Project Acronym	**Fed4FIRE**
Project Title	**Federation for FIRE**
Instrument	**Large scale integrating project (IP)**
Call identifier	**FP7-ICT-2011-8**
Project number	**318389**
Project website	**www.fed4fire.eu**

D3.2 - Infrastructures community federation requirements, version 2

Work package	WP3 Infrastructures
Task	Task 3.1 Community requirements
Due date	30/09/2013
Submission date	24/12/2013
Deliverable lead	Timur Friedman (UPMC)
Version	1.1
Authors	Ciro Scognamiglio (UPMC) Michael Sioutis (UPMC) Stefan Bouckaert (iMinds)
Reviewers	Wim Vandenberghe (iMinds) Peter Van Daele (iMinds)

Abstract	This document provides high level requirements to WP2 "Architecture", WP5 "Experiment lifecycle", WP6 "Measurement and Monitoring" and WP7 "Trustworthiness" from the Infrastructure community's perspective. This input is based on both requirements originating from the 1st Open Call proposals as well as specific use cases.

Keywords	Requirements, Infrastructure, Community

Nature of the deliverable	R	Report	X
	P	Prototype	
	D	Demonstrator	
	O	Other	
Dissemination level	PU	Public	X
	PP	Restricted to other programme participants (including the Commission)	
	RE	Restricted to a group specified by the consortium (including the Commission)	
	CO	Confidential, only for members of the consortium (including the Commission)	

Disclaimer

The information, documentation and figures available in this deliverable, is written by the Fed4FIRE (Federation for FIRE) *– project consortium under EC co-financing contract FP7-ICT-318389 and does not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.*

Executive Summary

The objective of the Fed4FIRE project is to provide a federation of FIRE experimentation facilities, which will offer many benefits to its stakeholders. It is therefore necessary to gather information on requirements originating from these stakeholders when developing and building this federation.

As seen from Fed4FIRE, two facility provider communities can be identified within FIRE: the infrastructures community, which is concerned with experimentation that has to do with networking technology and protocols, and the services and applications community, in which experimentation takes place on top of the networked infrastructure. The main purpose of this document is to specify requirements from the infrastructure community's perspective in order to build a federation of FIRE facilities. This deliverable should be read in conjunction with deliverable D4.2, which describes requirements for the services and applications community.

At the start of the project, no information was available which directly originated from these communities, so at the 1st development cycle, the requirements towards WP2 "Architecture", WP5 "Experiment lifecycle", WP6 "Measurement and Monitoring" and WP7 "Trustworthiness" were based on ideas and expertise available within the consortium.

For the second development cycle, the target was to broaden the scope of input sources used for the requirements analysis. Therefore the set of requirements defined in cycle 1 was used as the starting point, but updated and extended based on inputs from other sources such as the set of proposals submitted to Fed4FIRE for experimentation in the 1st Open Call. These proposals represent real examples of the kind of experiments that will run on the federated facilities, and are considered to be a very valuable source of information regarding the needs of the FIRE community. This is clearly the first source of information originating from one group of stakeholders for Fed4FIRE.

The WP3 testbeds also interacted with their specific research communities directly in order to identify missing requirements. During Cycle 1, no specific functionality gaps could be identified in the requirements and corresponding architecture. It seems therefore that the use cases as presented in D3.1 were already good representation of the kind of experiments that the WP3 communities typically have in mind.

This first set of use cases, as defined in Deliverable D3.1, proofed to be very valuable, but did not yet push the boundaries or federation to the limit. They clearly demonstrated the value of federating testbeds by combining different technologies that cannot be found in a single testbed; or by performing the same experiment in sequence on different testbeds in order rule out environment-related biases in the results. However, these envisaged experiments were still limiting itself to the usage of a few testbeds. The same trend could be identified in the analysed open call proposals, but as in input to the second cycle architecture, it was perceived important also to derive requirements from use cases that really push the concept of federation to its limits. For this purpose, two new use cases that push these limits were specifically written for this deliverable, and included in the requirements analysis.

The requirements that were identified by all these input sources have been assembled under the functional areas existing in Fed4FIRE:

- Experiment lifecycle: including discovery, reservation and experiment control of resources and services.
- Measurement and Monitoring: covering metrics, instrumentation, data management
- Trustworthiness: security and privacy.
- Interconnection: including access networks, routing, etc.

These requirements have been prioritised according to how many proposals and use cases benefit from a requirement. The requirements with the highest priority together constitute the output of this deliverable that will be used as an input for the definition of the architecture of the second Fed4FIRE development cycle by WP2.

Table of Contents

1 Introduction

This deliverable, "D3.2 Infrastructure community federation requirements, version 2", is one of the starting documents for the 2nd development cycle in the Fed4FIRE project. The document describes the gathering of a second set of requirements from the Infrastructure community's perspective in order to build a federation of FIRE facilities. It is the second deliverable in a cycle of three which will all focus on these evolving, more stringent, more specific and more demanding requirements.

This document follows D3.1, which contained the initial requirements. These initial requirements in D3.1 were based on generic use cases whose interest was the combined use -and the federation- of several facilities as the mechanism to enable "research by experimentation" over these complex scenarios. These scenarios were inspired by real ongoing and realistic needs coming from different sources, from current market and/or research trends to FIRE projects that run on these facilities. The deliverable D3.1 however did not take any input directly from experimenters or user communities from outside the Fed4FIRE consortium as these were not available at that time.

For the second development cycle, the set of requirements as defined in cycle 1 were used as the starting point, but were now updated and extended based on inputs from other sources including direct contact with specific research communities and also the set of proposals submitted to Fed4FIRE for experimentation in the 1st Open Call. These proposals represent real examples of the kind of experiments that will run on the federated facilities, and are considered to be a very valuable source of information regarding the needs of the FIRE community. This is clearly the first source of information originating from one group of stakeholders for Fed4FIRE.

A total of 55 experimentation proposals have been received in the context of the first open call, 37 of which involve WP3 testbeds. All the material provided by these experimenters has been analysed in order to confirm or update current requirements or to incorporate new ones. This analysis was a joint effort of representatives from all WP3 testbeds. As an illustration of the amount of information included in each of those analysed open call documents, two of those experimentation proposals submitted in the 1st Fed4Fire open call have been selected and described in this document for illustration of the process. In addition, both proposals are also clearly showing the advantage of the federation from the perspective of the experimenter

The WP3 testbeds also interacted with their specific research communities directly in order to identify missing requirements. However, during Cycle 1, no specific functionality gaps have been identified in the requirements and corresponding architecture. It seems therefore that the use cases as presented in D3.1 were already good representation of the kind of experiments that the WP3 communities typically have in mind

This first set of use cases, as defined in Deliverable D3.1, were valuable in the initial phase of the Fed4FIRE project, but did not push the federation to the limit. They clearly demonstrated the value of federating testbeds by combining different technologies that cannot be found in a single testbed; or by performing the same experiment in sequence on different testbeds in order rule out environment-related biases in the results. However, these envisaged experiments where still limiting themselves to the usage of a few testbeds and did not also push the testbeds to their imits (e.g. in number of nodes, capacity,..). The same trend could be identified in the analysed open call proposals, but as in input to the second cycle architecture, it was perceived important also to derive requirements from use cases that really push the concept of federation to its limits. For this purpose,

two new use cases that push these limits were specifically written for this deliverable, and included in the requirements analysis. Both new use cases originated from discussions within the Fed4FIRe consortium as well as with outside partners as e.g. GENI during the Fed4FIRE-GENI workshop held in Leuven.

As in the first cycle, and in view of consistency, the requirements have been classified according to the following functional areas:

- **Experiment lifecycle**: including discovery, reservation and experiment control
- **Measurement and Monitoring**: covering metrics, instrumentation, data management
- **Trustworthiness**: gathering federated identity management and access control, privacy, accountability, SLA management
- **Interconnection**: including access networks, routing, etc.

In this deliverable, these requirements have also been prioritised according to how many proposals and use cases benefit from a requirement. The resulting listing of requirements categorised according to their priority constitutes the output of this deliverable that will feed into the definition of the architecture of the second Fed4FIRE development cycle by WP2.

This deliverable is structured as follows:

- Chapter 2 gathers the description of the sources of the remaining, updated and new requirements. These sources include the use cases that formed the basis of the 1st cycle, interactions with specific research communities, the new defined use cases and the open call submissions.
- Chapter 3 describes the process used for deriving the requirements from these sources.
- Chapter 4 describes the requirements prioritization process as well as the complete and updated list of the requirements.
- Chapter 5 contains a summarized version of the most important requirements. This can be considered as the main output towards the architectural work of D2.4.

2 Sources used in the requirement elicitation process

This chapter describes the sources that were used in this deliverable to define the WP3 requirements for the second Fed4FIRE development cycle. These sources include the use cases that formed the basis of the 1st cycle, interactions with specific research communities, the new defined use cases and the open call submissions. Compared to the previous requirement elicitation described in D3.1 [1], we can say that these additional sources helped us to reach the target of broadening the scope of our requirement analysis in this deliverable and made use of all available information at this point in the project.

In the process described here, the translation of Open Call proposals to actual requirements was based on interpreting the proposal texts by the Fed4FIRE Partners as no specific section was used in the proposal template for proposers to identify requirements. The advantage of this approach is that we have a direct impact on the quality of the requirement elicitation ourselves. It was felt that if listing the requirements was left solely to the proposers, we would have had no indication about the care that they have taken to define their requirements, about the fact if they correctly focused on the federation needs instead of on the testbed-specific needs, etc. Also, there is always a risk that their requirements are biased because of their desire to win the Open Call.

In view of the labour intensive work this analysis represents and in order to guarantee to be able to generate a complete list of requirements, a specific section on this will be implemented in the template for the next Open Calls. This input will be considered as additional information, and will not replace the analysis of open call proposals by Fed4FIRE consortium members.

Another additional input, planned for the third and last requirement elicitation in D3.4, will be the recommendations coming from the finished experiments of the first open call. A specific milestone for this feedback (M10.4) has always been part of the Fed4FIRE work plan, and is planned to be available before the deadline of D3.4. These recommendations based on the extensive hands-on experience gained by the open call experimenters will be a very important new element of that future requirements analysis, just as the analysis of our first open call proposals was in this deliverable.

2.1 Pending requirements from cycle 1

At the start of Cycle 1 a full list of requirements and a reduced requirements matrix of high priority requirements were defined and served as input towards the first iteration of the architecture. Some of these requirements have been (partially) fulfilled in this first cycle of Fed4FIRE, but others remain.

For this deliverable serving as the input for the 2nd development cycle the selected starting point was the Cycle 1 full list of requirements, and not the prioritized matrix. This approach was chosen because we opted for a more transparent prioritization scheme incorporating the input regarding requirements originating from the open calls and the new set of use cases defined in this deliverable. For this full list, no analysis was however available to annotate to which degree requirements have been met or not in the first cycle. Such an analysis is only available for the reduced matrix in D2.1. Therefore the analysis of this full set of cycle 1 WP3 requirements has been one of the first steps towards the second set of requirements defined in this deliverable.

2.2 Interactions with specific research communities

The WP3 testbeds also interacted with their specific research communities directly in order to identify missing requirements. However, no specific functionality gaps could be identified in the cycle 1 requirements and corresponding architecture. Only iMinds reported that IPv6 support is crucial since some of the testbeds (e.g. Virtual Wall and w-iLab.t) need to support IPv6 due to a shortage of public IPv4 addresses. If resources of the other testbeds need to interact with these IPv6-based testbeds, it is a must that these other testbeds also support IPv6, or no communication between the resources will be possible. This finding is an important input for the prioritization of the IPv6 requirement that was already defined in cycle 1.

2.3 Open Calls

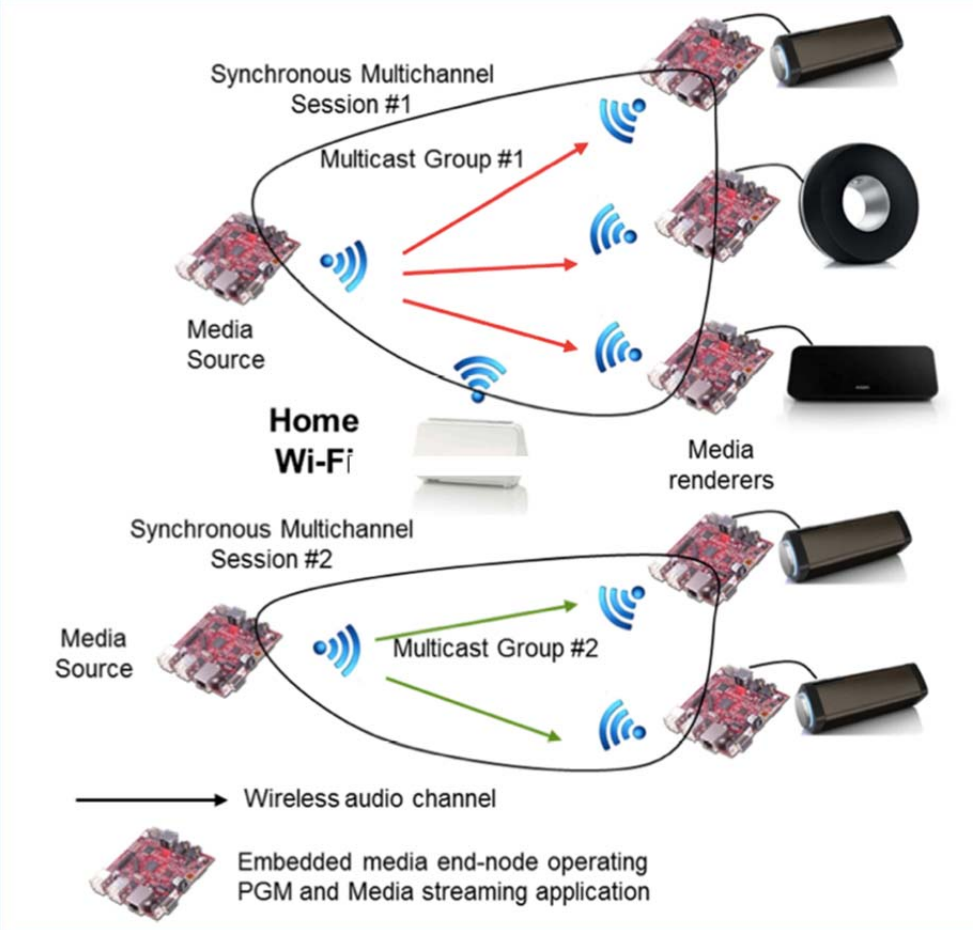
The Fed4FIRE open calls process is a very important source of requirements. The proposals submitted to Fed4FIRE for experimentation are real examples of the kind of experiments that will run on the federated facilities and represent a first hand input as community requirements from the Fed4FIRE Stakeholders. All the material provided by these experimenters has been thoroughly analyzed in order to confirm or update current requirements or to incorporate new ones.

