE_Req8 | UC-1, UC-2, UC-3 | REQ-3, REQ-13, REQ-14 | Consider the use of SLAs and OLAs to reduce the need for manual interaction with infrastructure owners. Currently this affects non-conventional resources only. |
| SE_Req9 | UC-2, UC-3, UC-5 | REQ-8, REQ-9, REQ-13 | Take account of results from the analysis of SE_Req3. |
| SE_Req10 | UC-1, UC-3, UC-4 | | Liaise with non-technical WPs over the use of community services, which may lead to new architectural requirements. |
| SE_Req11 | N/A | N/A | Liaise with WP8 and WP9 concerning business models for sustainability, which may lead to new architectural requirements. |
| SE_Req12 | UC-2, UC-5 | REQ-4, REQ-9 | |
| SE_Req13 | UC-2, UC-5 | REQ-3, REQ-9 | |
| SE_Req14 | UC-2 | REQ-8 | Consider whether any federated efficiency and sustainability features might be appropriate, e.g. energy monitoring and management. |
| SE_Req15 | UC-2 | REQ-8 | As SE_Req14. |
| SE_Req16 | UC-2, UC-3 | REQ-6, REQ-7 | |
| SE_Req17 | UC-4 | REQ-10 | Liaise with non-technical WPs over the use of training services, which may lead to new architectural requirements. |

*Table 5: Socio-economic requirements mapped to Use Case Scenario*

Although the socio-economic requirements are more specifically associated with impact and user involvement, from the description above there is a clear alignment possible between those requirements, the five basic use case scenarios previously identified and the technical requirements. Possible extension of the technical requirements, in order to cover also detailed aspects of the SE requirements, will be analysed in subsequent deliverables.

## 2.4 Additional requirements

This section describes and addresses some requirements related with non-conventional services and resources like sensor networks, LTE networks etc. These requirements are related with scenario UC-2 and REQ-3, REQ-12 and REQ-14.

### 2.4.1 Non-conventional resources and services

Resources and services of the federation that cannot be summarized under the term 'data centre resources' are herein denoted as 'non-conventional'.

In particular, this refers to infrastructure resources that cannot be easily virtualised. This may be because:

i. they are bound to dedicated hardware resources;
ii. they need to managed according to external regulatory or legal constraints;
iii. they need to be shared across different (internal as well as external) stakeholders;
iv. they are specific to a particular geographic location; or
v. they comprise expensive and sensitive or vulnerable infrastructure materials.

 

The following examples for non-conventional infrastructures can be considered as representative of the problems to address.

- Local radio access networks (RANs), because of sharing a wireless medium and underlying mandatory regulatory constraints regarding non-interference or coexistence in shared frequency bands.
- Wireless sensor, actor and actuator networks (WSNs), because they interface with physical world infrastructures, are associated with particular geographical areas, or control machinery or facilities in the physical world.
- Vehicular networks (VANETs, terrestrial, submarine or airborne), because the equipment is mobile, needs exhaustive maintenance or must be supervised while operating, or because potential failure may be safety-of-life critical.

In order to federate 'non-conventional' resources, several distinct objectives must be considered. In the following obvious requirements are given in more detail.

- Requirement for a dedicated node-centric resource reservation and control for 'non-conventional' resources.

  The nature of 'non-conventional' resources, their capacities, availability and operational and deployment constraints are only known to the node providing these. Nodes may need to dynamically disable or restrict access to their 'non-conventional' resources depending on reasons only observable by the node such as weather conditions or changing local regulations, for example. Hence, planning of resource utilization, resource reservation and access to the resource as well as obtaining user or monitoring data from the resource must be conducted under control of the node. In that the node has to respect the service level agreement (SLA) or Operational Level Agreement (OLA) between the node providing the resource, the federation brokering the resource and the user allocating the resource.

- Requirement for a dedicated node-local resource and policy management.

  'Non-conventional' resources may consist of shared, sensitive or vulnerable equipment deeply buried into the node infrastructure such as a RAN cellular base station and associated mobile equipment, for example. Access policies may thus need to consider availability of equipment and, in addition, availability of a trained operator for the equipment, as well as potential usage fees for third parties (e.g. spectrum or point-of-presence licence fees). Thus managing a shared pool of physical resources may include set-up and tear-down of a certain physical configuration, which can make a greedy reservation of resources an expensive experience. In particular if resource consumption is non-regenerative (such as fuel consumption of an engine or vehicle) or degrading the equipment used (such as drawing battery power from a remote sensor, such as an off-shore buoy in a Tsunami warning network). In addition, a 'non-conventional' resource may be non-replaceable if used-up, thus vanishing from the resource pool after use.

  Policy management and policy enforcement as well needs to be realized on a node-local basis since policies may be dynamically enabled/disabled (or defined and deployed) according to temporary coexistence requirements valid at the time of resource instantiation. For example, operating a RAN and minimizing interference with nearby RANs may require to manage shared frequency bands, which is the usual case for wireless testbeds operating in close vicinity.

- Requirement for streamlined resource monitoring and node-local management of Service Level Agreements.

When operating shared physical resources the physical environment influences the operation of the resource. For example, a RAN may not achieve its best performance due to interference, obstacles or vegetation blocking the line-of-sight (LOS), due to a lack of base stations in a certain area, or simply due to a lack of licensed frequency bands to use. Such limitations are very common and well known by mobile operators. For a VANET, a certain road or flight path might be blocked, or a certain vehicle may be not available, and for a WSN some optimal sensors may be out of energy or in maintenance mode such that other locations must be chosen as a sub-optimal replacement. In consequence, a monitoring infrastructure regarding 'non-conventional' resource infrastructures is dynamic for the same reasons that apply for the 'non-conventional' resources themselves. Any decision taken by the resource management (e.g. reassigning sensors) also reflects in the need to (re-) define a matching monitoring resource at instantiation time. For being able to perform relevant end-to-end measurements, matching monitoring resources must be allocated. This requires intimate knowledge about the allocation and use of 'non-conventional' resources such as the mobile end-system, the base stations, and the core network attachment points involved in a certain RAN set-up. This is in particular valid for monitoring compliance with SLAs, where the impact of monitoring on the monitored resource must be known (e.g. when monitoring data is also using a wireless channel). Choosing inappropriate monitoring locations will produce unsuitable results not matching the real situation.

- Requirement for a dedicated user interface for 'non-conventional' resource management and control.

  The details for 'non-conventional' resources provided by a certain node's infrastructure are usually non-disclosed towards a federated resource manager due to the complexity and diversity of such resources. For example, a descriptive resource specification for use by a federated resource manager would require cross-domain ontologies. This also applies to an associated monitoring infrastructure. In consequence a federation portal, which provides a number of user interfaces (UIs) for security, monitoring, identity management, authorization, resource management and control, or similar dashboards lacks versatility to cover also 'non-conventional' resources. A suitable 'non-conventional' resources UI thus should be provided by the node providing the resources. The 'non-conventional' resources UI can be linked into the federation portal when the user requests a certain service from the federation, or when a particular Generic or Specific Enabler is selected for deployment. This UI can be made dynamic with respect to only offering selections that can be provided at the time a request is issued. The burden of UI flexibility is thus removed from the federation portal and put onto the node having available up-to-date information needed to build a matching UI. In consequence, a dedicated node-local 'non-conventional' resources UI can present a node's offerings regarding 'non-conventional' resources as a particular node service integrating the various aspects of a resource (i.e. authorization, reservation, configuration, policies, utilization, monitoring, ...) into a single dynamic 'landing page'.

As the previous description highlights, given the complexity and heterogeneity of the domain related to ‘non-conventional’ services, at least in a first phase of XIFI we foresee a loose integration between XIFI and these ‘non-conventional’ resources. This means that this type of services will be advertised on the XIFI portal/marketplace (see REQ-3.1) but the resource management and the usage of those resources should be handled contacting directly the infrastructure owner and by-passing the XIFI Federation (here is applicable the one-stop-shop federation model).

Later, it is possible that some of the constraints arising from heterogeneity of resources and also terms of access may be addressed under the self-service provisioning model by extending the existing requirements (see SE_Req8 above). This will be analysed in the next iteration of WP1 and covered in D1.4 and D1.5.

 

# 3 FEDERATION MODEL AND ARCHITECTURE

The first draft of the federation architecture has been provided in D1.1. Here we describe the improvements of that federation architecture taking into account the evolution of the XIFI project in the last five months but also the changes made to the FI-WARE project. The following FMC diagram depicts the current status of the federation architecture (as of 31st January 2014).