All proposals involving WP3 testbeds have been reviewed following a common procedure that consisted of an analysis from the point of view of a support service helping/guiding the experimenters to deploy their experiments across the WP3 testbeds. Both the functionalities needed by the experiments and what Fed4FIRE can provide at federation level were taken into account during the requirements identification process. More details about this process are given in section 3.

The following two subsections contain an extract of two of the mentioned proposals for which permission was granted to have them included in this document. They are an illustration of the amount of information included in each of those 37 analysed open call submissions. These two proposals can also be seen as use cases that represent the value of a federation of testbeds as defined in Fed4FIRE since the users will have the possibility of performing the experiments on multiple testbeds and taking advantage of the varied environment and different technologies that characterise each testbed.

2.3.1 MEDIANET: Experiments on synchronous wireless media streaming entertainment applications in residential networks, exploiting Pragmatic General Multicasting (PGM)

Title	**Experiments on synchronous wireless media streaming entertainment applications in residential networks, exploiting Pragmatic General Multicasting (PGM)**
Background / Rationale	Wireless synchronous media steaming applications represent a huge emerging market segment for connected entertainment CE products. Means of streaming synchronously and wirelessly audiovisual content residing within the home network to connected speakers and TVs enable dynamic creation of immersive entertainment environments that seems to bear important consumer value due to the convenience of the wireless connectivity (double TV applications, home cinema experiences, multi-stereo experiences). Existing synchronous media streaming applications implemented using IP unicasting overload the network, lack scalability, while communication complexity leads to complex and thus expensive for the consumer designs. MEDiANET addresses experimentation on a new technology for in-home network media streaming based on IP multicasting that exploits the Pragmatic General Multicast protocol. As simple IP multicasting is implemented over UDP, loss of packet in media streaming applications cannot be corrected leading to perceivable quality distortion of media streams. PGM supports packet retransmission for error detection in a very simplistic way to avoid increasing bandwidth utilization. As PGM is an experimental protocol not exploited up-to-date in real life multicasting applications, MEDiANET will create a number of network configurations for synchronous wireless media streaming applications to test IP multicasting performance over PGM under various application configurations and congestive home network conditions. The final goal of the experiments is to assess PGM level of maturity for adoption on commercial connected entertainment audio solutions.

Title	**Experiments on synchronous wireless media streaming entertainment applications in residential networks, exploiting Pragmatic General Multicasting (PGM)**
Picture	
Scenario description (Storyboard)	The present proposal focuses on a particular area of the wireless entertainment applications, that of the synchronous media streaming. The latter is a non-technical term that refers to the ability of streaming content, e.g. music, video, from one source (smartphone or PC) to many media renders, e.g. TVs and home audio speakers, using the local WiFi network. The concept of the wireless synchronous media streaming enables new entertainment experiences for the domestic users for music and movies rendering on home audio systems (speakers, soundbars, woofers, portable speakers, FM radio channels, ...) and TVs. For example, consumers can create on demand immersive sound environments (surround effects, multi-stereo with bass reflects, etc) by grouping wirelessly several speakers in a single or multiple rooms. The main technical challenge of the real time wireless media streaming in home networks is to compensate the errors created on the air interface due collisions of background traffic generated by other connected appliances and the residents, or by impairments of the radio link due to the co-location of several WiFi Access Points (AP).

Title	**Experiments on synchronous wireless media streaming entertainment applications in residential networks, exploiting Pragmatic General Multicasting (PGM)**

Current technical solutions exploit TCP and IP unicasting between the source and each renderer to overcome the problem of packet loss. Although such a solution performs well in 1-to-2 configurations, it does not scale as the source has to generate as many IP unicast sessions towards the renders as many renders are included in the synchronous group, thus leading to wireless bandwidth saturation and overutilization of computational power of the source, which in most cases is the smartphone. Moreover, software engineering of such a solution becomes very complicated, requiring dedicated mechanisms for maintaining synchronization on the buffers of the renderers to ensure content rendering with perceived by the consumer delay jitter.

The main objectives of MEDiANET are to:

- Integrate the PGM protocol on embedded Linux devices so as to create the so referenced later media end-nodes able to function as media renderers (in particular audio speakers).
- Integrate PGM functionality on the media streaming applications of the embedded media end-nodes for content streaming and rendering.
- Deploy the media end-nodes in several wireless home network simulated configurations on the Fed4FIRE experimental facilities to form reference streaming applications for testing.
- Specify and conduct experiments for the performance evaluation of the PGM protocol and the implementation of synchronous wireless media streaming using PGM.
- Compare the performance of reliable multicasting with that of simple unicasting being used today for synchronous media streaming for audio applications.

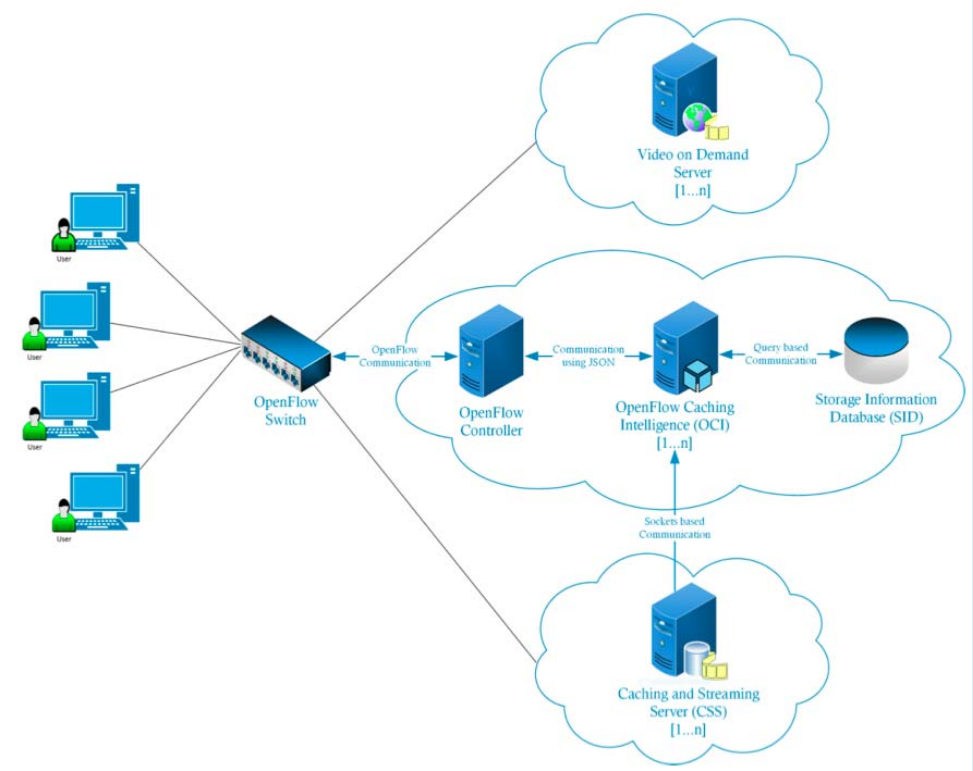
The experiment will exploit the resources of the w-iLab.t experimental facility, owned by iMinds in Ghent. W-iLab.t offers a dense wireless (WiFi) network environment with controllable resources, traffic profiles and radio conditions, which are mandatory requirements for the experiments. In addition w-iLab.t is made up of different vendor WiFi nodes, an important offering that will allow testing PGM protocol interoperability on several home network implementations.
Since the targeted use-case requires that both the media content and the media end-nodes are physically located in the same network, experimentation will basically require resources from one wireless testbed facility. However, inclusion of a second facility of wireless access points will help to identify any potential networking issues and the general behavior of the protocol over wide area networks in the context of services related to remote joint experience sharing. As such, during the last month of the experiment it is planned to interconnect with w-iLab.t one more of the available Fed4FIRE experimental facilities; either NITOS at Volos and NETMODE at Athens. Final selection of one of the two shall be based on the basis of available connectivity means and communication control versatility (e.g. means for bandwidth control) on the interconnection path with w-iLab.t.

Title	**Experiments on synchronous wireless media streaming entertainment applications in residential networks, exploiting Pragmatic General Multicasting (PGM)**
	Potential networking issues for analysis under the federated experiment setup include PGM performance when traversing multiple routers in the network and PGM collaboration with the NAT protocol.
Services and facilities involved	Based on the provider facility: • W-iLab.t o testing PGM protocol interoperability • NETMODE o interconnection path with w-iLab.t o PGM performance when traversing multiple routers o PGM collaboration with the NAT protocol • NITOS o interconnection path with w-iLab.t o PGM performance when traversing multiple routers o PGM collaboration with the NAT protocol
Technology involved	Physical PC/server, Virtual Machines, Wi-Fi access

2.3.2 MEVDDS: Multi-testbed Experimentation of a Video-on-Demand Distribution Service

Title	**Multi-testbed Experimentation of a Video-on-Demand Distribution Service**
Background / Rationale	The use and popularity of Video-on-Demand (VoD) services are increasing, as is their impact on distribution networks. When huge content is delivered through independent unicast flows, naively ignoring that much of it is identical to transmissions hours or days earlier, network provision is greatly challenged. Mechanisms are therefore sought to reduce the impact of repeated delivery of identical content over the network, and to reduce the cost of delivery to the network operator. Peer-to-Peer (P2P) networking has been demonstrated as a highly competent means to deliver a live video stream to multiple users simultaneously. However, P2P is much less effective for VoD, as the time between requests for identical content may be in the order of minutes, hours or days. Content Delivery Networks (CDNs) are often used to reduce the impact of repeated VoD content traversing the network. However, despite the fact that CDNs install nodes in data centres worldwide to minimise source server load, it is unrealistic to expect CDNs to deploy in all ISP networks and, in particular, benefit last-mile delivery environments.

Title	Multi-testbed Experimentation of a Video-on-Demand Distribution Service
	OpenFlow promises the ability to rapidly deploy new protocols and services over a network of fast flow-based switches that are highly configurable. We have designed and built a prototype architecture that offers in-network intelligent and transparent VoD caching with the use of OpenFlow at a single physical site. We propose to extend and evaluate our OpenFlow-assisted VoD distribution architecture running across multiple geographically distributed testbeds. This will be realised with the use of the Fed4FIRE facility. The main aim of this innovative multi-site experimentation is to thoroughly evaluate the VoD in-network caching service from both a network and user's point of view. The goal is to reduce the overall network utilisation of the network (being Fed4FIRE federated testbeds in this instance), whilst simultaneously improving the Quality- of-Experience (QoE) perceived by the end-user. Our work will focus on using OpenFlow to optimise important VoD delivery metrics, such as buffering time, throughput, and video quality. The Fed4FIRE multi-site facility provides a realistic setting for experimentation on a distributed infrastructure with different types of testbeds over the Internet. Such innovative experimentation with a highly demanding traffic profile (i.e. video) will go beyond what is possible with a laboratory evaluation. Most importantly, Fed4FIRE does not only provide a federated network, but it also provides powerful experimentation tools, including tools for discovery and reservation of resources, controlling of experiments and collecting experimental measurements. These Fed4FIRE tools will offer unique and significant value to the proposed experimentation and evaluation of our architecture. In addition, the proposed resource intensive experiments will stress-test the inter-communication of the federated multi-site testbeds and the performance of the resource provisioning and monitoring tools across these testbeds and will provide an important opportunity for thorough feedback. We intend to provide feedback on the usefulness and maturity of the Fed4FIRE experimentation tools and facility in general, and suggest improvements back to the Fed4FIRE consortium.

Title	**Multi-testbed Experimentation of a Video-on-Demand Distribution Service**
Picture	 **An OpenFlow-assisted VoD Distribution Architecture**
Scenario description (Storyboard)	Recent years have also seen Video-on-Demand (VoD) surpass live streaming as the dominant mode of consumption. With increasing popularity, online video websites are becoming the main media hub for people to retrieve music, catch-up on TV shows and enjoy films. The increasing growth in Internet video and the popularity of HD on-demand services is leading to an alarming challenge to the underlying network infrastructure concerning its distribution efficiency. The distribution of live content involves the streaming of data to all target users simultaneously, so it is able to exploit multicasting mechanisms that provide such simultaneous delivery. In contrast, VoD requests must be handled individually, leading to an independent media flow in the distribution network for each user request. Using such a unicast content delivery paradigm, an enormous amount of identical media objects (in the order of gigabytes for each HD film) is likely to be delivered on the same network segments repeatedly. This puts an additional burden on the network, as the end-to-end capacity of a network infrastructure must continuously increase to match the growing number of Internet video users and the growing popularity of HD VoD services. Storage and caching servers have traditionally been used to store web content (e.g. HTML pages, images, web documents etc.) in an effort to improve the efficiency of content delivery in a network. By serving content from local caches, network administrators manage to reduce the overall bandwidth utilisation, decrease the user perceived latency and minimise server load. More recently, attention has moved to the caching of video- based content. With the use of OpenFlow, we can provide an efficient transparent in-network

Title	Multi-testbed Experimentation of a Video-on-Demand Distribution Service
	caching service of large media objects, such as video files for a VoD service. OpenFlow allows us to specify and develop the appropriate caching and storage extensions to optimise the repeated delivery of this identical content (even across different sites). With OpenFlow this can be achieved in a user transparent fashion, whilst allowing the applications deployed across the network to continue using apparently simple unicast flows. Such an approach maintains the underlying video delivery mechanism and does not require any fundamental changes in service, therefore potentially achieving much higher efficiency and manageability of the network. In addition, an OpenFlow-assisted VoD service based on transparent in-network caching will potentially reduce the overall bandwidth usage, improve the user perceived latency and increase Quality-of-Experience (QoE) for its users. For Fed4FIRE we propose to extensively evaluate our OpenFlow-assisted VoD distribution architecture for cross-site deployment, using the multi-site testbeds of Fed4FIRE. In particular, we propose to: 1. Perform the appropriate configuration and extend the OpenFlow-assisted VoD distribution architecture to facilitate efficient caching and distribution of VoD flows that have to traverse 8 multiple geographically distributed sites of differing capabilities. 2. Use the experimentation tools of the Fed4FIRE facility to evaluate end-to-end cross-site VoD traffic over different types of testbeds, from both a network and a user's point of view. The aforementioned innovative experiments are to be carried out over multiple testbeds on the Fed4FIRE facility. Currently the scope of a laboratory environment is very limited and restrictive for evaluation of an architecture that is to be eventually deployed over the Internet. The Fed4FIRE multi-site facility provides the ideal realistic setting for experimentation on a distributed infrastructure with different types of testbeds over the Internet.
Services and facilities involved	Testbeds that have OpenFlow capabilities: • w-iLab.t • i2CAT island • Koren • NITOS • Norbit • PlanetLab
Technology involved	Physical PC/server, Virtual Machines, OpenFlow

2.4 Use cases

The use cases as defined in the deliverable D3.1 and which formed the basis for a set of requirements feeding into the first development cycle, are briefly summarized below as they are still included in the information mentioned in the column "source scenarios" of the requirements tables of section 3.

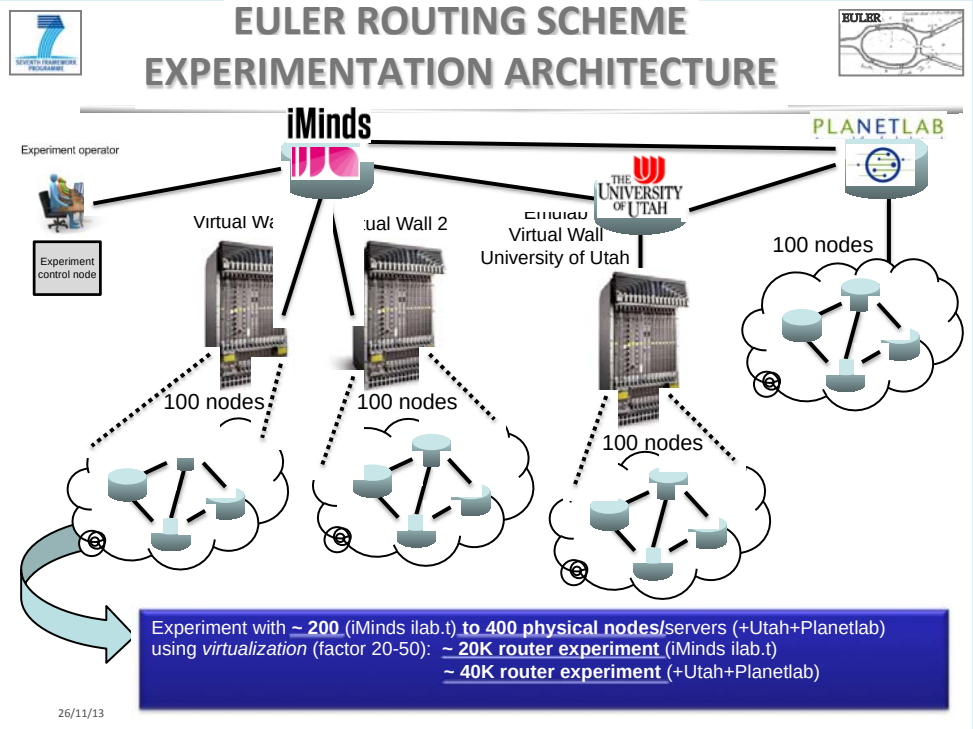
- *Teaching computer science using FIRE facilities*
  The scope of this use case is to introduce students to the basics of IP based networking through lab exercises that use the FIRE facilities.
- *Testing a networking solution for wireless building automation on different platforms*
  Building automation systems lowers the total cost of ownership, increases the security level and raises the comfort of the people inside the building. In this scenario an SME wants to use the federated FIRE facilities to perform optimisation tests on wireless mesh network (WMN), sensor and actuator networks (SANETs) and test the compatibility between the two.
- *Researching the concept of geographical elasticity in cloud computing*
  In cloud computing, horizontal and vertical elasticity are two techniques to handle increasing user demands. In this use case, a researcher has developed the concept of geographical elasticity where users of a cloud service can be clustered in two distinct locations and the corresponding VM could be split into two instances which are deployed close to these two locations.
- *Benchmarking a service platform for information retrieval*
  The scenario described here involves information retrieval from a document archive where documents should have to be analyzed by an algorithm that e.g., automatically identifies all named entities present in the document, and automatically defines appropriate keywords and news categories.
- *Mobile cloud service platform*
  In this use case the developer decides to build his solution on top of several facilities, he then needs to discover what infrastructures are available, that means which wireless technologies are available, which cloud infrastructures could be used, whether mobility can somehow be introduced, if part of the experiment can happen over licensed cellular technologies, which devices in the federation are more resource constrained and can be used to model the end-user devices, and so on.

However, as mentioned before, these cycle 1 use cases did not yet push the boundaries or federation to the limit. They clearly demonstrated the value of federating testbeds but they restricted themselves to the boundaries as set by the testbeds themselves..

As input to the second cycle architecture, it was perceived important to also derive requirements from use cases that really push the concept of federation to its limits. For this purpose, two new use cases are proposed for this deliverable, and included in the requirements analysis. They are inspired by ideas for real experiments that have been expressed in the FIRE community, but were enlarged to a higher scale, or extended with additional technologies to be tested. They are introduced in the remainder of this section.

2.4.1 Design and evaluation of alternative inter-domain routing schemes

Title	Design and evaluation of alternative inter-domain routing schemes
Background / Rationale	This experiment was presented at the GENI-FIRE collaboration workshop that was hosted by Fed4FIRE on October 14th-15th 2013 in Leuven, Belgium. It originates from the EULER project. The problem statement behind this experiment is the fact that BGP (Border Gateway Protocol) cannot cope adequately with the current Internet growth rate. Every year, Internet traffic increases roughly with 50%. The routing tables grow at a rate of 15-25% per year, and the number of Autonomous Systems (AS) grows with 10% per year. As a result, BGP prefix update and withdrawal rates have also increased significantly. This is alarming many network operators, equipment vendors and researchers since these increased rates of BGP updates have the potential to destabilize the Internet routing infrastructure and increase both the design complexities and cost of next-generation routers. One example of these effects is the observation that BGP suffers from churn which increases load on routers (due to topological failures (link/nodes) and traffic engineering (prefix de-aggregation). BGP's path vector amplifies these problems (path exploration) To overcome these problems, the EULER project is researching if other routing schemes would be able to handle these challenges better. Many different routing schemes exist, they can be divided in several classes such as geometric/greedy routing (in Euclidean and hyperbolic space), path vector routing (extended BGP++, alternate path vector routing, …) and hybrid schemes. Finding the best candidate is not trivial, every routing protocol is characterized by a specific trade-off between memory space (RT size), stretch (path length) and adaptation cost (messaging and processing). Stringent requirements are imposed to any chosen candidate, both in terms of functional requirements (distributed protocol, dynamic and/or fault tolerant, support for unicast and multicast) as performance requirements (memory, routing performance/stretch, convergence time and communication costs). These possible solutions have to be tested in several ways, characterized by an increased level of realism but also by a higher cost: mathematical proof, simulation, emulation and testing on real systems. Fed4FIRE facilities come into play for the steps of simulation and emulation. In this specific use case we will focus on the emulation aspect. In that case, experiments are run on the real OS, using real applications, in a lab environment, and under synthetic conditions. Such scalability testing of alternative inter-domain routing schemes requires a large amount of nodes before the results have any scientific meaning in the context of this problem statement. No single testbed could provide them, this requires the combination of multiple suitable testbeds in a single experiment. In this example use case, we have adopted the iMinds Virtual Wall testbed as the most appropriate one, and assumed that there would by several similar Emulab testbeds available through the Fed4FIRE federation. This is a clear example of how scaling up existing testbeds with resources from similar testbeds in the federation paves the way for experiments that could not yet be supported by the FIRE testbeds before they were federated.

Title	**Design and evaluation of alternative inter-domain routing schemes**
Picture	
Scenario description (Storyboard)	The experiment will test to which degree 3 specifically chosen and developed inter-domain routing schemes could lead to better performance in very large scales compared to a reference BGP implementation. Three different parties of the EULER project have implemented these algorithms. The intention of these parties is to measure performance metrics of their solutions both related to routing performance (path length, routing table size, communication cost, convergence time, configuration time) and to forwarding performance (table size, delay, throughput, jitter and packet loss). The intention is to test this performance under different conditions, including different AS-level network topologies and network dynamics (topology failures, AS-Node mobility and new AS-node attachment). The scale of these experiments should be scaled from 100 AS until more than 10.000 AS. To start this experiment, researchers from the 3 groups involved in the experiment will need to have a Fed4FIRE account. In this example we assume that one of them is iMinds, which operates the Virtual Wall testbed and which has its own Fed4FIRE-compatible identity provider. So the iMinds researchers that already have an account on their own testbed can use this same account to work on the experiment using any Fed4FIRE resource. The other 2 groups involved do not operate such an identity provider, and therefore have to request a new Fed4FIRE account. For this, they want to just go to the Fed4FIRE portal to register and receive the appropriate credentials. Once all involved researchers have received their accounts, they want to team up in a single project/experiment, since they will perform all sub-experiments in a collaborative way, reusing each other's scripts, reference implementation of BGP,

Title	Design and evaluation of alternative inter-domain routing schemes

etc. They also want to be able to easily see each others results, so that it becomes easy for them to create charts that compare the performance of the different tested solutions.

With all this in place, they can start designing their experiment. Their requirements in terms of desired resources are quite simple: they want to have wired resources in a topology that they can control, and they want to have a lot of them! It should be easy to find such resources without needing days of work exploring all the documentation material that could be available. One possibility is that they should be able to browse through the list of testbeds in the federation in a very quick way, but also in such a way that they immediately get a grasp of what each testbed is intended for, but could also find more detailed info when they want it. You could make the analogy to browsing through a list of abstracts on a paper search tool such as IEEE Xplore or the Web of Science, where a researcher typically looks at several abstracts quickly, and only selects those that seem most interesting for further reading. Another possibility for finding the appropriate resources could be a user interface that not starts from the testbed viewpoint, but from the resource. So an experimenter would want to find an appropriate resource by defining some of its desired characteristics, getting back a list of suitable candidates without really caring about the underlying testbed. Note that in this example no single testbed would be able to provide the desired large amount of resources. However, as depicted above, the combination of several Emulab testbeds (Virtual Wall 1 and 2 from iMinds, the Emulab instance from the University of Utah - assuming that they would have joined the Fed4FIRE federation) and the PlanetLab Europe testbed would make it possible to implement the desired large-scale scenarios.

After identifying all suitable resources, the experimenters will want to design their actual experimentation topology. For first exploratory tests and development of their specific scripts, they want to be able to manually setup a small topology. As the scale increases, they also want to be able to create the desired topologies automatically, based on an appropriate description of its characteristics. The researchers also want to be able to share the used topologies with each other, and to reuse them later on themselves.

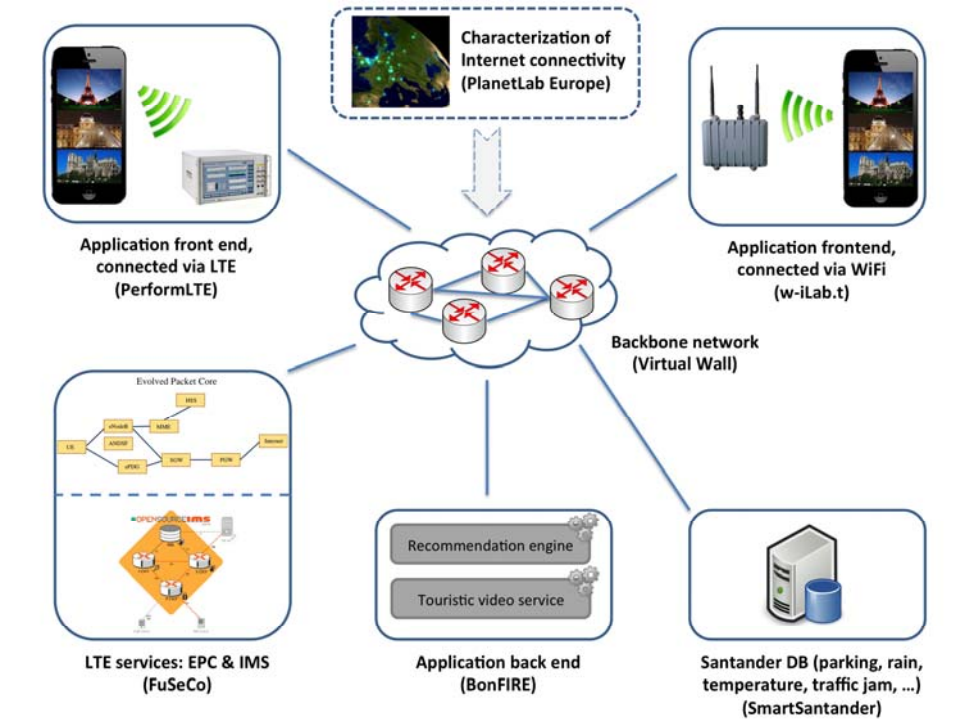
Once the topologies have been designed, and the corresponding resources have been instantiated for the experiment, the researchers want to install their software on the nodes. This means that they have to be able to SSH into the nodes, and to have root rights since they want to install specific inter-domain routing schemes in the kernel for performance reasons. To scale up the experiment, they also want to be able to use their physical nodes to create multiple virtual machines using their own specific optimized virtualisation containers. This requires root access but also guaranteed full ownership of the physical resource during the experiment.