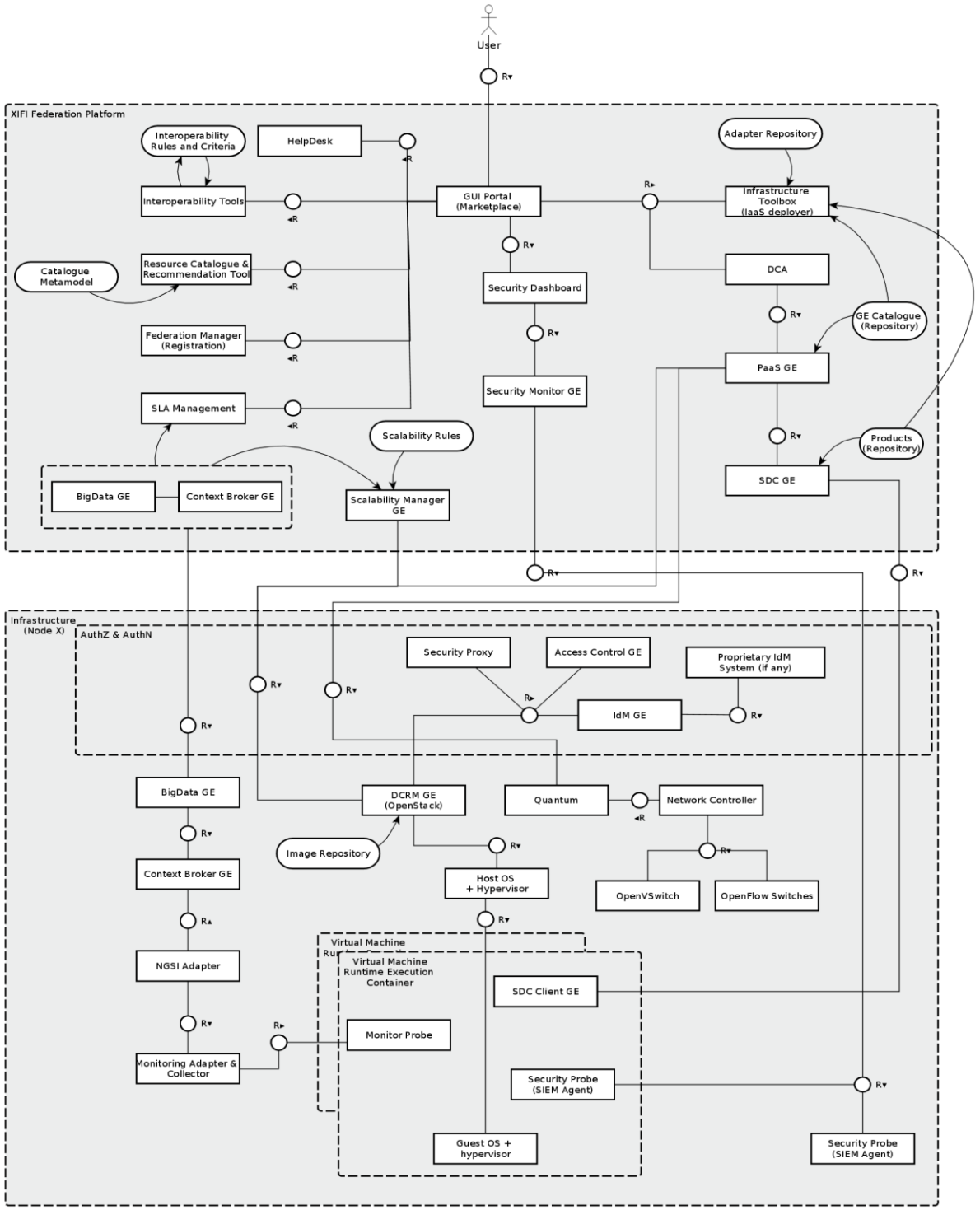


*Figure 9: XiFi architecture*

The bottom part of the figure contains the architecture and the components deployed on each node of the XIFI federation (both master and slave) whereas 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). 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 FI-WARE documentation.

## 3.1 Architecture of a generic XIFI Node

Three main functional groups can be identified in the bottom part of the previous figure

1. Components enabling cloud computing
2. Components enabling monitoring functionalities
3. Components enabling security functionalities.

### 3.1.1 Components enabling cloud computing

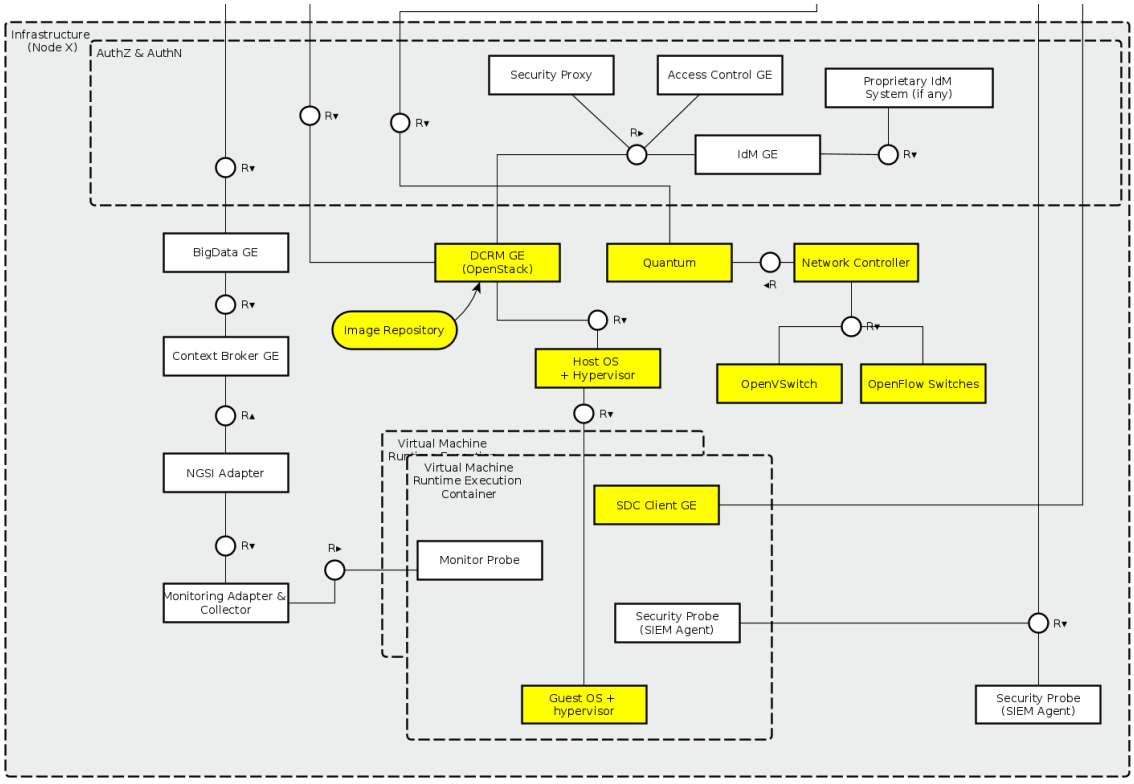


*Figure 10: Components enabling cloud computing*

These components (in yellow in the previous figure) enable the setup of a cloud computing environment based on OpenStack [4]: the *FI-WARE DCRM GE* wraps OpenStack and, together with *Quantum*, *Network Controller*, *Open vSwitch* and *OpenFlow Switches* components, provides all the services requested to a IaaS Management System. Moreover each virtual machine will be equipped with the *FI-WARE SDC GE client*, connected to the *FI-WARE SDC GE*, present only in the master node, in order to deploy different products and GEs.

 

### 3.1.2 Components enabling monitoring functionalities

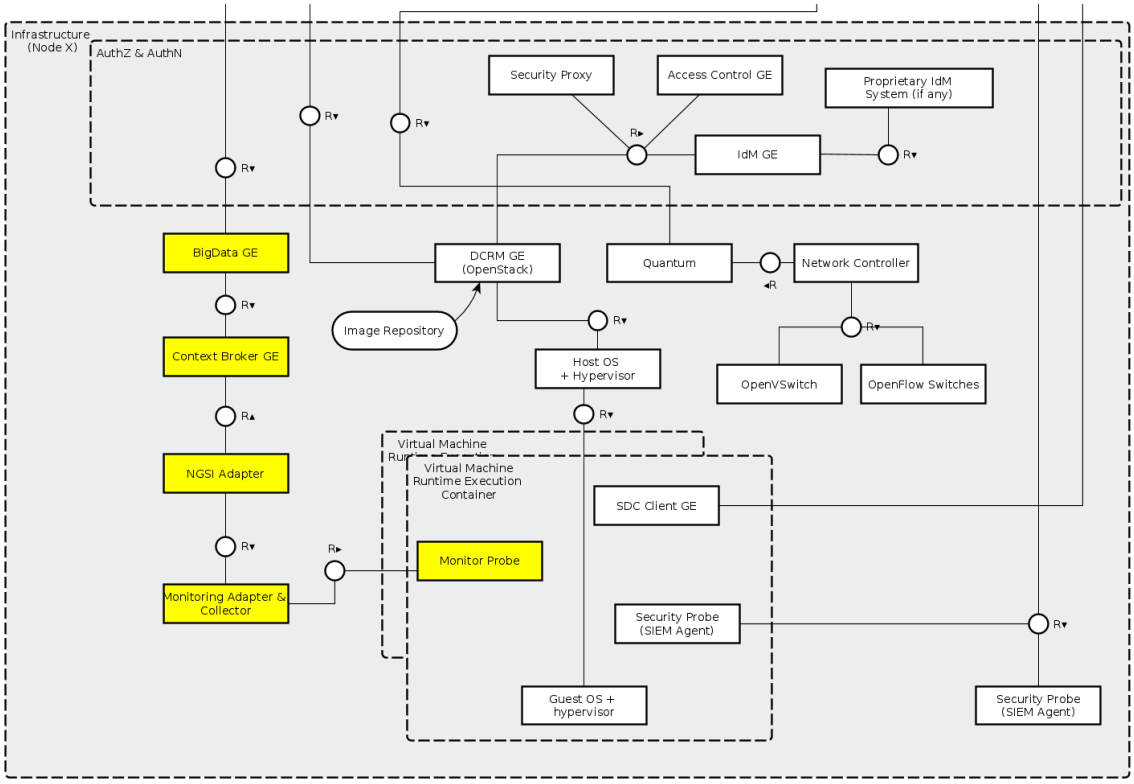


*Figure 11: Components enabling monitoring functionalities*

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 - http://www.nagios.org/) that can be managed directly by the infrastructure owners and then passed, through an *NGSI Adapter*, to the *FI-WARE Context Broker GE* and then to the *FI-WARE Big Data GE* 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.