Having all the needed software in place, the next step is to make the resources behave according to the desired experiment scenario. First the researchers will want to test if this is actually possible. For this they will start with controlling the resources' behaviour manually. In this case this means that they will login to the

Title	Design and evaluation of alternative inter-domain routing schemes
	nodes of a small-scale topology through SSH, and perform the appropriate control commands on the command line themselves. Examples are changing the running BGP algorithm and its parameters over time. However, when scaling up to thousands of nodes, this manual control becomes unfeasible. Therefore the experimenters want to have access to a mechanism that allows them to define the behaviour of all resources during the course of the experiment in a convenient manner. This mechanism should also allow experimenters to execute several identical runs of the same experiment to strengthen the observed results. To gather these results, some supporting tools will again be desired to handle the practical side of the very large-scale experiment runs. Similar to experiment control, the experimenters will want to explore suitable metrics through manual collection of them during small-scale experiments. But as soon as they start to scale up, it should be convenient to define where all these thousands of nodes should store specific results such as memory consumption, routing performance and convergence time. Afterwards, this data should be easily retrievable so that the experimenters can process this data from their own scripts, e.g. in MatLab. If this processing would be to CPU or memory intense for the experimenters' own resources, then it should be possible to add additional resources to the experiment on which this data processing can be performed in a distributed manner. The data should be shared between all experimenters involved in the project, and it should be possible to make sure that the data is stored for a longer period of time, since they might need the results quite some time after the experiment, e.g. when they have to revise a submitted paper based on these results, or when they want to use the results as input for other studies.
Services and facilities involved	Virtual Wall and other Emulab-based testbeds for the BGP experiment, same testbeds and other testbeds that can provide rather powerful generic purpose machines (e.g. BonFIRE) for the data processing aspect.
Technology involved	Physical PC/server, Virtual Machines, Emulab network emulation

2.4.2 Touristic video on demand services in smart cities

Title	Touristic video on demand services in smart cities
Background / Rationale	The idea is that in a city many tourists arrive in the morning at the same places (e.g. airport, train stations, central square), follow the same route along the major landmarks, and leave again in the evening at the same place at roughly the same time. As a result, tourists often have to queue for a rather long time to visit a certain site, while it was actually quite calm at this venue during other times of the day. To solve this problem, a certain SME is planning to develop a touristic application for smartphones. The intention is that tourists will use this application to decide which place to go to next when exploring a city. This application will calculate the top 5 of most suitable next stops based on the user's profile, current user location and details of the current environmental conditions in the city based on smart city information (e.g. parking sensors, temperature sensors, rain sensors, road sensors to find traffic jams, participatory sensing information, etc). To assist the tourist in making a good choice, the application will allow the user to open a video regarding each entry in the top 5. As for any SME, the development of such an application would require attracting new financial injections of interested investors. To convince these possible investors, it is needed that the SME can clearly show that this concept is actually feasible. For this reason, the SME wants to test an early prototype on FIRE facilities, investigating all aspects of the envisaged system. Therefore it is needed to combine data coming from a smart city testbed with a server that does al the tracking and calculations and with smartphones running the actual application. One of the biggest uncertainties is the question if the LTE network in the city would be able to support such a system. And if not, it is needed to know if the city-wide WiFi network that the city is planning to open in a few years time could support it instead. In this specific experiment quite a number of heterogeneous resources from different testbeds have to be combined. To be more specific, this experiment will rely on SmartSantander to provide the smart city input data, on w-iLab.t to emulate a city-wide WiFi network and some smartphones, on PerformLTE for the LTE radio access (using an eNodeB emulator) and some smartphones, on FuSeCo to provide all services of a an LTE Evolved Packet Core (EPC) and IMS system to setup media streams, on the Virtual Wall to emulate a realistic backbone network between the other components of the system, and on BonFIRE for the application back-end that calculates the most appropriate advised next visits, and provides the corresponding VoD feeds. Note that to make sure that the emulated backbone is realistically configured (e.g. emulating a setup with the back end running in the data centre of the SME in Brussels and serving users in Santander), PlanetLab Europe will first be used in a separate single-testbed experiment to characterize the specific connectivity between those locations on the real public Internet. It would not be possible to realize such a technologically diverse experiment without a FIRE federation as Fed4FIRE, especially not when considering that just as any other SME, this specific SME is continuously struggling to survive, meaning that they have very stringent constraints in terms of timing and available manpower.

Title	Touristic video on demand services in smart cities
Picture	
Scenario description (Storyboard)	In this experiment only a small team of engineers, all belonging to the same SME, will participate. Since this team only has limited time available for this specific experiment, they want to get going as soon as possible. Therefore it should be very easy to retrieve information about which kind of technologies are actually provided by the federated infrastructures. In a very short time they should be able to assess if Fed4FIRE could indeed deliver smart city sensor readings, smartphones, an LTE network (both radio access and packet core), a WiFi network, cloud computing capabilities, etc. They should be able to find this information even without first asking for an account on Fed4FIRE. Only once they are pretty confident that Fed4FIRE would enable them to implement the intended experiment, then they would want to request a user account. This again should be very simple, ideally a matter of filling in some personal details on a registration website, not more than that. They should also be able to start playing around with these resources right after requesting their account. If they have to wait a week before their request gets validated and they can start using the resources, this is a non-acceptable loss of valuable time. To use these resources efficiently, it is important that the way of doing this is similar across the different testbeds. If they have to study specific experimenter tools for every involved testbed, it becomes very cumbersome for them to implement the experiment. The specific tool has to be very user friendly, and easy to install. A website from which the experiments can be run would be very convenient, as long as such a website is compatible with the browser used by the experimenters. Alternatively, a stand-alone application would also be appropriate, as long as it is easy to install, and supports the different platforms used in the team (Windows, Mac and Linux).

Title	Touristic video on demand services in smart cities

Whatever tool being provided, the experimenters will first of all want to be able to select specific resources to be used in their experiment. For this it is important that they can easily select resources based on the technology that they support, or based on the testbed to which they belong. When selecting resources, the experimenters also want to be informed about how these specific geographically spread resources can be interconnected (layer 2/3, bandwidth and jitter characteristics, etc.). Once the experimenters have designed the entire topology of their experiment, including all different types of needed resources and interconnectivity configurations, it should be possible to easily start the entire experiment. When all these different resources are being provisioned for them, they should be able to grasp what is exactly going on.

Once the provisioning of the resources has been completed, the experimenters will first want to try what they can do with the resources manually. Typically this will mean login in to the different nodes via SSH. To do so, it should be very easy to derive the appropriate connection details for every resource involved in the experiment. It should be possible to login to these resources using the single set of Fed4FIRE credentials that they got when registering as a Fed4FIRE experimenter in the first step. Piece by piece the experimenters will want to setup components of their overall system, by putting the needed additional software on the resources, configuring these applications correctly and running them. The final goal is to have a full system running, in such a way that the final user experience and corresponding QoE can be shown live.

Since this experiment is not really about deep scientific validation of some concept, but merely a proof-of-concept demonstration for possible investors, there is no need to implement really advanced scenarios. As long as the entire system works and they can display the output of the application to the possible investors, then this is sufficient. What would be required though is a mechanism that allows the automatic bootstrapping of all services under test when starting an experiment. The rationale behind this is the following: it is expected that the duration of this experiment will be limited by specific constraints. Being able to let this experiment run without any discontinuation for a few months seems very unlikely. But it should be possible for experimenters to easily restart the final and fully working experiment later on, e.g. when meeting with possible investors for a live demo of the proof-of-concept. In that case all provisioned resources should not only be running the default images provided by the testbeds, but should also automatically start all the experiment-specific software. As a result, it should at least be possible to save an experiment topology at any given moment in time, and have this saved configuration provisioned automatically later on. Next to that it should also be possible to handover some boot scripts to the nodes automatically, or to save the image of the hard drive of the resources at any given moment, and have it being flashed back to the nodes when they are provisioned again for this experiment later on.

On a similar level, there is also not a real need for advanced collection of measurement data. The intention is not the create graphs for scientific publications. When developing the PoC, it is interesting though to have some

Title	Touristic video on demand services in smart cities
	background information available about the testbed itself (is it up and running? Are some specific components down? …) and the used resources (how much CPU is being used? How much RAM is available? ….) during development, to easy identification of causes of encountered problems. One thing that is important for this team though is the aspect of privacy. They are testing a very novel idea, and therefore would need to be sure that no information whatsoever about this is shared with other parties. This includes experiment design, motivation for request a Fed4FIRE account, experimental results, etc.
Services and facilities involved	SmartSantander to provide the smart city input data, w-iLab.t to emulate a city-wide WiFi network and some smartphones, PerformLTE for the LTE radio access (using the LTE eNodeB emulator) and some smartphones, FuSeCo to provide all services of a an LTE Evolved Packet Core (EPC) and IMS system to setup media streams, the Virtual Wall to emulate a realistic backbone network between the LTE radio access part and the EPC components and BonFIRE for the application back-end that calculates the most appropriate advised next visits, and provides the corresponding VoD feeds.
Technology involved	IoT, smart cities, LTE, EPC, IMS, WiFi, network emulation, cloud computing.

3 Description of the requirement elicitation process

In the following section a summary is presented of the process that has been used to analyze the sources mentioned in section 2:

- Regarding the pending requirements from cycle 1, these were copied from D3.1 [1], and their coverage in the cycle 1 architecture was analysed by the WP3 lead.
- The interaction with specific research communities was the responsibility of every WP3 testbed, each testbed had the freedom to organize this as it saw fit. It can be stated though that in general this interaction was a combination of talking in person to colleagues not involved in Fed4FIRE but which quite often represent a significant fraction of the testbed's local user base and corresponding research community; and email or telephone dialogue with other members of this community.
- The analysis of the new use cases was performed in an identical manner as for the ones listed in the previous requirements deliverable D3.1, and needs no further explanation.
- The analysis of the received open call proposals is a new element in the requirement elicitation process, and deserves some further explanation which is given below.

When analysing the open call proposals, we have focused on the requirements regarding the federation rather than requirements regarding individual testbeds. Some cases might indeed assume a certain testbed is offering functionality beyond their current capabilities as e.g. some wireless technology currently not offered on a testbed, but in that case, the individual facility might retain this as a requirement for its own internal roadmap as a testbed, but this has not been considered a requirement for the federation. In the previous case of a wireless technology, the requirement for the federation would be to expose the actual technology the testbed is offering allowing the experimenter to run his experiment on these nodes.

The process consisted of mapping each analyzed proposal to as many requirements as possible out of the list identified in D3.1, modifying the requirements when needed and adding new ones when no matching was possible. Also, for any requirement influenced by this new analysis we annotated if it was a requirement from cycle 1 which needed changes, or if it was an entirely new requirement. The overview tables below list for each of both existing and newly defined requirements from which proposals from the Open Call this requirement arises. The grouping of the requirements is according to the information mentioned in previous sections. All of the Open Call proposals received were framed according to the scheme illustrate in section 2 for the MEDiANET and MEVDDS proposals. The requirements were derived from these templates by the Fed4FIRE partners. The results are presented in table-format for ease of interpretation. Note that if a requirement is annotated green in these tables, then it is an existing requirement which formulation has been changed because of the Open Call analysis, while if it is annotated in blue then it is a new requirement produced by the Open Call analysis.

It should be noted that this identification process for the requirements is based on interpreting by the Fed4FIRE Partners as no specific section was used in the proposal template for proposers to identify requirements. This process can indeed be further improved for the next Open Calls. But given the large amount of analysed proposals, and the care taken by all partners throughout the analysis to look beyond the literal statements when defining the corresponding requirements, the consortium is confident the tables below are a good reflection of the actual needs.

Table 1: Open Call requirements - Resource discovery

	Node capabilities	Accurate location information	Wireless spectrum as a resource	Site location information	Discovery through federation-wide APIs	Intra-infrastructure topology information	Inter-infrastructure topology information	Background info virtualized resources	Query search	Catalogue search
AnyNetQoS@Fed4FIRE	1	1	1		1	1	1		1	
ARISTIDE	1			1	1		1	1		1
BILS	1	1			1	1	1		1	1
CEP4FIRE	1	1		1	1	1			1	1
CloudCONFetti	1				1	1	1			
DANCERS				1						
EARNEST	1	1	1	1	1	1	1		1	
ENA4FI	1	1		1	1	1	1			1
E-NETDAM	1				1	1	1			
EVIDENCE	1			1	1	1	1		1	
FED4NAV	1	1	1		1	1	1		1	1
FIRE4FoF	1	1	1	1	1	1	1		1	1
GEO-Cloud	1				1	1	1			
HEHET	1	1	1		1	1			1	
InLoc	1	1	1		1	1			1	1
INSURED	1				1	1	1			
IPCS4Fire				1						
LAMBDA	1				1	1	1			
MEDiANET	1	1	1			1	1		1	1
MEVDDS	1				1	1	1			
MH-ScaE	1			1	1		1	1		1
Mix4FIRE	1				1	1	1			
MMTE				1						
MobiSDN				1	1	1	1	1		
PRIORITY	1				1	1	1			
READINESS				1		1		1		
REIGN	1	1	1			1			1	1
SETONFIRE						1				
SILVERWoLF	1				1	1	1			
SoMAC	1	1	1			1	1		1	1
SSC	1				1	1	1			
Swiich4FIRE	1				1	1	1			
TELHEX@Fed4FIRE	1				1	1	1			
TITAN								1		
VIRTUALANDS	1					1	1			
WHISPER	1	1		1	1	1		1		
NOSC	1				1	1	1			

Table 2: Open Call requirements – Resource requirements

	Manually extract requirements from discovery query results	Extract requirements from discovery query results in an orchestrated manner	Describing required virtualized topologies	Requiring addition resources which are not yet exposed by the testbed
AnyNetQoS@Fed4FIRE		1		1
ARISTIDE	1			
BILS	1			
CEP4FIRE		1		
CloudCONFetti	1			
DANCERS	1			
EARNEST				1
ENA4FI				
E-NETDAM	1			
EVIDENCE			1	
FED4NAV	1			
FIRE4FoF	1	1		
GEO-Cloud	1			
HEHET	1			
InLoc	1			
INSURED	1			
IPCS4Fire	1			
LAMBDA	1			
MEDiANET	1			1
MEVDDS				
MH-ScaE				
Mix4FIRE	1			
MMTE	1			
MobiSDN			1	
PRIORITY	1			
READINESS				
REIGN	1			
SETONFIRE				
SILVERWoLF	1			
SoMAC	1			
SSC	1			
Swiich4FIRE	1			
TELHEX@Fed4FIRE	1			
TITAN				
VIRTUALANDS			1	
WHISPER	1		1	
NOSC	1			