### 3.1.3 Component enabling security functionalities

These components (in yellow in the previous figure) enable the security functionalities. The *FI-WARE IdM GE* together with the *Security Proxy*, and *Access Control GE* provide both authentication and authorization services for each node. In the future we foresee the integration with *Proprietary IdM System* (i.e. existing systems installed on the infrastructure and managed by the infrastructure owner), if present on the nodes. In this way the infrastructure owner can keep control of the security and in particular of the identity management; then the *IdM GE* will be federated with existing IdMs using the SAML protocol. *Security Probe (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 on each node of the XIFI federation will be provisioned in high availability so as to avoid any single point of failure.

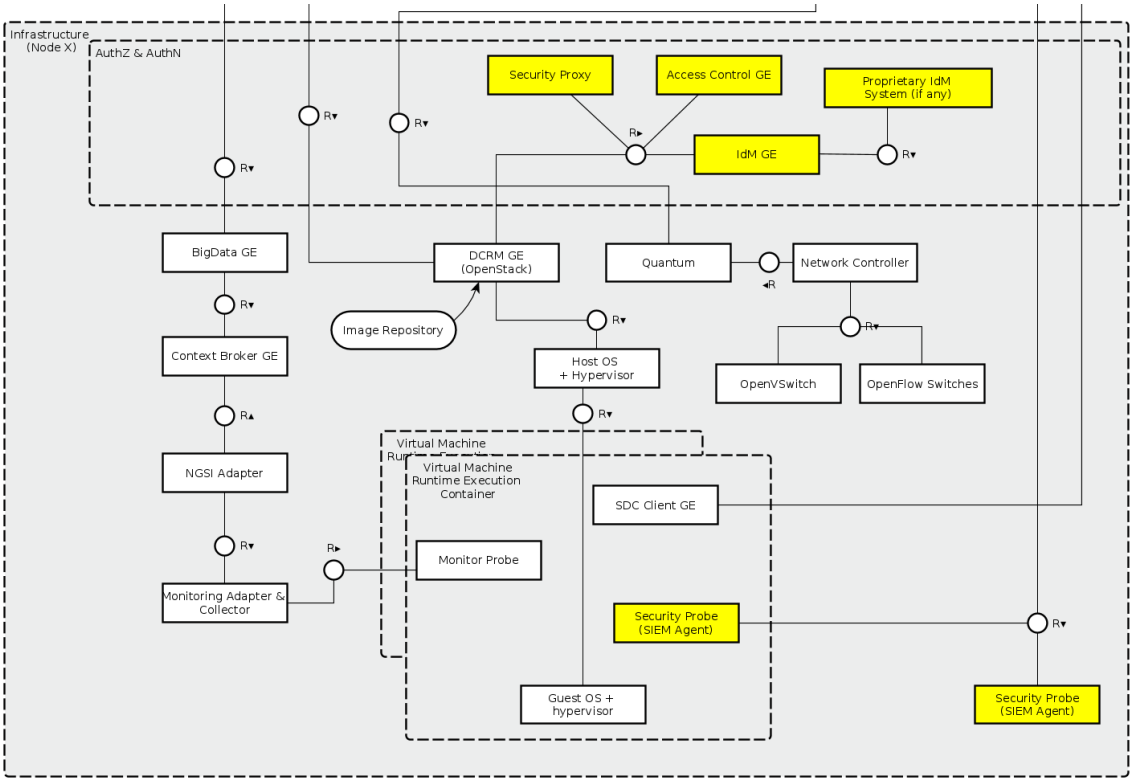


*Figure 12 : Component enabling security functionalities*

## 3.2 Architecture of a XIFI Master Node

In addition to the components described in the section 3.1, a master node comprises the following functional groups:

1. User oriented services and tools
2. Service and tools supporting the setup, deployment and operation of the Federation
3. Federation Security tools

### 3.2.1 User oriented services and tools

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* are oriented to find the right services offered by the federation; *Interoperability Tools* can verify the interoperability and compatibility of developed software with FI-WARE GEs based on some rules; *SLA Management* handles the SLA negotiation; *Federation Manager* governs the registration of a new infrastructure to the XIFI federation. Finally *Cloud Portal(Marketplace)* provides a single entry point and a graphical user interface for all these tools offering a sort of marketplace for all the services provided by XIFI.

 

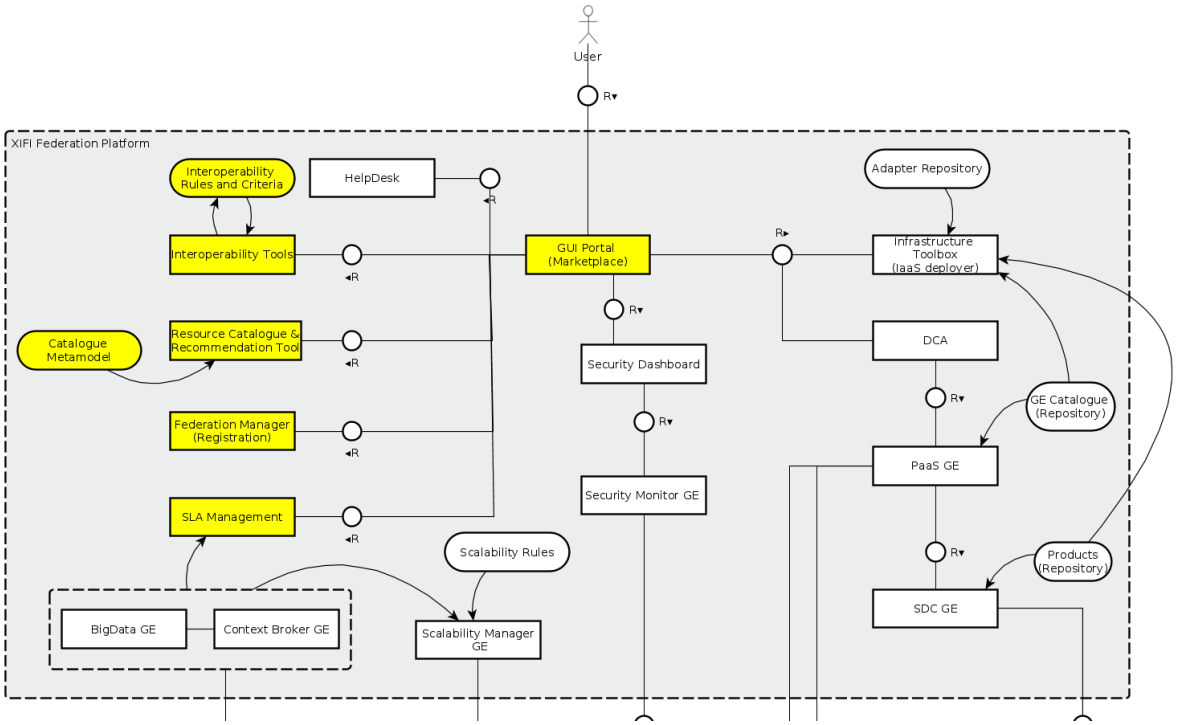


*Figure 13: User oriented services and tools*

### 3.2.2 Service and tools supporting the setup, deployment and operation of the Federation

This set of tools (in yellow in the previous figure) offers all the functionalities to deploy the software needed to install a 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 Network Controller), the components enabling monitoring functionalities (see Section 3.1.2) and the component enabling security functionalities (see Section 3.1.3). *DCA* (Deployment and Configuration Adapter), *FI-WARE PaaS GE* and *SDC GE* provide functionalities for deployment of the GEs and third parties products on the different nodes of the federation. 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 the "user oriented services and tools" (see previous section) and the Monitoring system components. The *FI-WARE Scalability Manager* implements elasticity and scalability rules. The *FI-WARE Big Data GE* and *FI_WARE Context Broker GE* 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 aggregated following different perspectives (e.g. average on time, average on resources belonging to a node, etc.). 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).

 

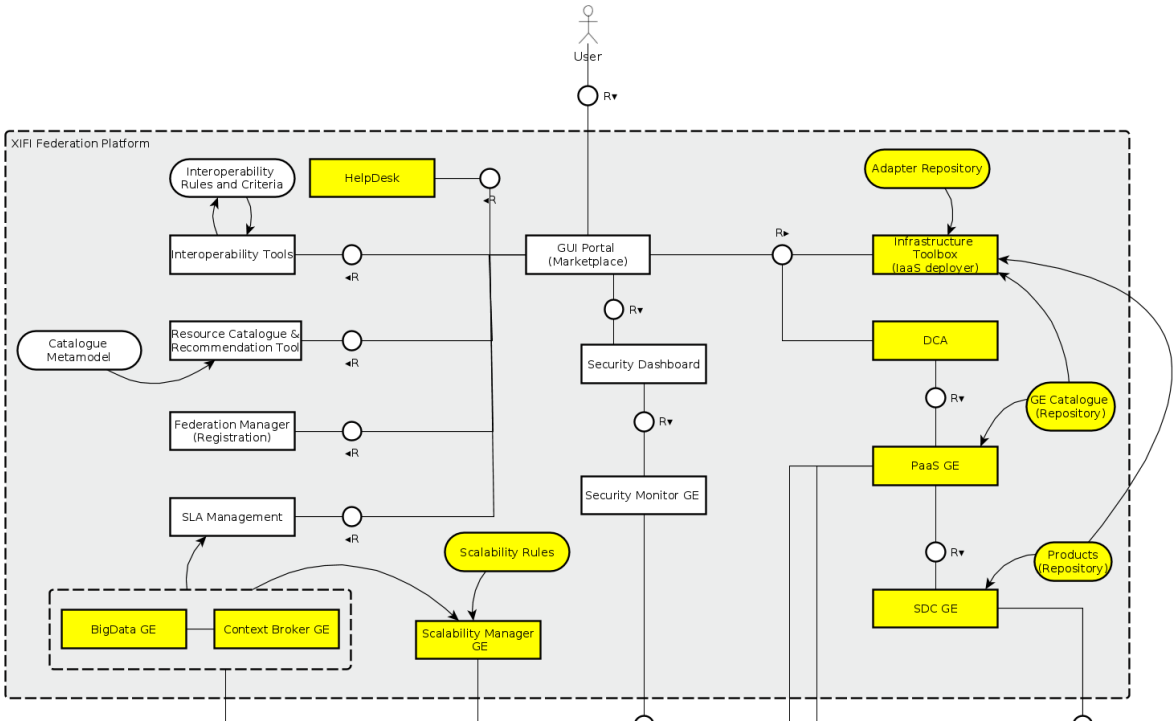


*Figure 14: Service and tools supporting the setup, deployment and operation of the Federation*

### 3.2.3 Federation Security tools

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

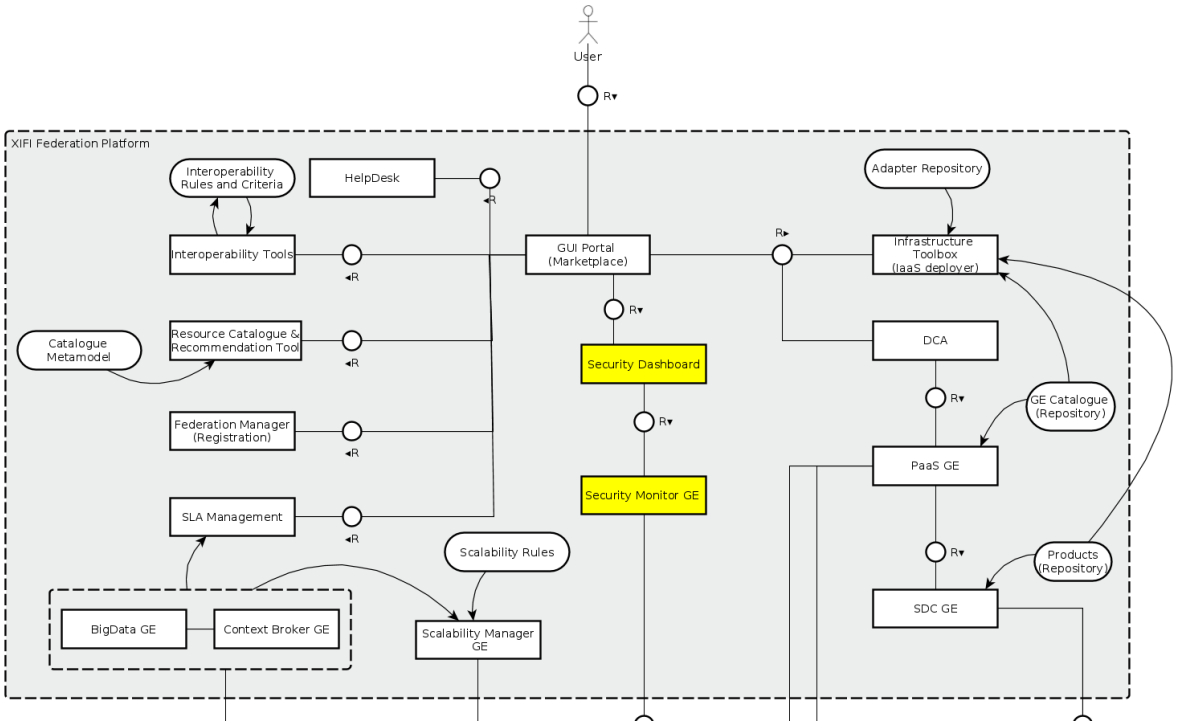


*Figure 15: Federation Security tools*

## 3.3 Nodes internetworking

The federation relies on a communication infrastructure that is heterogeneous to a certain degree. This heterogeneity is partially screened through the OpenStack networking model that allows in principle to interconnect sites (or XIFI infrastructures) on OSI layer 3 (L3) or layer 2 (L2). The basic methods are

- on **L3** through IP routing, or
- through **L2 in L3** encapsulation (e.g. generic routing encapsulation, GRE tunnels, IETF[5],
- on **L2** through partitioning by virtual LANs (VLAN, IEEE 802.1Q).

The selection of the connectivity is particularly important to enable private virtual networks across nodes. For example, some of the complex aspect related to this issue may be simplified with a L2 solution (e.g. allows for a single DCHP server and addressing plan across nodes), other issue may raise such as scalability using L2, that is better supported in the case of L3. A discussion on the specific issue is available in D3.1. Outside this specific issue, most of the federation services and OpenStack APIs requires an IP based communication (L3).

Both IPv4 and IPv6 could be used on L3 although IPv6 support by current OpenStack releases still has to be verified. This causes the need for establishing a dedicated address plan and, potentially, to define federation wide dynamic address management and address resolution services similar to (D)DNS or DHCP. The latter are related to the need for describing entities in the federation such that entities are searchable and described by a unique reference that can be linked to a human readable or descriptive identification format (e.g. linking a hostname and a VM instance serial number). The benefit of using L2 interconnects is not yet fully explored, but layer 2 forwarding (flat-L2), tunneling

 

layer 2 in layer 3 packets (L2oL3 overlay, virtual L2 domains) as well as hierarchical networking solutions (L3 routing between L2 domains) are considered state-of-the-art in current datacentre interconnect solutions. Similar requirements as for L3 interconnects arise when using L2 solutions, for example instead of planning and managing IP address ranges, MAC Ids of virtual network interfaces must be managed to avoid conflicts between exposed VMs.

Given the heterogeneity of nodes and network services available to them, we define as reference solution the usage of L3 for the data plan adopting the MD-VPN service by GEANT. MD-VPN provides a scalable solution for L3 multi-domain networks. The experiments on MD-VPN will be detailed in D5.2. A L2 management network is provided through L2 in L3 encapsulation. Specific nodes may require different solutions, for example because cannot access the MD-VPN trial. Detail on the reference solution and variants adopted will be provided in D5.2.

## 3.4 Summary of architectural decisions and rationale

The following table provides a list of the main architectural decisions together with the corresponding rationale that drove the choice:

| Decision | Rationale |
|---|---|
| Few Master nodes, many Slave nodes | • have Federation layer provided in high availability<br>• avoid to complicate too much the synchronization of information as in case of a peer2peer architecture. |
| Security system on each node | • avoid single point of failure<br>• keep it scalable<br>• provide integration with existing security systems<br>• can provide identity federation |
| Monitoring system on each node | • collect data from the given node<br>• provide scalability<br>• can connect with existing monitoring system |
| Monitoring system at federation level | • aggregate monitoring data coming from all nodes<br>• provide a federation view |
| Infrastructure Toolbox, Federation Manager | • provide a set of tools to deploy a XIFI node easing the joining federation process |
| User Tools (Resource Catalogue & Recommender, SLA Management, Cloud Portal) | • provide a transparent view (not coupled to a given node) at the federation level of the services offered by XIFI |

*Table 6: Summary of architectural decisions and rationale*

## 3.5 From Use Cases Scenarios to Requirements and to Architectural Components

This section provides a mapping between use case scenarios, requirements and architectural components. Up to date mapping can be found on Redmine (see 2.2).

| Use Case Scenario | Requir. | Architectural Component | Comments |
|---|---|---|---|
| UC-1 | REQ-1, REQ-2, REQ-6 | Infrastructure Toolbox, Federation Manager | OpenStack, DCRM GE, Network Controller, Monitoring System, IdM GE and other Security components will be deployed and configured by the Infrastructure Toolbox component. Federation Manager governs the registration process. |
| UC-2 | REQ3, REQ-4, REQ-5, REQ-7, REQ-8, REQ-12 | Cloud Portal, Resource Catalogue & Recommender, Interoperability Tools, DCA, PaaS GE, SDC GE, DCRM GE, Network Controller, Scalability Manager | Through the usage of Cloud Portal, Resource Catalogue & Recommender, DCA and the PaaS and SDC 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 (see Section 2.4) 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 and Big Data GE | 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 System 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 | Help Desk, Federation Manager | User can submit a support request through the Help Desk component. Moreover through Federation Manager it is possible to handle joining operations of the XIFI federation. |
| UC-5 | REQ-9, REQ-11 | Context Broker GE, Big Data GE, SLA Management, Monitoring Adapter & Collector, Security Monitoring GE, Security Dashboard | 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 Dashboard (Cloud Portal). Security Monitoring is made through Security probes, Security Monitoring GE and Security Dashboard. |

*Table 7: Technical Requirements Tracing*

# 4 FEDERATION FUNCTIONALITIES

## 4.1 Overview

This section analyses individual federation functions. This discussion is based upon the current XIFI architecture and requirements as presented in the previous sections, the state of the art in federation management, and the features provided by FI-WARE Generic Enablers. The objective here is to present from a technical perspective the features that are commonly implemented in a federation. The discussion on the actual adoption (and on the level of adoption) of these features is currently ongoing and will be finalised in the next deliverables. This content hence serves as general knowledge for input to the overall XIFI architecture design process that involves all of the related work packages in the project.