Table 3: Open Call requirements – Resource reservation

	Hard resource reservation	Fairness	Secure reserve-tion	Automated reservations handling	Reservation information	Experiment planning assistance
AnyNetQoS@ Fed4FIRE	1		1	1	1	
ARISTIDE	1	1	1	1		
BILS	1	1		1	1	1
CEP4FIRE	1	1		1	1	1
CloudCONFetti	1	1	1	1		
DANCERS						
EARNEST	1		1	1	1	
ENA4FI						
E-NETDAM	1	1	1	1		
EVIDENCE	1					
FED4NAV	1	1		1	1	1
FIRE4FoF	1	1	1	1		1
GEO-Cloud	1	1	1	1		
HEHET	1		1	1	1	
InLoc	1	1	1	1	1	1
INSURED	1	1	1	1		
IPCS4Fire			1	1		
LAMBDA	1	1	1	1		
MEDiANET	1	1	1	1	1	1
MEVDDS						
MH-ScaE						
Mix4FIRE	1	1	1	1		
MMTE	1	1				
MobiSDN	1		1	1	1	
PRIORITY	1	1	1	1		
READINESS	1	1	1			
REIGN	1	1	1	1	1	1
SETONFIRE						
SILVERWoLF	1	1	1	1		
SoMAC	1	1		1	1	1
SSC	1	1	1	1		
Swiich4FIRE	1	1	1	1		
TELHEX@Fed4 FIRE	1	1	1	1		
TITAN	1	1	1			
VIRTUALANDS						
WHISPER	1				1	
NOSC	1	1	1	1		

Table 4: Open Call requirements – Resource provisioning

	Provisioning API	Customizing Linux	Root access	Internet access to software package repositories	Hard disk imaging	Automated network stitching	Automated software installation at boot time	Provisioning error notification
AnyNetQoS@Fed4FIRE	1	1	1		1	1		
ARISTIDE	1			1	1		1	1
BILS	1	1	1	1	1		1	1
CEP4FIRE	1		1	1	1		1	1
CloudCONFetti	1	1	1	1	1	1	1	1
DANCERS	1						1	
EARNEST	1	1	1		1	1		
ENA4FI						1		
E-NETDAM	1	1	1	1	1	1	1	1
EVIDENCE	1	1	1	1	1	1		
FED4NAV		1	1	1	1		1	1
FIRE4FoF			1	1	1		1	1
GEO-Cloud	1	1	1	1	1	1	1	1
HEHET	1	1	1		1			
InLoc		1	1	1	1		1	1
INSURED	1	1	1	1	1	1	1	1
IPCS4Fire	1						1	
LAMBDA	1	1	1	1	1	1	1	1
MEDiANET	1	1	1	1	1		1	1
MEVDDS		1			1			
MH-ScaE	1			1	1		1	1
Mix4FIRE	1	1	1	1	1	1	1	1
MMTE	1							
MobiSDN	1	1	1	1	1	1		
PRIORITY	1	1	1	1	1	1	1	1
READINESS	1							
REIGN	1	1	1	1	1		1	1
SETONFIRE	1							
SILVERWoLF	1	1	1	1	1	1	1	1
SoMAC	1	1	1	1	1		1	1
SSC	1	1	1	1	1	1	1	1
Swiich4FIRE	1	1	1	1	1	1	1	1
TELHEX@Fed4FIRE	1	1	1	1	1	1	1	1
TITAN	1							
VIRTUALANDS								
WHISPER	1			1			1	
NOSC	1	1	1	1	1	1	1	1

Table 5: Open Call requirements –Experiment control

	SSH access	Scripted control engine	Threshold based events	Generality of control engine	Ease of use	Time syncrhonisation across testbedstestbeds	Manual interventions of testbed support staff
AnyNetQoS@Fed4FIRE	1	1					
ARISTIDE	1	1		1	1		
BILS	1	1		1	1	1	
CEP4FIRE	1	1	1	1	1		
CloudCONFetti	1	1	1	1			
DANCERS							
EARNEST	1	1		1			
ENA4FI				1			
E-NETDAM	1	1	1	1			
EVIDENCE	1			1			
FED4NAV	1	1		1	1	1	1
FIRE4FoF	1	1	1	1	1	1	
GEO-Cloud	1	1	1	1			
HEHET	1	1					
InLoc	1	1		1	1		
INSURED	1	1	1	1			
IPCS4Fire							
LAMBDA	1	1	1	1			
MEDiANET	1	1	1		1	1	
MEVDDS	1			1			
MH-ScaE	1	1			1		
Mix4FIRE	1	1	1	1			
MMTE							
MobiSDN	1	1	1				
PRIORITY	1	1	1	1			
READINESS							
REIGN	1	1	1	1	1		
SETONFIRE							
SILVERWoLF	1	1	1	1			
SoMAC	1	1	1	1	1	1	
SSC	1	1	1	1			
Swiich4FIRE	1	1	1	1			
TELHEX@Fed4FIRE	1	1	1	1			
TITAN							
VIRTUALANDS	1			1			
WHISPER	1	1		1			
NOSC	1	1	1	1			

Table 6: Open Call requirements - Monitoring

	Measurement support framework	Automatic measurement of common metrics	Wireless interference	Monitoring resources for operational support	Monitoring resources for suitable resource selection and measurement interpretation	Minimal impact of monitoring and measuring tools	On-demand measurements	Demand evaluation
AnyNetQoS@Fed4FIRE	1	1	1	1	1	1	1	
ARISTIDE	1	1				1	1	
BILS	1	1	1			1	1	
CEP4FIRE	1	1				1	1	
CloudCONFetti	1	1	1	1	1	1	1	1
DANCERS			1	1	1	1	1	1
EARNEST	1	1	1	1	1	1	1	
ENA4FI	1	1		1		1	1	
E-NETDAM	1	1	1	1	1	1	1	1
EVIDENCE	1	1			1	1	1	
FED4NAV	1	1	1		1	1	1	
FIRE4FoF	1	1	1			1	1	
GEO-Cloud	1	1	1	1	1	1	1	1
HEHET	1	1	1	1	1	1	1	
InLoc	1	1	1		1	1	1	
INSURED	1	1	1	1	1	1	1	1
IPCS4Fire								
LAMBDA	1	1	1	1	1	1	1	1
MEDiANET	1		1		1	1	1	
MEVDDS	1	1		1	1	1	1	
MH-ScaE	1	1				1	1	
Mix4FIRE	1	1	1	1	1	1	1	1
MMTE	1	1	1		1	1	1	
MobiSDN	1	1			1	1	1	
PRIORITY	1	1	1	1	1	1	1	1
READINESS	1			1	1	1	1	
REIGN	1		1			1	1	
SETONFIRE			1					
SILVERWoLF	1	1	1	1	1	1	1	1
SoMAC	1	1	1		1	1	1	
SSC	1	1	1	1	1	1	1	1
Swiich4FIRE	1	1	1	1	1	1	1	1
TELHEX@Fed4FIRE	1	1	1	1	1	1	1	1
TITAN	1	1		1	1	1	1	1
VIRTUALANDS	1	1		1		1	1	
WHISPER	1	1		1	1	1		
NOSC	1	1	1	1	1	1	1	1

Table 7: Open Call requirements –Permanent storage

	Data storage	Data security	Stored experiment configuration	Data sharing	Stored metadata	Storage management
AnyNetQoS@Fed4FIRE	1	1	1			
ARISTIDE		1	1			
BILS		1	1	1		
CEP4FIRE		1	1	1	1	
CloudCONFetti	1	1	1	1		1
DANCERS						
EARNEST	1	1	1			
ENA4FI	1		1			
E-NETDAM	1	1	1	1		1
EVIDENCE	1	1			1	
FED4NAV		1	1	1		
FIRE4FoF		1	1	1	1	
GEO-Cloud	1	1	1	1		1
HEHET	1	1	1			
InLoc		1	1	1	1	
INSURED	1	1	1	1		1
IPCS4Fire		1				
LAMBDA	1	1	1	1		1
MEDiANET		1	1	1		
MEVDDS	1		1			
MH-ScaE		1	1			
Mix4FIRE	1	1	1	1		1
MMTE						
MobiSDN	1	1	1			
PRIORITY	1	1	1	1		1
READINESS						
REIGN		1	1	1	1	
SETONFIRE						
SILVERWoLF	1	1	1	1		1
SoMAC		1	1	1	1	
SSC	1	1	1	1		1
Swiich4FIRE	1	1	1	1		1
TELHEX@Fed4FIRE	1	1	1	1		1
TITAN						
VIRTUALANDS						
WHISPER	1					
NOSC	1	1	1	1		1

Table 8: Open Call requirements – Dynamic federated identity management

	Single account	Public keys	OpenVPN handling	Authentication of API calls	Low barrier to create a Fed4FIRE identity
AnyNetQoS@Fed4FIRE	1			1	
ARISTIDE	1	1		1	1
BILS	1	1		1	1
CEP4FIRE	1	1		1	1
CloudCONFetti	1	1		1	1
DANCERS				1	1
EARNEST	1	1		1	
ENA4FI	1		1	1	
E-NETDAM	1	1		1	1
EVIDENCE	1			1	
FED4NAV	1	1		1	1
FIRE4FoF	1	1		1	1
GEO-Cloud	1	1		1	1
HEHET	1			1	
InLoc	1	1		1	1
INSURED	1	1		1	1
IPCS4Fire				1	1
LAMBDA	1	1		1	1
MEDiANET	1	1		1	1
MEVDDS	1		1	1	
MH-ScaE	1	1		1	1
Mix4FIRE	1	1		1	1
MMTE				1	1
MobiSDN	1	1	1	1	
PRIORITY	1	1		1	1
READINESS				1	1
REIGN		1		1	1
SETONFIRE				1	1
SILVERWoLF	1	1		1	1
SoMAC	1	1		1	1
SSC	1	1		1	1
Swiich4FIRE	1	1		1	1
TELHEX@Fed4FIRE	1	1		1	1
TITAN				1	1
VIRTUALANDS	1		1	1	
WHISPER	1	1	1	1	
NOSC	1	1		1	1

Table 9: Open Call requirements – Authorization and SLA management

	Per-experimenter restrictions	Temporary experimenter class upgrade	SLA towards companies
AnyNetQoS@Fed4FIRE			
ARISTIDE			1
BILS			1
CEP4FIRE			
CloudCONFetti	1	1	1
DANCERS			
EARNEST			
ENA4FI			
E-NETDAM	1	1	1
EVIDENCE			1
FED4NAV			1
FIRE4FoF			1
GEO-Cloud	1	1	1
HEHET			
InLoc			1
INSURED	1	1	1
IPCS4Fire			
LAMBDA	1	1	1
MEDiANET			1
MEVDDS			
MH-ScaE			1
Mix4FIRE	1	1	1
MMTE			
MobiSDN			
PRIORITY	1	1	1
READINESS			
REIGN			1
SETONFIRE			
SILVERWoLF	1	1	1
SoMAC			1
SSC	1	1	1
Swiich4FIRE	1	1	1
TELHEX@Fed4FIRE	1	1	1
TITAN			
VIRTUALANDS			
WHISPER			
NOSC	1	1	1

Table 10: Open Call requirements –Trust and user experience

	Testbed reliability information	Experiment descriptions	Accountability
AnyNetQoS@Fed4FIRE	1		
ARISTIDE	1		
BILS			
CEP4FIRE			
CloudCONFetti	1	1	
DANCERS			
EARNEST	1		
ENA4FI			
E-NETDAM	1	1	
EVIDENCE			
FED4NAV	1		
FIRE4FoF	1		
GEO-Cloud	1	1	
HEHET	1		
InLoc	1		
INSURED	1	1	
IPCS4Fire			
LAMBDA	1	1	
MEDiANET	1		
MEVDDS			
MH-ScaE			
Mix4FIRE	1	1	
MMTE		1	
MobiSDN			
PRIORITY	1	1	
READINESS			
REIGN			
SETONFIRE			
SILVERWoLF	1	1	
SoMAC	1		
SSC	1	1	
Swiich4FIRE	1	1	
TELHEX@Fed4FIRE	1	1	
TITAN			
VIRTUALANDS			
WHISPER			
NOSC	1	1	

Table 11: Open Call requirements – Interconnectivity

	Layer 3 connectivity between testbeds	Layer 2 connectivity between testbeds	Transparency	Per-slice bandwidth reservation	IPv6 support	Information about testbed Interconnections	Per-slice bandwidth limitation	Automatic notification of interconnectivity problems
AnyNetQoS@Fed4FIRE	1	1	1			1		
ARISTIDE	1		1			1		
BILS	1		1			1		1
CEP4FIRE								
CloudCONFetti	1	1	1	1	1	1	1	1
DANCERS								
EARNEST	1	1	1	1		1		
ENA4FI	1	1	1					1
E-NETDAM	1	1	1	1	1	1	1	1
EVIDENCE	1	1	1			1		
FED4NAV	1		1			1		1
FIRE4FoF	1		1		1	1	1	1
GEO-Cloud	1	1	1	1	1	1	1	1
HEHET								
InLoc								
INSURED	1		1		1	1		
IPCS4Fire								
LAMBDA	1	1	1	1	1	1	1	
MEDiANET	1		1			1	1	1
MEVDDS	1		1			1		1
MH-ScaE	1		1			1	1	
Mix4FIRE	1		1		1	1		
MMTE						1		
MobiSDN	1	1	1			1		
PRIORITY	1	1	1	1	1	1	1	1
READINESS								
REIGN								
SETONFIRE					1	1		
SILVERWoLF	1	1	1	1	1	1	1	1
SoMAC	1		1			1	1	
SSC	1		1		1	1		
Swiich4FIRE	1	1	1	1	1	1	1	
TELHEX@Fed4FIRE	1		1		1	1		
TITAN						1		
VIRTUALANDS	1		1			1		1
WHISPER	1		1			1		
NOSC	1		1	1	1	1	1	

4 Identified requirements

The following section describes the set of requirements identified, their priority and development status after cycle 1. The list is based on the requirements defined at the start of the 1st development cycle but was updated with new requirements originating from both the open calls as well as the new set of use cases. Several pieces of information are collected in the corresponding requirements tables:

- Source: for requirements deriving from use cases, this is the short descriptor of the use case. The labels are as follows:
  - TEACH: Teaching computer science using FIRE facilities
  - WIRELESS: Testing a networking solution for wireless building automation on different platforms
  - GEO-ELAS: Researching the concept of geographical elasticity in cloud computing
  - BENCHM: Benchmarking a service platform for information retrieval
  - MOBILE-CL: Mobile cloud service platform
  - BGP : Design and evaluation of alternative inter-domain routing schemes
  - TOURIST: Touristic video on demand services in smart cities
  - GEN: When the requirement is shared among several use cases, it is considered as generic -GEN.
- Requirement Id: Requirement Identifier to ease tracing ("I.Area.number")
  - I stands for Infrastructures
  - Areas:
    - 1: Experiment Workflow and Lifecycle Management
    - 2: Measurement and Monitoring
    - 3: Trustworthiness
    - 4: Interconnection
  - Requirement number: 001, 002, etc.
- Requirement statement: Brief description of the requirement.
- Requirement description: Descriptive text for the requirement.
- Priority: the priority has been set according to the following criteria (from the source use cases point of view):
  - High: Should be implemented for the second development cycle of Fed4FIRE
  - Medium: Should be implemented in the third (and last) development cycle of Fed4FIRE
  - Low: nice to have but not essential.
- Comments: additional information regarding the requirement
- Req. coverage cycle 1: This column indicates whether a requirements has been covered in cycle 1: Y (Yes), N (No) or PARTIAL (P).
- **Open Call Influence (OCI): this column gives information regarding which requirements are the result of the analysis of the open calls. If a cell in this column is empty, this means that the requirement on that row was not changed compared to cycle 1. If it mentions "C", then** it was changed because of the open call analysis. If it mentions "N", then it is a new requirement introduced by the open call analysis.
- N. exp: Number of experiments requesting this requirement. For requirements derived from proposals presented to the open calls, this represents the number of proposals that would benefit from this requirement.