## 4.2 Security functions

### 4.2.1 Identity management

Identity management (IdM) is the process of assigning identities and other attributes to individual federation users. Federations often support 'cross-domain' identity management, i.e. multiple domains have users, whose identities must be understandable (and verifiable) in more than one domain. In this situation, the domains can form an identity management federation of which they are members, and this federation supports cross-domain identity management in a variety of ways, including but not limited to the following:

- the federation establishes its own identity management system which all domains can understand, and domains to register users needing cross-domain identities with it;
- the federation provides token exchange services, and defines mapping rules between identities and attributes from each domain and some understandable federated equivalents;
- the federation defines mapping rules but leaves each domain to decide whether and how to use them to understand identities specified in other domains.

The first of these is a highly centralised approach, in which the federation acts as a single domain and manages the cross-domain identities on behalf of individual members. The last is a decentralised approach, in which each domain assigns identities to its own individual users, and each domain decides for itself whether and if so how to recognise identities assigned in other domains – the only role of the federation is to define the rules. The middle option is the centre ground, in which identities are assigned by each domain, but the federation provides the means for other domains to understand them.

Our concern here is to establish what models can or should be considered in the XIFI federation, and how these may be implemented using the FI-WARE core platform in conjunction with XIFI or (if and only if necessary) by using other available technologies.

The XIFI approach requires that the Federation is able to issue its own identity tokens. In this way an identity model exists at the level of federation and it will be common throughout the different nodes. Hence the second and third approaches described above do not fit in the XIFI landscape; this decision is explored further below.

IdM development and deployment is composed of multiple steps. The first step is to provide a centralized IdM approach, ensuring that the IdM system is available in any of the nodes. This access point serves any type of end-user: the infrastructure owners, the software developers and the application end-users. A later step is to introduce a redundancy mechanism for preventing failure. Finally, SAML 2.0 is supported and therefore allows "third party" authentication. This feature will

 

partially implement the federated IdM support, limiting it to the authentication, while it will be left to any of the infrastructure to manage the identity and the roles associated with an issued token. The following table summarizes these steps:

| Feature | Description |
|---|---|
| Support Web API based registration of a new region and related services | A portal administrator should be able to register through Web APIs a new region and the related open stack services |
| Support for High Availability | The IdM should not be a single point of failure and each region, as regarding locally deployed services, should rely on a local IdM as master. A developer should be able to access services on a node even though the master node is offline. |
| Support for SAML 2.0 | A user registered on another IdM system should be able to authenticate locally if his IdM system is trusted by the local IdM. |

*Table 8: Roadmap for IdM implementation*

With this approach the identity itself (i.e. the set of attributes composing an identity) must be defined at the federation level and while currently the plan is to simply replicate the model across the nodes, some local mapping should exist for allowing a local conversion from the standard federation-based SAML assertion and the local user identity. This approach is necessary if an infrastructure wants to port its old users to the XIFI federation. The mapping is highly dependent on local existing models that can be very different, so the best solution is to leave the infrastructure to create its own mapping, while all the federation nodes use the common-defined shared identity data model.

### 4.2.2 Generic Enabler Usage

Identity Management encompasses a number of aspects involved with users' access to networks, services and applications, including secure and private authentication from users to devices, networks and services, Authorization & Trust management, User Profile management, Single Sign-On (SSO) to service domains and Identity Federation towards applications. The Identity Manager is the central component that provides a bridge between IdM systems at connectivity-level and application-level. Furthermore, Identity Management is used for authorizing foreign services to access personal data stored in a secure environment also supporting the necessary consent-giving procedure. Identity Management is used in multiple scenarios spanning from Operator oriented scenarios towards Internet Service Providers (ISP). End users benefit from having simplified and easy access to services (User Centric Identity Management).

Currently there are 4 implementations of the IdM GE in FI-WARE. These are One-IdM and Digital Self IdM GEis from NSN, GCP IdM GEi from DT and KeyRock IdM GEi from UPM. As for the last one (also the latest one) called KeyRock developed by UPM in the context of OpenStack and FI-Lab integration as Open Source and deployable as on premise implementation of the IdM GE since this was also required (e.g. Cloud/FI-LAB, XIFI, FI-STAR, FI-Content2). KeyRock solution, that replaces the IdM service by OpenStack - KeyStone, allows also to manage in a unified way IaaS resource, SaaS resources and end-users applications (a requirement posed by FI-Lab). Since this version meets XIFI requirements, then it would be the one used in the context of this project.

#### 4.2.2.1 Future Opportunities: Distributed IdM Management

Identity Management is a key part of the XIFI architecture, and therefore it is important to consider possible extensions towards a more flexible implementation. One such option is to support a more

 

distributed IdM approach as previously discussed. Indeed the current direction of the implementation (i.e. through SAML or OAuth delegation) will facilitate this, simply by defining and implementing the necessary operational protocol.

To highlight the benefits of a more flexible IdM architecture, we present three different scenarios:

- Applications defining a custom user model on a node rely on the federation for the propagation of the information to the other nodes, so that the custom model is meaningful across all the nodes on even if it has been created on a specific one.
- Nodes having local or pre-existing users receive support from the federation so that through mapping mechanisms the node user may easily become federation users. In this way a user exists across nodes, even if it was created before on a single one.
- External trusted sources may be added to the federation so that users authenticated through these sources are directly authenticated also in the federation.

### 4.2.3 Authentication

Authentication is “the act of verifying the identity of an entity (subject)” [5]. TrustInCyberspace adds the term “level of confidence” to this definition: Authentication is the process of confirming a system entity’s asserted identity with a specified, or understood, level of confidence.” This definition contains all necessary parts to examine authentication in a broad sense. First of all it does not restrict the authentication to human users, but refers to a generic “system entity”. Secondly it introduces the often neglected concept of “level of confidence” which applies to each authentication of an identity. No computer program or computer user can definitely prove the identity of another party. There is no authentication method that can be secured against any possible identity-theft attack, be it physical or non-physical. It is only possible to apply one or more tests, which have been previously defined as sufficient to discriminate an attack from a legal access. The problem is to determine which tests are sufficient, and many such are inadequate. This leads to the general field of claims and trust management, because authentication could also mean to verify the “author” / issuer of any claim. The confirmation or validation process of authentication is actually done by presenting some kind of proof. This proof is normally derived from some kind of secret held by the principal. In its simplest form the participant and the authentication authority share the same secret. More advanced concepts rely on challenge/response mechanisms, preventing the secrets being transmitted. Refer to Authentication Technologies for a detailed list of authentication methods used today. As stated above, each authentication method assures only some level of trust in the claimed identity, but none could be definitive. Therefore it makes sense to distinguish the different authentication methods by an associated assurance level, stating the level of trust in the authentication process. As this assurance level depends not only on the technical authentication method, but also on the overall computer system and even on the business processes within the organization (provisioning of identities and credentials), there is no ranking of the authentication methods here. Authentication protocols are used to exchange authentication data between the client and server application. A single authentication protocol supports one or more authentication methods.

#### 4.2.3.1 Generic Enabler Usage

The KeyRock FI-WARE Identity Management GEi from UPM is free/open source software which complies with existing standards for user authentication and provides access information to services using it as a Single Sign-On platform. It supports OAuth 2.0 and HTTPS, and in the near future will be supporting SAML 2.0 and federation between different instances of IdM GEs.

#### 4.2.3.2 Future Opportunities

Currently no future opportunities have been defined for the authentication.

 

### 4.2.4 Access control

Access control is the prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner. More precisely, access control is the protection of resources against unauthorized access; a process by which use of resources is regulated according to a security policy and is permitted by only authorized system entities according to that policy [6].

The basic concepts come from the OAuth and XACML 2.0 standards. The OAuth standard is supported by the Identity Mangement GE essentially, the OAuth Authorization endpoint and Token endpoint in particular. The Access Control GE is only a consumer of OAuth tokens, therefore it only supports validation of OAuth tokens and getting authorization info from the token, such as user attributes. The XACML standard is only supported by the Access Control GE, using RESTful APIs.

The classical architecture of an Access Control system requires the definition of a Policy Decision Point (PDP) which evaluates access requests against authorization policies before issuing access decisions and of a Policy Enforcement Point (PEP) which intercepts user's access request to a resource, makes a decision request to the PDP to obtain the access decision (i.e. access to the resource is approved or rejected), and acts on the received decision.

The PEP (Policy Enforcement Point) can be deployed in several ways, including:

- Use with an HTTP Proxy. In this case, the owner of the Resource Server must provide the necessary information to the PEP proxy owner for integration with the Resource Server.
- Embedded into the (federated) resource. In this case, the PEP implementation is the responsibility of the owner of the Resource Server.

In all cases, the PEP would have to integrate with the IdM Generic Enabler for OAuth token validation, and the XACML PDP for requesting authorization decisions.

#### 4.2.4.1 Generic Enabler Usage

UPM has already developed for FI-WARE an Access Control Proxy that supports OAuth2 and integrates with their IdM GE implementation ‘KeyRock’; UPM is also currently working on the XACML connectivity of the proxy to the Access Control GE from Thales to support authorization based on its rules as well[6].

In the context of the XIFI and taking into account the work done by UPM on the Access Control Proxy and future work planned to fully connect to the Access Control GE, the second of the two options described above would be the preferred approach for PEP deployment. In the XIFI architecture described previously, the Proxy is referred to as *Security Proxy*.

#### 4.2.4.2 Future Opportunities

Currently no future opportunities have been defined for the Access Control.

## 4.3 Resource management functions

### 4.3.1 Resource monitoring

Monitoring is a critical part in every cloud environment, allowing both infrastructure owners and application developers/users to collect monitoring data regarding network devices, servers (either physical or virtual) and services. In a federated cloud environment, like XIFI, state-of-the-art solutions are not directly applicable at the moment. Thus, one has to determine the requirements and

[6]The Access Control proxy component is also open source and can be downloaded from https://github.com/ging/fi-ware-pep-proxy

carefully select and develop tools and procedures needed to support monitoring of several resources. To this end, resource monitoring functionality stems from the REQ-9, described above. From this perspective, resource monitoring requirements must be translated to specific functionalities and development steps to accomplish them:

**1. A precise data model has to be defined**.

A well-structured data model is needed in order to provide monitoring data in a meaningful way. The data model must include:

- the monitored services,
- the resources allocated for the deployment of a GE,
- processes running,
- specific information from physical and virtual servers, such as RAM utilization, CPU usage, network monitoring measurements and so on.

Moreover, the data model must take into consideration similar effort spent in the FI-WARE project in order to provide a compatible and unified approach throughout the FI-PPP project[7].

**2. The collection of monitored data on node level must be independent from installed monitoring tools**.

In a federated service cloud environment, each infrastructure owner might already be using his own monitoring tool. State of the art solutions include: Nagios, Zabbix, OpenNMS[8], etc. According to the respective analysis described in D1.1 [1], among the five nodes comprising the XIFI federation, there are several different monitoring tools utilized by infrastructure owners to monitor services summarized in the following table:

| Node | Monitoring Tools |
|---|---|
| Trento | OpenNMS |
| Berlin | It has already installed Nagios/Zabbix |
| Brittany | Nagios/cacti |
| Waterford | Nagios/Ganglia |
| Sevilla | It is open to any monitoring tool |

*Table 9: Monitoring tools used in XIFI nodes*

Moreover, as new nodes will join the XIFI federation, it is likely that other monitoring tools will be added to this list. Thus, the developed monitoring solution must capitalize on currently installed solutions and provide a tool that will collect monitored data from several tools.

**3. Provision of monitoring data in both push and pull manner**.

An infrastructure owner and any interested user must be able to query on a periodic basis or upon request. Moreover, one must be given the opportunity to select among a plethora of monitoring data

[7] https://www.fi-ppp.eu/projects/

[8] www.nagios.com, www.zabbiz.com, http://www.opennms.org/ .

 

related to his own specific interest, i.e. SLA. This functionality must be taken into account during the design and development phase of the XIFI monitoring solution.

**4. Compatibility with XIFI modules on federation level**.

Apart from the requirements at the node level, as described above, there are also specific functionalities that have to be addressed in order to cope with the federation aspect of XIFI. Monitored data must be stored both locally (per node) and in the Master node, providing also means for aggregation, filtering, etc. The monitoring solution must be connected with other modules developed within XIFI, such as the Scalability Manager .and SLA Management.

#### 4.3.1.1 Generic Enabler Usage

Currently XIFI bases its monitoring solution on FI-WARE GEs (Context Broker GE, Big Data GE) together with some specific adapters developed inside XIFI.

#### 4.3.1.2 Future Opportunities

The FI-WARE Monitoring GE appears to be a promising solution that could be used to standardise the monitoring data format (NGSI9/10), and provide more monitoring and transparency features for application developers and operators as well as helping to unify developments across FI-PPP projects. It was decided that XIFI will follow an abstract approach, decoupling monitoring data collection from the monitoring tools. In this perspective, a specific component/adapter will gather data from several sources, offering the opportunity to current and future infrastructure owners joining XIFI federation to easily integrate the data arising from their own, already installed monitoring tool.

### 4.3.2 Resource discovery

The main goal of Resource discovery is to help the user to discover and compare the available resources (enablers and advance capabilities) that are available in the XIFI federation environment. The federation members (infrastructure owners) should publish/advertise their resources/services in order to allow the user to find and access the XIFI federated resources/services.

Resource discovery needs to be able to access to the information of the available services in order to discover what the user needs. This information comes from the infrastructure owners and it must flow through the federation environment. To provide these functionalities, we can consider different approaches:

- Decentralised model: all the information of the services is contained only by the federation members and the federator only aggregates the different way to access it. The shown information doesn't have a common description.
- Distributed model: part of the information is centralized, following a common structure. So, the main information, to provide the discovery and comparison, is unified and centralized. The service details are described by each federation members.
- Centralised model: all the information about the federated services is centralized, following the same structure. In this case the whole information is unified and published homogeneously and it is not different among all the federation members.

These approaches are aligned with the different definitions of the Federation Models in 2.1. The first of these is a highly decentralized approach, in which the federation acts as a dispatcher/browser and the information is stored only on the federation members and there is no uniformed description. The last is a highly centralized approach, in which the federation layer contains all the information in a common structure, and the federation member always synchronizes all their data on the federation environment. The middle option is a distributed data, in which not all the data have been centralized in a common description and part of this are on the federation members side.

 

The following table provide advantages and disadvantages of each approach.

| Approach | Advantages | Disadvantages |
|---|---|---|
| Decentral. model | • The federation members always have the information and it is not necessary to synchronize the data through the centralized repository. | • There is no common metadata, so that it is complicated to show them homogeneously and compare them.<br>• When the number of the federation members increases, the performance of the central dispatcher will be decreased.<br>• It is the responsibility of every federation member to maintain high availability, since the centralized module is only responsible for dispatching the particular calls. |
| Centralised model | • The service description has the same structure for the different federation members; this allows showing all the information homogenously and comparing them.<br>• The high availability is centralized in the federation layer.<br>• The increase of the federation members doesn't penalize the performance. | • A synchronization system is needed to actualize the centralized information.<br>• It is necessary to map the whole description of the resources, following the common schema.<br>• The initial boot in order to populate the central repository is heavy. |
| Distrib. model | • The main information of the services has the same structure for the different federation members; this allows showing part of the information homogeneously and comparing them.<br>• It is modular and scalable to increase or modify the common structure in order to cover all the user necessities.<br>• The high availability is centralized in the federation layer. | • It is necessary a synchronization system to actualize the common information.<br>• It is necessary to map the part of the resource description following the common schema.<br>• The initial boot is heavy, nevertheless it is directly associated to the size of the common data. |

*Table 10: Resource Discovery Different approaches*

After this analysis with the identified requirements, we can consider that the best approach should be a mix of *Distributed* (related to non-conventional resources*)* and *Centralized* model.

#### 4.3.2.1 Generic Enabler Usage

There are several FI-WARE GEs that, when combined, can create a centralized module that contains the common description of the services and will allow users to discover and compare the federated XIFI services. We can consider basing our development on the GEs associated to the Application/Services Ecosystem and Delivery Framework in particular the Store [7] and the Repository GEs.

#### 4.3.2.2 Future Opportunities

Currently no future opportunities have been defined for the resource discovery.

### 4.3.3 Resource allocation

The resource allocation should maximize and optimize the use of resources. In the context of the federation it would be appropriate to avoid waste of resources as a result of their underutilization and avoid long response times due to excessive use.

Resource allocation, intended as "conventional" resources like CPU, RAM, storage and network facilities can be handled directly by the XIFI federator without any intervention of the federation members (infrastructure owners) taking the "integrator" model as a reference (see 2.1).

On the other hand when "non-conventional" resources are concerned, the resource allocation should be managed directly by the infrastructure owner (i.e. federation member) by-passing the XIFI federator taking the "one-stop-shop" model as a reference. The different approach and model used in the two cases is due to the complexity to handle the idiosyncrasies of non-conventional services allocation at the federation level: in particular in case of non-virtualizable resources, their allocation and sharing should be carefully considered.

#### 4.3.3.1 Generic Enabler Usage

XIFI currently base resource allocation through the FI-WARE Cloud-Hosting GEs for cloud services and the Store GE in combination with the IdM GEs to allocate SaaS deployed GEs.

#### 4.3.3.2 Future Opportunities

As a possible enhancement of the resource allocation system as it is conceived today, we foresee the support to the definition and set up of quotas (in term of number of VMs, RAM size, disk size etc.) for each user/tender of the XIFI federation. This will be done also considering the solution of Quota Management as currently under development in FI-WARE.

Another possible extension is the integration in XIFI of tools, provided by the federation members, for the management of non-conventional services beyond the simple help desk based solution selected for now.

## 4.4 Usage management functions

### 4.4.1 Usage monitoring

Usage monitoring differs from resource monitoring because the goal is on the measurement of what the user does (monitoring a user's activities involving many resources), rather than what the resources are doing (when used by potentially many different users). From a technical perspective, some of the required functionalities are common to both types of monitoring, but others are specific for the usage monitoring. For instance, usage monitoring should monitor cases where a user fails to get access to the requested resources, as well as cases where resources are successfully allocated. This obviously cannot be detected by monitoring the resources.

The monitoring of the usage of resources in a federated environment, like XIFI, can be a complex problem, depending on the architecture and operational model adapted. Usage monitoring pertains to UC-5 "Network and Data Centre operations" and is covered by REQ-9. The key issues to be considered are:

- Resource usage lifecycle: what constitutes usage of a resource? The simplest approach is to monitor the allocation and de-allocation of resources, so they are considered 'used' by a user while they are allocated to that user.
- Resource granularity: at what level is usage to be monitored? For conventional resources, this can be covered by typical parameters (e.g. storage capacity, processing power) or profiles as defined by FI-WARE. For some types of resources one may also be concerned with the intensity of usage, but this fluctuates in real time so it may not be practical to monitor usage in such detail.
- Traceability of users: clearly it is necessary that resource allocation requests can be traced back to a registered, authenticated and authorized XIFI user.