- P1: Number of points assigned to the requirement as far as use cases are concerned.
- P2: Number of points assigned to the requirements as far as proposals from the open call are concerned.
- Total: number of total points (PT = P1+P2)
- Priority: High, Medium, Low (*)

(*) Once requirements have been grouped, the prioritisation has been made according to how many proposals and use cases benefit from a requirement, equal to the prioritisation adopted in WP4 in D4.2.

The procedure followed has been to assign points to each requirement according to the following rules:

- Number of use cases requesting a requirement:
  - The requirement is only interesting for a single use case (points: 1)
  - The requirement is shared among several use cases (points: 2)
- Number of open calls requesting a requirement, with 37 proposals analyzed:
  - X<12 (points: 1)
  - 12<=x<24 (points: 2)
  - 24<=x (points: 3)

According to this scoring system, requirements can sum up to 5 points. Thus we assign a final priority as follows:

- If a requirement has a total number of points between 0 and 2, it is a low-level priority requirement.
- If a requirement has a total number of points of 3 or 4, it is a medium-level priority requirement.
- Requirements having 5 points are high-level priority ones.

4.1 Experiment workflow and lifecycle management

Experiment lifecycle management includes resource description and discovery, resource requirements of the experiment, resource reservation, resource provisioning, experiment control, monitoring and permanent storages.

4.1.1 Resource discovery

Source scenario	Req. id	Req.	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.1.101	Node capabilities	Fed4FIRE must provide a clear view on what node capabilities are available, and this should be defined in the same way across the federation. This view should be returning all the nodes that are offered by the testbeds, and should not filter out those that have been reserved for now. This should also not only include hardware characteristics such as CPU type or available types of network interfaces, but should also contain information about other capabilities such as is it mobile, can it be accurately steered remotely, etc. If resources have a static relation with each other (e.g. node X is installed onto mobile robot Y) then this should also be represented.	Node capabilities can be described in terms of CPU architecture and speed, RAM, supported 802.11 standards, optical networking interfaces, software defined radio, measurement resource type, OpenFlow support, etc. It can be beneficial to adopt proven standards to represent these capabilities (e.g., FOAM which provides a comprehensive OpenFlow resource description).	Y	C	30	2	3	5	High
GEN	I.1.102	Accurate location information	Wireless nodes should provide accurate location information (1 m accuracy). For this location it should also be known with which kind of environment it corresponds (outdoor, office, industrial indoor, etc.)	Coordinates should be displayed both in text as on a map showing the actual topology. If the actual distance between any pair of nodes can easily be retrieved, this would also be valuable.	N		13	2	2	4	Medium

Source scenario	Req. id	Req.	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.1.103	Wireless spectrum as a resource	The different available wireless channels should be considered to be an infrastructure resource.	If channels are a resource, they can easily become discovered and reserved later on.	N		9	2	1	3	Medium
GEN	I.1.104	Site location information	For all nodes the location of the site where they are physically deployed should be known.	Per site the location might be the same value for all nodes, no need for the accuracy level of I.1.002 here.	P		13	2	2	4	Medium
GEN	I.1.105	Discovery through federation-wide APIs	Resource discovery must be integrated into uniform tools through federation-wide APIs. Ideally, these APIs would be compatible with discovery APIs already supported by the infrastructures and/or existing uniform tools. This would decrease the development costs for the infrastructure providers and tool builders.	The APIs supported by all infrastructures in the Fed4FIRE federation should be able to support both the Fed4FIRE portal and any other standalone tool that wishes to adopt them. Therefore the APIs should be well documented.	Y		27	2	3	5	High
GEN	I.1.106	Intra-infrastructure topology information	For nodes that have wired and/or wireless network connections to other nodes within the same testbed, it should be possible to identify the physical topology. This relates to connections which are part of the data plane of an experiment, not the control interfaces. Similar, if virtualized topologies are supported, the corresponding possibilities should also be communicated to the experimenter. As a result, it should be easy for experimenters to assess the scale of the testbed.	Examples of wireless topologies are Wi-Fi or 802.15.4 connectivity charts (possibly with variable channel and modulation type selection), or connectivity map between WAN base stations and nodes. Examples of virtualized topologies are those based on wavelength allocations in the optical domain, or those based on VLAN configurations in Emulab.	N	C	31	2	3	5	High

Source scenario	Req. id	Req.	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.1.107	Inter-infrastructure topology information	It should be known how different infrastructures are/can be interconnected. Important parameters are the type of interconnection (layer 2, layer 3), and the support for bandwidth reservation. If resources are also reachable beyond the boundaries of the Fed4FIRE partners' infrastructures (e.g., because they are directly connected to the public Internet), this should also be mentioned. Information regarding IPv6 support on the inter-infrastructure topologies is also required.		N		26	2	3	5	High
GEN	I.1.108	Background info virtualized resources	Resource virtualization (VMs, flows) must provide information about the supporting physical devices and their location		N		6	2	1	3	Medium
GEN	I.1.109	Query search	It should be able to build tools that allow a query search for suitable infrastructures/nodes	An experimenter should be able to fill in some specific technical details about the hardware he/she is looking for, and it should be possible for the resource discovery tool to construct a suitable response based on the resource information provided by the infrastructures.	Y		12	2	3	5	High
GEN	I.1.110	Catalogue search	If an experimenter does not know which parameters to fill in using the query search, it should be able to browse through some kind of Fed4FIRE infrastructures catalogue		Y		11	2	1	3	Medium

Source scenario	Req. id	Req.	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
			to find pointers towards the suitable facilities. Likewise, when in doubt regarding resources returned by the query search, such a catalogue would also be useful.								

4.1.2 Resource requirements

Source scenario	Req. id	Req.	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.1.201	Manually extract requirements from discovery query results	When the query in the discovery phase returns a certain list of resources, it should be possible for the experimenter to select the resources he/she would like to include in the experiment. This should be supported in relation with a specific resource ID (e.g., I want this specific node at this specific Wi-Fi testbed).		Y		25	2	3	5	High
GEN	I.1.202	Extract requirements from discovery query results in an orchestrated manner	It is possible that the query in the discovery phase returns a list of resources which are all suitable for the planned experiment. In this case it should be possible for the experimenter to define the requirements in such a way to define a bigger group of candidates (e.g., all Emulab instances, or all		N		3	2	1	3	Medium

Source scenario	Req. id	Req.	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
			PlanetLab Europe sites in the Benelux.), and let the reservation /provisioning tools select a suitable node at experiment runtime, based on availability.								
GEN	I.1.203	Describing required virtualized topologies	A Federated API is needed which would enable specifying the desired virtualized topologies that will be deployed over the existing physical topology	Examples are drawing a topology on the Virtual Wall that will be automatically translated to a correct selection of machines and VLAN configuration on all ports. Another example is defining the topology of a Flowspace on an OFELIA infrastructure.	N		4	2	1	3	Medium
WIRELESS	I.1.204	Requiring additional resources which are not yet exposed by the testbed	The experimenter should be able to specify the need for additional experimentation devices (such as spectrum analyzers or Smartbits measurement devices). These can be devices of which the experimenter knows that the testbeds could optionally provide them, although that they are not exposed as available resources. But these could also be additional equipment (both supporting equipment but also devices under test) owned by the experimenter himself which he temporarily installs at the testbed	E.g., in the discovery phase wireless resources were searched that have 2 IEEE 802.11n interfaces per node. Since there is a large amount of suitable nodes, the experimenter has the luxury to define additional requirements. In this case, the experimenter does not only wants to define the node requirements, but also the fact that a spectrum analyzer should be present at the same site.	N	C	3	1	1	2	Low

4.1.3 Resource reservation

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.1.301	Hard resource reservation	Fed4FIRE must provide hard reservations of the available resources. It should be able to perform immediate reservations (starting from now), or more advanced reservations (given a specific future timeslot that the experimenter would want, or have the reservation system look for the first available slot where all desired resources are available).	It should even be possible to reserve all nodes that could interfere with an experiment, even if they are not actually used during the experiment. E.g., the experimenter could not trust channel reservation, since in reality orthogonal channels often do interfere due to imperfect radio hardware implementations. Therefore he/she wants to reserve all nodes in a specific infrastructure for the experiment. Another example resource for which hard reservation would be indispensable is that of wavelengths in optical OFELIA resources.	P		30	2	3	5	High

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.1.302	Fairness	A means to enforce fairness with hard reservations is required. Situations where a few users reserve all nodes for too long should be avoided. Similarly, situations where a large amount of users reserve a few resources for a very long time should also be avoided. This kind of reservations is typically done to develop new solutions on the testbed, but makes it harder to schedule other large-scale experiments.	This could be achieved by specifying an expiration date for reservations via calendar or a scheduler, or through the usage of reservation quota. It could be interesting to look at existing techniques applied in high performance computing clusters.	N		24	2	3	5	High
GEN	I.1.303	Secure reservation	Fed4FIRE must provide a reservation system with adequate security to provide assurance to industrial users		P		24	2	3	5	High
GEN	I.1.304	Automated reservations handling	The Fed4FIRE reservation system should be able to approve/deny reservation requests in a fully automated manner, without any manual intervention by the infrastructure operators.		P		26	2	3	5	High
GEN	I.1.305	Reservation information	Fed4FIRE must provide the information on the availability of resources at a specific timeslot. The other way around, it should also be possible for experimenters and infrastructure providers to receive a clear view on which resources are already reserved, and when.		P		12	2	2	4	Medium
	I.1.306	Experiment planning assistance	When an experimenter has defined all resources that it wants to include in its experiment, and the desired duration of the experiment, then he should be shown a screen which gives information about all	Could be considered to be a tool requirement instead of an architectural requirement.	N	N	8	0	1	1	Low

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
			the oncoming timeslots in which all the nodes of the experiment are still available. Selecting the desired timeslot and reserving all of them should be possible with the click of a single button.								

4.1.4 Resource provisioning

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.1.401	Provisioning API	APIs are required to enable direct instantiation of both physical and virtualized resources for experiments. Ideally, these APIs would be compatible with provisioning APIs already supported by the infrastructures and/or existing uniform tools. This would decrease the development costs for the infrastructure providers and tool builders.	Instantiation of physical node would involve powering the node on, and appropriately steering the node boot process. Instantiation of virtualized resources can be related to the setup of virtual machines, OpenFlow flows, etc. For such virtualized resources the API should support an annotation mechanism to define the actual physical resource on which the virtual one should be instantiated.	Y		31	2	3	5	High
GEN	I.1.402	Customizing Linux	Fed4Fire must provide the ability to install a specific custom Linux kernel or distribution on the nodes	Experimenters could e.g., require specific Linux kernels because some experimental hardware drivers might only	N		24	2	3	5	High

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
				be supported on such a specific kernel. It can also allow them to deploy specific Linux distributions, e.g., OpenWrt. Infrastructure providers could assist their experimenters by providing several pre-installed Linux distributions (Some Ubuntu Long Time Support versions, Fedora distributions, OpenWrt, etc.) to choose from.							
GEN	I.1.403	Root access	Fed4FIRE must provide the possibility to access a node as root user	Often experimenters will install additional software on their resources. This can be external software packages, compiled code, new hardware drivers, and so on. To do so, root access to the node is required. In many cases the experimenters also want to configure the network interfaces according to their experimentation needs. This also requires root access.	Y		25	2	3	5	High

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.1.404	Internet access to software package repositories	In Fed4FIRE software installation through a packet manager (e.g., apt-get) must be possible. Hence the package manager should have Internet access to external software package repositories.		Y		25	2	3	5	High
GEN	I.1.405	Hard disk imaging	Once experimenters have finished the configuration of their nodes, they should be able to create a binary image of the entire hard disk drive, which can be stored and reloaded to the node in a future experiment. This allows to setup experiments which need a longer runtime, but pause its execution from time to time to allow other experimenters to also use the testbed in case of hard reservation of the resources.	Infrastructures can operate perfectly without HDD imaging support. In that case experimenters have to write appropriate installation scripts that are automatically started at boot-time.. However, manually installing all software once on a node, and creating an image from that prototype is more convenient/efficient. HDD imaging support also allows infrastructure operators to provide pre-configured images that include specific valuable functionality. An example would be a fully configured GNU radio image that can be flashed to nodes connected to a SDR.	P	C	28	2	3	5	High

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.1.406	Automated network stitching	In case of experiments that require layer 2 connectivity between different infrastructures, the network stitching between them should be performed automatically by the Fed4FIRE system.		N		17	2	2	4	Medium
GEN	I.1.407	Automated software installation at boot time	In advance you should have been able to already define what these specific resources should do at startup (install additional software, copy specific files, start specific daemons/tools)		P	N	25	2	3	5	High
	I.1.408	Provisioning error notification	In case some of the resources cannot be provisioned successfully, you should be informed so that you can decide to continue the experiment, or change it to another testbed or reserve again later on. Such information coming from all the different testbeds involved in a single experiment should be presented to the experimenter in a single location	Could be considered to be a tool requirement instead of an architectural one.	P	N	22	2	2	4	Medium

4.1.5 Experiment control

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.1.501	SSH access	Nodes must be accessible via SSH.	This is most valuable during the development phase or for debugging purposes.	Y		30	2	3	5	High
GEN	I.1.502	Scripted control engine	It must be possible to describe advanced experiment scenarios by the use of a script that will be executed by a control engine. The engine will perform all required shell commands on the appropriate resources at the appropriate time. Ideally, this control engine would be compatible with engines already supported by some of the infrastructures. This would decrease the development costs for the infrastructure providers.	This way the experimenter can alter the behaviour of the resources in an automated manner from a single location, without having to manually login on all nodes during experiment runtime. This is not only more convenient, but also increases repeatability and hence scientific value of the experimental results. It also allows the quicker setup of complex experiments at different infrastructures.	P		27	2	3	5	High
GEN	I.1.503	Threshold based events	Next to time based events, events based on monitored metrics/thresholds should be supported by the experiment control engine.	The geographical-elasticity and mobile cloud service solutions under test could also be in charge of monitoring the load on its services, and activating the elasticity process/service relocation when needed. However, if the control engine would support this, this would be more convenient/efficient for the experimenter. Hence the	P		18	2	2	4	Medium

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
				medium priority is considered appropriate.							
GEN	I.1.504	Generality of control engine	The experiment control engine should be general enough to support the control of all possible kinds of Future Internet technology: wireless networks, optical networks, OpenFlow devices, cloud computing platforms, mobile robots, etc. If it is not feasible to control all these resource types with a single tool, you should at least be able to use the same tool per domain (e.g. one tool for all cloud computing tasks, one for all wireless tasks, and one to control the robots movements).		P	C	26	2	3	5	High
GEN	I.1.505	Ease of use	The experimenter should be able to describe the whole experiment in a human readable and uniform way.		P		10	2	1	3	Medium
BGP	I.1.506	Time synchronisation accross testbeds	If you want to perform scenarios across multiple testbeds relying on time-based events, it is essential that the internal clocks of the resources across all testbeds are synchronized very accurately.		N	N	5	1	1	2	Low
	I.1.507	Manual interventions of testbed support staff	It can be necessary that an experimenter require manual live participation by a person present (e.g. walking around an office environment with a smartphone to validate indoor positioning solutions). For this one should be able to contact the actual support staff of the testbed to make		N	N	1	0	1	1	Low

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
			practical arrangements (local staff does it for the experimenter, or the experimenter travels to the testbed for one day)								

4.2 Measurement and monitoring

Measuring and monitoring covers procedures and tools supporting the observation and measurement of system and experimentation facility properties. Monitoring is useful for many reasons - accounting, decision-making in resource management, accountability - who used what when, capacity planning and management.

4.2.1 Monitoring

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.2.101	Measurement support framework	Fed4FIRE must provide an easy way for experimenters to store measures during the experiment runtime for later analysis. The data should be clearly correlated to the experiment ID.	Measurements can be related to common metrics for which existing tools such as ping or iperf can be used. However, they can also be very specific to the experiment, and hence calculated somewhere within the experimental software under test. It should be possible to take measurements on a large variety of resources: Linux servers/embedded devices, OpenFlow packet switches	P		34	2	3	5	High

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
				and optical devices, cellular base stations, etc.							
GEN	I.2.102	Automatic measurement of common metrics	Common characteristics should be stored automatically during an experiment (CPU load, free RAM, Tx/Rx errors, etc.)	.	N		31	2	3	5	High
GEN	I.2.103	Wireless interference	Information about external wireless interference during the execution of the experiment should be provided.	The interference can be detected using monitor interfaces in dedicated nodes and/or spectrum analysers that offer more exact results. This functionality should be easily provided to every experimenter who is not “spectrum analysing” expert.	N	C	25	2	3	5	High
GEN	I.2.104	Monitoring resources for operational support	Fed4FIRE must provide tools to continuously monitor the state of the resources so testbed managers can prevent and/or solve problems with them. In case of detected issues with the infrastructure, Fed4FIRE should warn the facility providers about them. If experimenters are actually trying to use these resources at that moment, they should also be informed.	Examples of metrics to be monitored are: state of the power source, actual energy consumption, is SSH working on the node, is telnet working on the node, etc.	Y		27	2	3	5	High

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.2.105	Monitoring resources for suitable resource selection and measurement interpretation	Fed4FIRE must also provide the monitoring info regarding the state of the resources to the experimenters. This way they can choose the best resources for their experiments. This information also provides the experimenters with the means to distinguish error introduced by the experiment from errors related to the infrastructure.	In this monitor view that an experimenter has on his/her resources, it could also be interesting to display some non-monitored background information, for instance the IP address of the control interface, the DNS name, etc.	P		27	2	3	5	High
GEN	I.2.106	Minimal impact of monitoring and measuring tools	As less overhead as possible should be expected from the monitoring and measurement support frameworks. The impact of the measurement tools over the experiment results should be negligible.		Y		35	2	3	5	High
GEN	I.2.107	On-demand measurements	The user must be able to request on-demand measurements. In order to do so, they will need to express that they want agents with such on-demand polling capacities	The same information can be retrieved by looking into the output of the monitoring and measurement tools that will continuously provide measurements during the experiment run-time. However the on-demand measurement is more convenient during experiment development and debugging.	N		34	2	3	5	High
GEN	I.2.108	Demand evaluation	Infrastructure providers will need to evaluate experimenters' measurements request automatically in order to know if they can be met. If not, the	If the measurement is not available, the returned zero or random values will most likely be noticeable by the	N		14	2	2	4	Medium

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
			experimenters should be informed about this.	experimenter. However, a formal notification of missing measurements (e.g., because a given metric is not applicable in all domains) is more convenient.							

4.2.2 Permanent storage

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.2.201	Data storage	Fed4FIRE must provide the means to store the data measured during an experiment. This data should be accessible during and after the experiment, and should be clearly correlated to the experiment run ID		N		20	2	2	4	Medium
GEN	I.2.202	Data security	Access to the data should be properly secured		N		28	2	3	5	High
GEN	I.2.203	Stored experiment configuration	Experiment configurations should be stored in order to replay experiments and compare results of different runs. These configurations should be versioned in a way that corresponds with significant milestones in the experiment development.	Experiment configurations can contain deployment descriptors, experiment control scripts, etc.	N		28	2	3	5	High

GEN	I.2.204	Data sharing	When desired, it should be possible for an experimenter to share stored experiment data or configurations with specific individuals, groups of people or even make them publically available.		N		20	2	2	4	Medium
GEN	I.2.205	Stored metadata	Fed4FIRE should give the possibility to store metadata about the data of the experiments.	This allows easy lookup of experiment results, and eases the assessment of the meaning of the data. This seems most valuable when looking for shared data of other experimenters.	N		6	2	1	3	Medium
GEN	I.2.206	Storage management	Storage space must be monitored/limited. Bad or useless stored data should be identified so it can be deleted		N		12	2	2	4	Medium

4.3 Trustworthiness

Trustworthiness gathers tools and services targeted to increase the trust in FIRE considering community views on trust (e.g. academic vs. commercial), id management, SLA management, etc.

4.3.1 Dynamic federated identity management

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.3.101	Single account	Fed4FIRE must provide the mean of accessing all testbeds within the federation using one single account (username/password). Ideally, this	This means both accessing the web interfaces of the federated infrastructures, as accessing the actual	Y		30	2	3	5	High

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
			authentication framework would be compatible with those already supported by some of the infrastructures. This would decrease the development costs for the infrastructure providers.	resources belonging to the experiment, retrieving the experiment results, and so on.							
GEN	I.3.102	Public keys	Fed4FIRE should also provide authentication by the use of public SSH keys.	It is possible that for some resources it is technically more feasible to authenticate through public SSH keys. Therefore Fed4FIRE should not only provide the single account based on username/password, but also on a pair of public/private keys.	Y		25	2	3	5	High
GEN	I.3.103	OpenVPN handling	Fed4FIRE should take into account that some facilities are now behind an OpenVPN based authentication system. A seamless relation with the single Fed4FIRE account should be put in place, or the OpenVPN based interconnections should be abandoned.		N		5	2	1	3	Medium
GEN	I.3.104	Authentication of API calls	Access to the Fed4FIRE APIs (discovery, reservation, provisioning, etc.) should also be protected by an authentication mechanism		Y		37	2	3	5	High
GEN	I.3.104	Low barrier to create a Fed4FIRE identity	It should be easy for new experimenters without any affiliation to the federation to create their Fed4FIRE identity.		P	N	28	2	3	5	High

4.3.2 Authorization

Source scenario	Req. Id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.3.201	Per-experimenter restrictions	It should be possible for infrastructures to dynamically decide which resources they should make available to a certain Fed4FIRE experimenter, and which experimentation quota that will be appropriate. This can be bases on a set of possible experimenter roles, on specific attributes, etc.	Example of roles could be: master student, PhD student, post-doc, professor, paying customer, etc. Example of attributes could be: affiliation, years of experience, credit card limit, etc.	N		12	2	2	4	Medium
GEN	I.3.202	Temporary experimenter class upgrade	Fed4FIRE should provide the possibility for an experimenter to temporarily use more resources than he/she is allowed according to their experimenter class. This could be useful in specific cases, such as a close publication submission deadline.		N		12	2	2	4	Medium

4.3.3 SLA management

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.3.301	SLA towards companies	In the use cases where the experimenters are affiliated with a company, it can be interesting for them to have some ideas about expected up- and downtime, etc. This allows them to plan their developments tighter, resulting in a reduced cost and time-to-market.	Even companies realize that the Fed4FIRE facilities are state-of-the-art experimental infrastructures, and will most likely not require the same kind of SLAs as they would from their other services providers. In the case of academic research, the demand for SLAs is rather low.	N		22	2	2	4	Medium

4.3.4 Trust and user experience

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.3.401	Testbed reliability information	Fed4FIRE should provide a method for querying and reporting the reliability of a testbed in terms of provided hardware, software and present wireless interference.	A possible approach could be to monitor facility resources to observe service experience. Regular questioning of the experimenters about their experience could be another possibility. In this case attention should be given to minimizing the burden on the experimenter, while making sure that untrue vicious feedback is not considered.	P		21	2	2	4	Medium

Source scenario	Req. id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
				Anyway, both the monitoring and the feedback approaches would need specific functionality to be in place in the Fed4FIRE federation.							
GEN	I.3.402	Experiment descriptions	In Fed4FIRE experimenters that create an experiment will need to provide a short high-level description of the experiment and its purpose. This allows infrastructure providers to keep track of the usage of the infrastructure, and enables them to report about this to their funding sources.	Funding and sustainability is a key issue for all infrastructures.	P		13	2	2	4	Medium
GEN	I.3.403	Accountability	Fed4FIRE should provide the possibility to trace network traffic back to the originating experiment. This is useful when misuse of the infrastructure has been detected and the corresponding experimenter should be sanctioned (e.g., by revoking his/her account). The fact that accountability mechanisms are in place will automatically increase the level of trust that infrastructure providers can have in Fed4FIRE experimenters which are unknown to them.	FIRE facilities can be powerful tools, and misuse should most definitely be handled adequately.	N		0	2	1	3	Medium

4.4 Interconnectivity

Source scenario	Req. Id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
GEN	I.4.001	Layer 3 connectivity between testbeds	The resources within a Fed4FIRE infrastructure should be able to reach the resources deployed in the other Fed4FIRE infrastructures through a layer 3 Internet connection.	Ideally all infrastructures are connected to high-capacity research Internet backbones such as Géant.	Y		27	2	3	5	High
GEN	I.4.002	Layer 2 connectivity between testbeds	For some experiments it can be required that the included testbeds are interconnected through a layer 2 link.		N		12	2	2	4	Medium
GEN	I.4.003	Transparency	Providers must be able to offer in a transparent way the resources of all the federated testbeds. Interconnectivity solutions should not introduce unneeded complexity in the experiment.	Solutions based on VPN or other tunnels require that the experimenter is aware of the corresponding configurations when developing the experiment, while he/she should be concentrating on the content of the experiment, and not these practical preconditions. Besides, VPN tunnels will work initially, but they will not scale when a larger number of infrastructures has to be interconnected,	P		27	2	3	5	High

Source scenario	Req. Id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
				due to conflicts in address spaces.							
GEN	I.4.004	Per-slice bandwidth reservation	Per experiment, Fed4FIRE should provide the possibility to reserve bandwidth on the links that interconnect specific infrastructures.		N		9	2	1	3	Medium
GEN	I.4.005	IPv6 support	The ability to conduct IPv6 measurements and to interact with the nodes of other testbeds over IPv6 should be enabled.	After interaction with its research community, iMinds reported that IPv6 support is crucial since some of the testbeds (e.g. Virtual Wall and w-iLab.t) need to support IPv6 due to a shortage of public IPv4 addresses and the need for transparent interconnectivity. If resources of the other testbeds need to interact with these IPv6-based testbeds, it is a must that these other testbeds also support IPv6, or no communication between the resources will be possible. Because of this	P		14	2	2	4	High

Source scenario	Req. Id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
				element, it was chosen to upgrade the priority of this requirement to high.							
GEN	I.4.006	Information about testbed Interconne-ctions	The experimenter needs to know how the several testbeds are interconnected e.g., via layer 3 or layer 2. Especially he/she needs to know which gateways should be used by the resources in order to interconnect them along with other testbed resources. The experimenter also wants to know the type of interconnection between the testbeds used in its experiment (dedicated direct links, interconnected through Géant using best effort, interconnected through Géant with bandwidth reservation, connected to the public internet but not to Géant, ...)		N	C	29	2	3	5	High
GEN	I.4.007	Per-slice bandwidth limitation	Since many of the F4F testbeds are connected to géant, it can be expected that the interconnectivity between testbeds is not representative of the actual interconnections of a realistic deployment using the normal public internet. Therefore experimenters should be able to easiliy limit the bandwidth		N	N	12	2	2	4	Medium

Source scenario	Req. Id	Req. statement	Req. description	Comments	Req. coverage cycle 1	OCI	N. exp	P1	P2	Total	Priority
			available between sites to emulate more realistic settings.								
TOURIST	I.4.008	Automatic notification of inter-connectivity problems	If problems arise with the interconnectivity of testbeds, then the experimenter should be actively informed about this.		N	N	12	1	2	3	Medium

5 Conclusion: enumeration of the high-priority requirements

As described in the previous section the prioritisation mechanism used has permitted to identify the high-priority requirements by weighting the importance at the federation level for the use cases and open calls analysed. These requirements are considered essential and their implementation should be included in the second cycle developments as much as possible and as long as this is feasible in terms of other constraints (e.g. effort).

5.1 Experiment Workflow and Lifecycle Management

Req. id	Req. statement	Req. description	Comments
I.1.101	Node capabilities	Fed4FIRE must provide a clear view on what node capabilities are available, and this should be defined in the same way across the federation. This view should be returning all the nodes that are offered by the testbeds, and should not filter out those that have been reserved for now. This should also not only include hardware characteristics such as CPU type or available types of network interfaces, but should also contain information about other capabilities such as is it mobile, can it be accurately steered remotely, etc. If resources have a static relation with each other (e.g. node X is installed onto mobile robot Y) then this should also be represented.	Node capabilities can be described in terms of CPU architecture and speed, RAM, supported 802.11 standards, optical networking interfaces, software defined radio, measurement resource type, OpenFlow support, etc. It can be beneficial to adopt proven standards to represent these capabilities (e.g., FOAM which provides a comprehensive OpenFlow resource description).
I.1.105	Discovery through federation-wide APIs	Resource discovery must be integrated into uniform tools through federation-wide APIs. Ideally, these APIs would be compatible with discovery APIs already supported by the infrastructures and/or existing uniform tools. This would decrease the development costs for the infrastructure providers and tool builders.	The APIs supported by all infrastructures in the Fed4FIRE federation should be able to support both the Fed4FIRE portal and any other standalone tool that wishes to adopt them. Therefore the APIs should be well documented.
I.1.106	Intra-infrastructure topology information	For nodes that have wired and/or wireless network connections to other nodes within the same testbed, it should be possible to identify the physical topology. This relates to connections which are part of the data plane of an experiment, not the control interfaces. Similar, if virtualized topologies are supported, the corresponding possibilities should also be communicated to the experimenter. As a result, it should be easy for experimenters to assess the scale of the testbed.	Examples of wireless topologies are Wi-Fi or 802.15.4 connectivity charts (possibly with variable channel and modulation type selection), or connectivity map between WAN base stations and nodes. Examples of virtualized topologies are those based on wavelength allocations in the optical domain, or those based on VLAN configurations in Emulab.

Req. id	Req. statement	Req. description	Comments
I.1.107	Inter-infrastructure topology information	It should be known how different infrastructures are/can be interconnected. Important parameters are the type of interconnection (layer 2, layer 3), and the support for bandwidth reservation. If resources are also reachable beyond the boundaries of the Fed4FIRE partners' infrastructures (e.g., because they are directly connected to the public Internet), this should also be mentioned. Information regarding IPv6 support on the inter-infrastructure topologies is also required.	

Req. id	Req. statement	Req. description	Comments
I.1.201	Manually extract requirements from discovery query results	When the query in the discovery phase returns a certain list of resources, it should be possible for the experimenter to select the resources he/she would like to include in the experiment. This should be supported in relation with a specific resource ID (e.g., I want this specific node at this specific Wi-Fi testbed).	

Req. id	Req. statement	Req. description	Comments
I.1.301	Hard resource reservation	Fed4FIRE must provide hard reservations of the available resources. It should be able to perform immediate reservations (starting from now), or more advanced reservations (given a specific future timeslot that the experimenter would want, or have the reservation system look for the first available slot where all desired resources are available).	It should even be possible to reserve all nodes that could interfere with an experiment, even if they are not actually used during the experiment. E.g., the experimenter could not trust channel reservation, since in reality orthogonal channels often do interfere due to imperfect radio hardware implementations. Therefore he/she wants to reserve all nodes in a specific infrastructure for the experiment. Another example resource for which hard reservation would be indispensable is that of wavelengths in optical OFELIA resources.

Req. id	Req. statement	Req. description	Comments
I.1.302	Fairness	A means to enforce fairness with hard reservations is required. Situations where a few users reserve all nodes for too long should be avoided. Similarly, situations where a large amount of users reserve a few resources for a very long time should also be avoided. This kind of reservations is typically done to develop new solutions on the testbed, but makes it harder to schedule other large-scale experiments.	This could be achieved by specifying an expiration date for reservations via calendar or a scheduler, or through the usage of reservation quota. It could be interesting to look at existing techniques applied in high performance computing clusters.
I.1.303	Secure reservation	Fed4FIRE must provide a reservation system with adequate security to provide assurance to industrial users	
I.1.304	Automated reservations handling	The Fed4FIRE reservation system should be able to approve/deny reservation requests in a fully automated manner, without any manual intervention by the infrastructure operators.	

Req. id	Req. statement	Req. description	Comments
I.1.401	Provisioning API	APIs are required to enable direct instantiation of both physical and virtualized resources for experiments. Ideally, these APIs would be compatible with provisioning APIs already supported by the infrastructures and/or existing uniform tools. This would decrease the development costs for the infrastructure providers and tool builders.	Instantiation of physical node would involve powering the node on, and appropriately steering the node boot process. Instantiation of virtualized resources can be related to the setup of virtual machines, OpenFlow flows, etc. For such virtualized resources the API should support an annotation mechanism to define the actual physical resource on which the virtual one should be instantiated.
I.1.402	Customizing Linux	Fed4Fire must provide the ability to install a specific custom Linux kernel or distribution on the nodes	Experimenters could e.g., require specific Linux kernels because some experimental hardware drivers might only be supported on such a specific kernel. It can also allow them to deploy specific Linux distributions, e.g., OpenWrt. Infrastructure providers could assist their experimenters by providing several pre-installed Linux distributions (Some Ubuntu Long Time Support versions, Fedora distributions, OpenWrt, etc.) to choose from.
I.1.403	Root access	Fed4FIRE must provide the possibility to access a node as root user	Often experimenters will install additional software on their resources. This can be external software packages, compiled code, new hardware drivers,

Req. id	Req. statement	Req. description	Comments
			and so on. To do so, root access to the node is required. In many cases the experimenters also want to configure the network interfaces according to their experimentation needs. This also requires root access.
I.1.404	Internet access to software package repositories	In Fed4FIRE software installation through a packet manager (e.g., apt-get) must be possible. Hence the package manager should have Internet access to external software package repositories.	
I.1.405	Hard disk imaging	Once experimenters have finished the configuration of their nodes, they should be able to create a binary image of the entire hard disk drive, which can be stored and reloaded to the node in a future experiment. This allows to setup experiments which need a longer runtime, but pause its execution from time to time to allow other experimenters to also use the testbed in case of hard reservation of the resources.	Infrastructures can operate perfectly without HDD imaging support. In that case experimenters have to write appropriate installation scripts that are automatically started at boot-time.. However, manually installing all software once on a node, and creating an image from that prototype is more convenient/efficient. HDD imaging support also allows infrastructure operators to provide pre-configured images that include specific valuable functionality. An example would be a fully configured GNU radio image that can be flashed to nodes connected to a SDR.
I.1.407	Automated software installation at boot time	In advance you should have been able to already define what these specific resources should do at startup (install additional software, copy specific files, start specific daemons/tools)	

Req. id	Req. statement	Req. description	Comments
I.1.501	SSH access	Nodes must be accessible via SSH.	This is most valuable during the development phase or for debugging purposes.
I.1.502	Scripted control engine	It must be possible to describe advanced experiment scenarios by the use of a script that will be executed by a control engine. The engine will perform all required shell commands on the appropriate resources at the appropriate time. Ideally, this control engine would be compatible with engines already supported by some of the infrastructures. This would decrease the development costs for the infrastructure	This way the experimenter can alter the behaviour of the resources in an automated manner from a single location, without having to manually login on all nodes during experiment runtime. This is not only more convenient, but also increases repeatability and hence scientific value of the experimental results. It also allows the quicker setup of complex

Req. id	Req. statement	Req. description	Comments
		providers.	experiments at different infrastructures.
I.1.504	Generality of control engine	The experiment control engine should be general enough to support the control of all possible kinds of Future Internet technology: wireless networks, optical networks, OpenFlow devices, cloud computing platforms, mobile robots, etc. If it is not feasible to control all these resource types with a single tool, you should at least be able to use the same tool per domain (e.g. one tool for all cloud computing tasks, one for all wireless tasks, and one to control the robots movements).	

5.2 Measurement & Monitoring

Req. id	Req. statement	Req. description	Comments
I.2.101	Measurement support framework	Fed4FIRE must provide an easy way for experimenters to store measures during the experiment runtime for later analysis. The data should be clearly correlated to the experiment ID.	Measurements can be related to common metrics for which existing tools such as ping or iperf can be used. However, they can also be very specific to the experiment, and hence calculated somewhere within the experimental software under test. It should be possible to take measurements on a large variety of resources: Linux servers/embedded devices, OpenFlow packet switches and optical devices, cellular base stations, etc.
I.2.102	Automatic measurement of common metrics	Common characteristics should be stored automatically during an experiment (CPU load, free RAM, Tx/Rx errors, etc.)	.
I.2.103	Wireless interference	Information about external wireless interference during the execution of the experiment should be provided.	The interference can be detected using monitor interfaces in dedicated nodes and/or spectrum analysers that offer more exact results. This functionality should be easily provided to every experimenter who is not “spectrum analysing” expert.

Req. id	Req. statement	Req. description	Comments
I.2.105	Monitoring resources for suitable resource selection and measurement interpretation	Fed4FIRE must also provide the monitoring info regarding the state of the resources to the experimenters. This way they can choose the best resources for their experiments. This information also provides the experimenters with the means to distinguish error introduced by the experiment from errors related to the infrastructure.	In this monitor view that an experimenter has on his/her resources, it could also be interesting to display some non-monitored background information, for instance the IP address of the control interface, the DNS name, etc.
I.2.106	Minimal impact of monitoring and measuring tools	As less overhead as possible should be expected from the monitoring and measurement support frameworks. The impact of the measurement tools over the experiment results should be negligible.	
I.2.107	On-demand measurements	The user must be able to request on-demand measurements. In order to do so, they will need to express that they want agents with such on-demand polling capacities	The same information can be retrieved by looking into the output of the monitoring and measurement tools that will continuously provide measurements during the experiment run-time. However the on-demand measurement is more convenient during experiment development and debugging.

Req. id	Req. statement	Req. description	Comments
I.2.202	Data security	Access to the data should be properly secured	
I.2.203	Stored experiment configuration	Experiment configurations should be stored in order to replay experiments and compare results of different runs. These configurations should be versioned in a way that corresponds with significant milestones in the experiment development.	Experiment configurations can contain deployment descriptors, experiment control scripts, etc.

5.3 Trustworthiness

Req. id	Req. statement	Req. description	Comments
I.3.101	Single account	Fed4FIRE must provide the mean of accessing all testbeds within the federation using one single account (username/password). Ideally, this authentication framework would be compatible with those already supported	This means both accessing the web interfaces of the federated infrastructures, as accessing the actual resources belonging to the experiment, retrieving the experiment results, and so on.

Req. id	Req. statement	Req. description	Comments
		by some of the infrastructures. This would decrease the development costs for the infrastructure providers.	
I.3.102	Public keys	Fed4FIRE should also provide authentication by the use of public SSH keys.	It is possible that for some resources it is technically more feasible to authenticate through public SSH keys. Therefore Fed4FIRE should not only provide the single account based on username/password, but also on a pair of public/private keys.
I.3.104	Authentication of API calls	Access to the Fed4FIRE APIs (discovery, reservation, provisioning, etc.) should also be protected by an authentication mechanism	
I.3.104	Low barrier to create a Fed4FIRE identity	It should be easy for new experimenters without any affiliation to the federation to create their Fed4FIRE identity.	

5.4 Interconnectivity

Req. Id	Req. statement	Req. description	Comments
I.4.001	Layer 3 connectivity between testbeds	The resources within a Fed4FIRE infrastructure should be able to reach the resources deployed in the other Fed4FIRE infrastructures through a layer 3 Internet connection.	Ideally all infrastructures are connected to high-capacity research Internet backbones such as Géant.
I.4.003	Transparency	Providers must be able to offer in a transparent way the resources of all the federated testbeds. Interconnectivity solutions should not introduce unneeded complexity in the experiment.	Solutions based on VPN or other tunnels require that the experimenter is aware of the corresponding configurations when developing the experiment, while he/she should be concentrating on the content of the experiment, and not these practical preconditions. Besides, VPN tunnels will work initially, but they will not scale when a larger number of infrastructures has to be interconnected, due to conflicts in address spaces.
I.4.004	Per-slice bandwidth reservation	Per experiment, Fed4FIRE should provide the possibility to reserve bandwidth on the links that interconnect specific	

Req. Id	Req. statement	Req. description	Comments
		infrastructures.	
I.4.005	IPv6 support	The ability to conduct IPv6 measurements and to interact with the nodes of other testbeds over IPv6 should be enabled.	After interaction with its research community, iMinds reported that IPv6 support is crucial since some of the testbeds (e.g. Virtual Wall and w-iLab.t) need to support IPv6 due to a shortage of public IPv4 addresses and the need for transparent interconnectivity. If resources of the other testbeds need to interact with these IPv6-based testbeds, it is a must that these other testbeds also support IPv6, or no communication between the resources will be possible. Because of this element, it was chosen to upgrade the priority of this requirement to high
I.4.006	Information about testbed Interconne-ctions	The experimenter needs to know how the several testbeds are interconnected e.g., via layer 3 or layer 2. Especially he/she needs to know which gateways should be used by the resources in order to interconnect them along with other testbed resources. The experimenter also wants to know the type of interconnection between the testbeds used in its experiment (dedicated direct links, interconnected through Géant using best effort, interconnected through Géant with bandwidth reservation, connected to the public internet but not to Géant, ...)	

References

[1] Scognamiglio, Sioutis, Vandenberghe et al.; “D3.1 – Infrastructures community federation requirements, version 1”, deliverable of the Fed4FIRE project, 2012