 

Data collected on resources usage should be aggregated on a per user basis in order to monitor the usage of the resources by a given users. Also the failures during the usage of a given resource (attempt to perform an operation that failed) should be considered.

Non-conventional resources present greater challenges, both in terms of what constitutes usage, and how the resources are allocated. Currently this is handled via manual interaction with the resource owner, independently of the federation, so the analysis of these issues has been deferred.

#### 4.4.1.1 Generic Enabler Usage

FI-WARE Cloud Hosting GEs currently do not provide complete information on resource allocation requests (especially the unsuccessful ones). Future evolution of FI-WARE will be considered if possible.

#### 4.4.1.2 Future opportunities

The FI-WARE Cloud Hosting GEs, together with the GEs used in order to implement the monitoring system in XIFI (see 3.1.2) are good candidate also for implementing the usage monitoring functionality. But as in case of resource monitoring, adapters and data collectors should be developed in order to gather the relevant data.

The DCA (see 3.2.2) component developed in XIFI would be a good candidate for gathering and collecting resource allocation requests. This would allow unsuccessful as well as successful resource requests to be monitored and aggregated on a per-user basis.

Non-conventional resource usage monitoring at federation level will become an issue if in future they can be provisioned automatically via the federation, based on terms specified by the owner, rather than by manually contacting the owner (see SE_Req8 in Section 2.3). In that case, consideration should be given to monitoring non-conventional resource usage by the federation to verify that the owner's terms of access have been respected.

These options will be further analysed and elaborated in D1.5.

### 4.4.2 SLA Management

The main goal of SLA management is to ensure Services Levels, through agreeing terms and quality of service between the Federator, Federation Member and Users. It will be used by the Federator (XIFI Federation) together with the Federation Member (Infrastructure owners) in order to define the characteristics and QoS for their services and the Users for monitoring and following up of those SLAs.

Two different models can be considered with regards SLA Management.

(1) *Federation provides centralized SLAs management*: The SLA is agreed between the user and the federation as an entity. The federation deals with the federator members in order to provide a unique SLA to the user.
(2) *The federation provides P2P SLA setup:* The SLA is agreed between the user and every single federator member involved in a composition service. The federation layer is responsible to ensure there is communication between users and every infrastructure owner, if it has been defined an associated SLA.

 

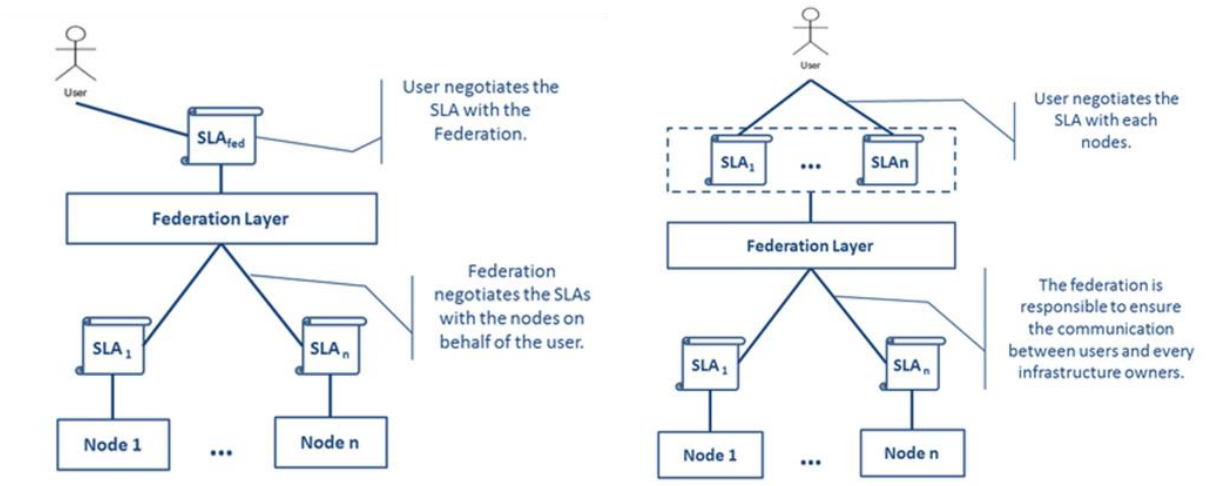


*Figure 16. Centralised versus P2P SLA management*

Both models are similar in terms of their requirements; nevertheless the implications for the federation point of view are different. These approaches are aligned with the different levels of federation models previously described. The first of these is a highly centralized approach, in which the federation layer acts as an entity, hiding the federation member's negotiation. The second is a delegated approach, in which the federation layer is only responsible to facilitate the negotiation between the user and federation members.

In the case of XIFI we foresee the usage of a hybrid model covering in part aspects of the *P2P SLA setup* and the *Centralized SLA Management.* The current starting point for the scope of the project is to provide instruments and/or tools to handle in a unique environment SLA management hiding to the user the complexity of dealing with separate instruments to manage SLAs. The SLA management model may evolve in the future, and require a more prominent role of the federator, also in light of potential commercial exploitation related to accounting and billing services offered by the federation. For non-conventional services, we foresee the usage of a pure P2P SLA set-up, given that the federator, unless instrumented, is not able to monitor them.

#### 4.4.2.1 Generic Enabler Usage

Currently, there are no enablers that cover SLA management. So, it is necessary to find existing solutions for the proposed implementation in XIFI; this is based on existing state-of-the-art SLA solution such as Cloud4SOA [8].

The Cloud4SOA project supports cloud-based application developers with multiplatform matchmaking, management, unified applications, cloud monitoring and migration. It interconnects heterogeneous cloud offerings across different providers that share the same technology through the concept of adapter that provides a REST-based API for any cloud access.

This framework needs to be isolated, adapted and integrated with the rest of the XIFI components. It is also necessary to create the graphical user interface according to XIFI needs.

#### 4.4.2.2 Future Opportunities: SLAs and OLAs

A possible useful extension is to implement in the XIFI federation the so-called OLA (Operational Level Agreement) [9]. The difference between a Service Level Agreement (SLA) and an Operational Level Agreement (OLA) is what the federation as a whole is promising to the customer (SLA), versus what the different nodes promise to each other and to the federation (OLA).

It is recommended that a simple OLA model for the XIFI federation should be considered further, focusing on a small set of attributes relatively easy to check (e.g. the availability, or specific terms of

 

access). It is also clear that other discussions at business levels have to been performed and therefore that this feature is likely to be available only in the long term.

### 4.4.3 Accounting and billing

The final stages in the resource allocation and usage cycle are accounting and billing, which normally have to be implemented in accordance with SLA terms that may be in force, and depend on data from usage monitoring. These two functionalities are needed in the middle-to-long term where new partners and potentially more sophisticated business models come into play in the XIFI federation.

Accounting provides a basis for deciding which resources have been used to what extent by each accountable user. In a typical scenario, the accountable user will be the application developer or operator who requested resources from XIFI (possibly from multiple infrastructure owners) to be used for their application. In some cases, further resources will be requested in response to application user activity, e.g. where the application needs a scalable resource. These requests should still be attributed to the accountable user and captured by usage monitoring (see above).

Billing then provides a mechanism for deciding how much each accountable user should pay for the resources they have used. This is typically done by taking the resource usage data (provided by the accounting function), computing the fees due based on the terms of the SLA and communicating this to the relevant actors (the actual way of doing this is related to the chosen model and implementation).

#### 4.4.3.1 Generic Enabler Usage

Part of the functionalities included Store GE by FI-WARE could be used to support billing. The Store GE provides for accounting callbacks, rich pricing models (including pricing parts for service compositions, composite apps and mashups, pay-per-use modalities, etc.), charging and billing including integrated payment processing via PayPal, and even revenue sharing (i.e. determining from the rich pricing model how any revenue should be distributed to back-end service providers).

FI-WARE also provides a Business Modeler GE allowing users to define a business model (i.e. the terms for use of a service, and the relationship to the terms of use for components of which it is composed). There is an associated Business Calculator GE that allows revenue shares to be computed in simulation mode, allowing models to be checked (in a simplistic fashion) by the parties who will be involved.

#### 4.4.3.2 Future Opportunities: Adding Accounting and Billing

Today, XIFI is not a commercial facility and a business model has not been defined, so accountable users do not have to pay for the resources they use. Some participating nodes (i.e. resource providers) are not permitted to run services for profit, e.g. where they are publicly funded as a free service and competition with commercial providers is not considered appropriate. This is the case for many NRENs and therefore also for GEANT, for example.

However, beyond the end of the XIFI project, and certainly beyond the end of the FI-PPP Phase III, it seems likely that XIFI will want to accommodate commercial as well as free infrastructures within its federation. Nevertheless before considering any implication on the XIFI requirements and architecture of a possible billing system, it is needed to define a business and sustainability model for XIFI.

# 5 CONCLUSIONS

This document has described the first completed version of the XIFI architecture showing the extent to which reviewed XIFI requirements related to federation are addressed. Requirements coming from three sources have been considered:

- technical requirements derived from the initial analysis of use case scenarios and Use Case Projects carried out in WP1 and described in deliverables D1.1 and D1.2;
- a survey of XIFI nodes carried out more recently, to capture operational requirements and constraints related to federation management functionality;
- a review of previous federation efforts in the context of Future Internet developments from a socio-economic perspective, extracted from WP8 and described in D8.1.

The federation model chosen for XIFI falls between the "one-stop-shop" and the "integrator" model presented in Section 2.1 depending on the resource/service considered: the federator acts as an integrator for the "conventional" data centre services (computational capacity) but acts more as a broker for the non-conventional ones (sensor networks, LTE networks etc.) and users interact with the infrastructure owners directly when wishing to use such types of resources.

The current XIFI architecture fully supports the first and second of these sets of requirements as designed, and this has been shown by explicitly relating each technical requirement to the architectural features that support it. These requirements encompass the features of a federated system and the operational approaches identified as potentially desirable from the subsequent survey among the XIFI nodes.

The last set of requirements is extracted directly from WP8 and covered from a socio-economic perspective only. These requirements have not yet been fully analysed and mapped onto technical requirements that could be addressed by implementers. However, it is clear that many are already met, although others would require some extensions of the current architecture.

Different options have then been analysed for implementing federation functions and identifies options that could be considered further when planning the evolution of XIFI in Year 2. This report does not seek to define which of these options will actually be implemented in Year 2, as further analysis is needed in the next WP1 iteration. Requirements and federation functionalities currently not covered in the architecture will be analysed in the scope of D1.4 and eventual architectural solution detailed in D1.5/D1.6 following a discussion in the consortium that will be finalized by end of M12.

# REFERENCES

References are inserted as note in the same page of the citation.


[1] XIFI Deliverable (2013) D1.1 : XIFI Core Concepts, Requirements and Architecture Draft *available at http://wiki.fi-xifi.eu/Public:D1.1*

[2] 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

[3] XIFI Deliverable (2014) D8.1 Socio-economic factors affecting the FI-PPP v1

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

[5] RFC 3588: http://tools.ietf.org/html/rfc3588

[6] RFC 2828: https://www.ietf.org/rfc/rfc2828.txt

[7] FI-WARE wStore: http://catalogue.fi-ware.eu/enablers/store-wstore

[8] D'Andria F., Chulani I., Strube P., Ruland T. "Cloud4SOA Service Lifecycle Governance Framework". http://www.cloud4soa.eu

[9] The FitSM standard a*vailable* at http://www.fedsm.eu/sites/default/files/FitSM-0-2013.pdf


 

# APPENDIX: SURVEY – FEDERATION FUNCTIONS

**Type of organisation**

What type(s) of organisation do you represent?

| Type | Yes/No | Comments |
|---|---|---|
| An operator of Future Internet infrastructure | | |
| A provider of application services that make use of Future Internet infrastructure | | |
| A community of users for application services that make use of Future Internet infrastructure | | |
| A developer of applications that make use of Future Internet infrastructure | | |
| Other (please specify): | | |

Please answer the following questions considering the resources at your infrastructure that are usable within XIFI.

**1) Identity Provision**

Issuing and managing identities and attributes to users. Do you wish to provision new users in your domain (as users within the federation)?

Please place one X in the *Choice* column against the statement that best matches the needs of your infrastructure within the federation. Where existing solutions are in place (and one you wish to user), please comment in the *Existing solution* column.

| | Choice | Existing solution |
|---|---|---|
| We have no local identity management solution. XIFI should provide this feature. | | |
| We already issue and provide identities to users but we would like the XIFI federation to provide them if possible. | | |
| We would prefer to provide identities ourselves, but accept the XIFI federation architecture solution. | | |
| We provide identities ourselves, and these must be used by all our users even if they reach us through XIFI, i.e. we could not join XIFI if that meant using XIFI identities. | | |

Additional Comments:

| |
|---|
| |

### 2) Authentication

Perform authentication of a user identity.

| | Choice | Existing solution |
|---|---|---|
| We have no local authentication solution. XIFI should provide this feature, i.e. user authenticates with federator. | | |
| We already have an authentication solution but we would like the users to authenticate with the federator it possible. | | |
| We would prefer to authenticate users ourselves with our technology, but accept the XIFI federation architecture solution. | | |
| We have our own authentication solution and this must be used by all our users even if they reach us through XIFI, i.e. we could not join XIFI if that meant using its implementation. We authenticate users. | | |

Additional Comments:

| |
|---|
| |

### 3) Authorisation

Authorising whether a user has access to a particular resource at a particular time. Do you have own authorisation policies and corresponding technologies?

| | Choice | Existing solution |
|---|---|---|
| We have no authorisation policy management technology. XIFI should provide this feature. | | |
| We already have authorisation policies but we would like the XIFI federation to provide them if possible. | | |
| We would prefer use our own technologies for authorisation policies, but accept the XIFI federation architecture solution. | | |
| We must use our own technologies for authorisation policies and rules, i.e. we could not join XIFI if that meant using XIFI | | |

| | | |
|---|---|---|
| identities. | | |

Additional Comments:

| |
|---|
| |

### 4) Access Control

Enforce security policies and control access to local resources

| | Choice | Existing solution |
|---|---|---|
| We have no access control technology. User interacts with federator who enforces access control to resources. | | |
| We already have access control solutions but we would like the XIFI federator to enforce access control | | |
| We would prefer use our own access control solution but accept the XIFI federation architecture solution. | | |
| We must use our own access control solution, i.e. we could not join XIFI if that meant using an alternative. We must enforce access control policies. | | |

Additional Comments:

| |
|---|
| |

### 5) Resource Discovery

Maintain a registry of resources that can be searched by users, e.g. LDAP

| | Choice | Existing solution |
|---|---|---|
| We have no discovery or registry solution. XIFI provides a searchable registry. User interacts directly with federator. | | |
| We already have a registry but would like the user to search via the federation registry. | | |
| We would prefer use our own registry solution but accept the XIFI federation architecture solution. | | |
| We must use our own registry, i.e. we could not join XIFI if that | | |

| | | |
|---|---|---|
| meant using an alternative. User must search our resources using our registry technology. | | |

Additional Comments:

| |
|---|
| |

**6) Resource Allocation**

3. Allocating my resources to match user requirements (i.e. a match-making service).
4. Informing users that my resources match their requirements, but leaving them to contact me to get resources allocated (i.e. a recommendation service)

| | Choice | Existing solution |
|---|---|---|
| We have no resource allocation solution. XIFI provides a matchmaking service. User interacts with federation matchmaking service. | | |
| We already have a matchmaking service but would like the XIFI federation to provide if possible. | | |
| We would prefer use our own matchmaking solution but accept the XIFI federation architecture solution. | | |
| We must use our own matchmaker, i.e. we could not join XIFI if that meant using an alternative. User must interact with our matchmaker. | | |
| We have no resource allocation solution. XIFI provides a recommendation service. User must interact with federator recommender. | | |
| We already have a recommendation service but would like the XIFI federation to provide if possible. | | |
| We would prefer use our own recommendation solution but accept the XIFI federation architecture solution. | | |
| We must use our own recommender, i.e. we could not join XIFI if that meant using an alternative. User interacts with our recommender directly. | | |

Additional Comments:

| |
|---|
| |

### 7) Resource Monitoring

Monitoring the availability of resources.

| | Choice | Existing solution |
|---|---|---|
| We have no resource monitoring solution. XIFI provides this feature. | | |
| We already have a monitoring technology in place but would use the XIFI federation solution if possible. | | |
| We would prefer to use our own monitoring solution but accept the XIFI federation architecture solution. | | |
| We must use our own monitoring solution, i.e. we could not join XIFI if that meant using an alternative. | | |

Additional Comments:

| |
|---|
| |

### 8) Usage Monitoring

Monitoring usage of my resources and generating accounting data (based on SLA terms) for users coming through XIFI

| | Choice | Existing solution |
|---|---|---|
| We have no accounting solution. XIFI provides this feature. | | |
| We already have an accounting technology in place but would use the XIFI federation solution if possible. | | |
| We would prefer to use our own accounting solution but accept the XIFI federation architecture solution. | | |
| We must use our own monitoring solution, i.e. we could not join XIFI if that meant using an alternative. | | |

Additional Comments:

| |
|---|
| |

### 9) Exception Tracking

Handling of errors and bugs in the provision of resources to users; the reporting of those exceptions, and the tracking of the handling of the exception until resolution.

| | Choice | Existing solution |
|---|---|---|
| We have no exception tracking solution. XIFI provides this feature. Users interact with the federator regarding errors. | | |
| We already have a tracking and error handling solution in place but would use the XIFI federation solution if possible. | | |
| We would prefer to use our own support solution but accept the XIFI federation architecture solution. | | |
| We must use our own support solutions, i.e. we could not join XIFI if that meant using an alternative. Users interact directly with us to resolve errors. | | |

Additional Comments:

| |
|---|
| |

### 10) SLA Negotiation

Users and providers agree on a level of service that will be provided.

| | Choice | Existing solution |
|---|---|---|
| We have no SLA negotiation solution. XIFI provides this feature. Users interact with the federator to negotiate SLAs. | | |
| We already have a SLA solution in place but would like for this to be handled by the federator. | | |
| We would prefer to use our own SLA solution but accept the XIFI federation architecture solution. | | |
| We must use our own SLA solutions, i.e. we could not join XIFI if that meant using an alternative. Users must negotiate SLAs with us directly. | | |

Additional Comments:

| |
|---|
| |

### 11) Billing (Optional question)

How users pay for the resources that they utilise.

| | Choice | Existing solution |
|---|---|---|
| We have no billing solution. XIFI provides this feature. Users interact with the federator to pay for resources. | | |
| We already have a billing solution in place but would like for this to be handled by the federator. | | |
| We would prefer to use our own billing solution but accept the XIFI federation architecture solution. | | |
| We must use our own billing solutions, i.e. we could not join XIFI if that meant using an alternative. Users interact directly with us to pay bills. | | |

Additional Comments:

| |
|---|
| |

**Further Information**

If you believe that additional features (i.e. ones provided by your infrastructure) should be considered within the federation architecture, please state these here: