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**

D3.1 - Infrastructures community federation requirements, version 1

Work package	WP3 Infrastructures
Task	Task 3.1 Community requirements
Due date	30/11/2012
Submission date	30/11/2012
Deliverable lead	Timur Friedman (UPMC)
Version	1.0
Authors	Ciro Scognamiglio (UPMC) Michael Sioutis (UPMC) Wim Vandenberghe (iMinds) Stefan Bouckaert (iMinds) Okung-Dike Ntofon (UNIVBRIS) Georgios Androulidakis (NTUA) Thierry Parmentelat (INRIA) Carlos Bermudo Abad (i2CAT) Donatos Stavropoulos (UTH) Kostas Choumas (UTH) Harris Niavis (UTH)

Reviewers	Ciro Scognamiglio (UPMC) Michael Sioutis (UPMC) Wim Vandenberghe (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
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

In Fed4FIRE, two communities have been 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 D4.1, which describes requirements for the services and applications community.

A federation of experimentation facilities is very important, because it will significantly accelerate Future Internet research. This will be made possible by delivering open and easily accessible facilities to the FIRE experimentation communities, which focus on fixed and wireless infrastructures, services and applications, and combinations thereof.

In keeping with the overall project workflow, the requirements captured and synthesized here will be given to WP2 Architecture as input for the first development cycle.

Five varied use cases have been developed in order to illustrate the benefits of joining several testbeds together. These use cases have been designed by different project partners based on their experiences in real-world collaborative scenarios involving heterogeneous resources and thus represent research and commercial trends:

- **Teaching computer science using FIRE facilities** is a scenario that involves FIRE facilities used to teach computer science to students and introduce them to the basics of IP networking. Using a specific required topology (e.g., a topology representing a typical corporate IT deployment) laboratory exercises can be assigned to students.
- **Testing a networking solution for wireless building automation on different platforms** describes how building automation systems lowers the total cost of ownership, increases the security level, and raises the comfort of the people inside the building. Modern building automation systems are build on bus systems, this scenario describes how replacing the wired bus with a wireless network would result in significant cost savings and would also enable new applications such as indoor positioning.
- **Researching the concept of geographical elasticity in cloud computing** describes a scenario where techniques of horizontal and vertical elasticity can be used to handle increasing user demands. The concept of elasticity allows quick down scaling and up scaling. This means that peaks in demand can be met while no excess capacity is wasted when not needed.
- **Benchmarking a service platform for information retrieval** involves information retrieval from a document archive, such as that of a large newspaper containing all articles ever published. The document processing would be based on text analysis and training documents, not on matches with existing lists of words. This means that in this case, processing of a document is a very calculation intensive task.
- **Mobile cloud service platform** describes a way to build and test a software service to the diverse characteristics of wireless based platforms and adapt it to problems linked to each platform by relying to a cloud based service platform.

Each use case has produced a set of requirements from the perspective of what would be required to deploy them in a federated infrastructure. These requirements have been assembled under the functional areas that Fed4FIRE has considered for achieving a federation:

- **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.

These requirements have also been prioritised (high, medium, low) according to how critical they are for the use case (or cases) that produced them. If requirements were valid for the majority of the described use cases, they received the status “generic”. The generic requirements with a high priority are considered essential and their implementation should be included in the first cycle of developments as much as possible and as long as this is feasible in terms of other constraints (e.g. effort). They are listed in section 4, and can be considered as the main input towards WP2.

Acronyms and Abbreviations

CDN	Content Data Network
FI-PPP	Future Internet Public Private Partnership
GUI	Graphical User Interface
IP	Internet Protocol
LTE	Long Term Evolution
M2M	Machine to Machine
Wi-Fi	Wireless Fidelity

Table of Contents

1 Introduction 8
2 Use cases 10
3 Requirements 31
3.1 Experiment workflow and lifecycle management 32
3.1.1 Resource discovery 32
3.1.2 Resource requirements 34
3.1.3 Resource reservation 36
3.1.4 Resource provisioning 38
3.1.5 Experiment control 40
3.2 Measurement and monitoring 41
3.2.1 Monitoring 41
3.2.2 Permanent storage 43
3.3 Trustworthiness 44
3.3.1 Dynamic federated identity management 44
3.3.2 Authorization 45
3.3.3 SLA management 46
3.3.4 Trust and user experience 46
3.4 Interconnectivity 48
4 Requirements matrix 50
Appendix A: Overview of experiment lifecycle management 52
Appendix B: Overview of trustworthiness 53
Appendix C: Questionnaire as distributed to the WP3 partners 54
Appendix D: Future Internet Research Facilities 67

1 Introduction

The purpose of this document, "D3.1 Infrastructure community federation requirements", is to gather requirements from the Infrastructure community's perspective in order to build a federation of FIRE facilities. It is the first deliverable in a cycle of three which will all focus on these same requirements.

The infrastructures community is large and varied, and can be structured around the types of facilities that experimenters use, which are connected with the types of questions that they are investigating. We have identified nine different types of facilities (see Appendix D for more details):

- PlanetLab-based facilities
- Open network measurement infrastructures
- Optical testbeds
- Switching testbeds
- Emulation testbeds
- Wireless LAB testbeds (Wi-Fi, Bluetooth, etc.)
- Software defined radio testbeds
- Sensor networking / embedded object testbeds
- Cellular wireless testbeds (LTE, 3G, WiMAX, etc.)

These are general categories, and of course some facilities might include technologies from more than one category.

There are Fed4FIRE WP3 partners working with each of these kinds of facilities. Considering the relatively short time available for the preparation of this first deliverable (submission deadline is M2 of the project), we chose to focus on collecting requirements only from these partners in this stage of the project. In future iterations, we intend to look at requirements expressed by experimenters and owners of these types of facilities beyond the Fed4FIRE partners. To collect requirements, we distributed a specific questionnaire (see appendix C). Based on the inputs received from every WP3 partner, this deliverable could then compiled. Careful consideration was taken to ensure that the questionnaire and hence the corresponding input covers both the infrastructure provider and the experimenter point of view. This way the maximum level of quality is pursued within the established timing boundaries. Additional sources of input are however recognized as valuable inputs. Therefore, in the other two requirements deliverables, requirements will also be gathered from:

- Running FIRE IP projects (there is at least one partner in Fed4FIRE from each of the FIRE IP projects)
- FIRESTATION and the new FIRE CSAs
- The project's non-European partners, the many international contacts that the Fed4FIRE partners have in the community, and in general from members of the community who respond to Fed4FIRE invitations to requirement gathering activities.

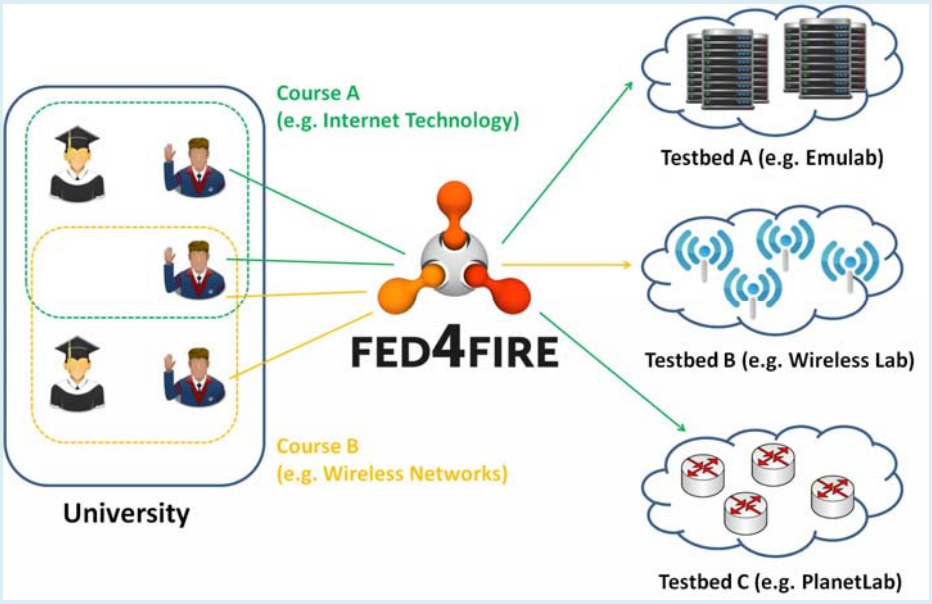
This document is structured as follows:

- Chapter 2 gathers the description of initial use cases including, among other information, a storyboard and the technology that will be involved.
- Chapter 3 provides high level requirements derived from these use cases
- Chapter 4 contains a summarized version of the most important requirements. This can be considered as the main input towards the architectural work of D2.1.

2 Use cases

This section includes some of the use cases that have been studied as a first source of requirements for the Infrastructures Community.

Scenario Name	SCENARIO 1: Teaching computer science using FIRE facilities
Background / Rationale	In this use case, FIRE facilities are used to teach computer science to students. To introduce students to the basics of IP based networking, lab exercises could be given that start from a specific required topology (e.g., representing a typical corporate IT deployment). Students should then have to decide how to configure the subnets and individual IP addresses to make optimal use of the available address space. Given some specific limitations, they should also design the firewall rules to make sure that some services on the public Internet can be accessed (websites of the main suppliers, SAP system of the headquarter located somewhere else, etc.). On the other hand, access to other services should be prohibited (Facebook, MSN messenger, etc.). To validate their design, the students should have to actually configure their setup on a live experiment. Other labs could for instance focus on IEEE 802.11 technology (Wi-Fi). Doing exercises on real hardware, students could get a clear view on the aspects that can negatively influence end to end throughput (interference by other Wi-Fi networks which operate on non-orthogonal other channels, difference in range when using the 2.4 or the 5 GHz band, etc.). The difference in security between the different Wi-Fi security schemes, such as WEP, WPA, WPA2., could also be taught very efficiently when students are provided with practical exercises in which they actually attempt to hack these different deployments. These are just a few examples of possible lab exercises which provide students with more thorough insights in the technologies that they face. Other teaching domains could be OpenFlow networking, wireless sensor networks, cloud computing, high performance computing, mobile wide area data networks, and so on. To support all these domains with lab exercises on actual hardware, significant investments are needed both in terms of hardware, maintenance, and development of the exercises. However, through a FIRE federation, universities could rely on the available federated infrastructures for their labs. Hence, no specific hardware should be purchased, and no manpower is to be invested in installation tasks. Additionally, if the Fed4FIRE federation would adopt common tools for resource discovery, reservation, provisioning, experiment control and monitoring, it would be sufficient to get acquainted with one set of tools to be able to setup lab exercises targeting quite diverging courses. This means that professors and assistants could really focus on their core activity in this context: developing the actual exercises with a high educational value. Since different universities could all adopt the usage of federated FIRE facilities for teaching, it would be even possible to exchange exercises between universities. Such a focused cooperation could again bring the educational value of the developed exercises to a higher level.

Scenario Name	**SCENARIO 1: Teaching computer science using FIRE facilities**
	It is clear that teaching computer science using FIRE facilities enables universities to develop lab exercises with less effort than required today, while in fact the educational value could be higher. But other parties are also positively influenced by this methodology. Again, if the Fed4FIRE federation would adopt common tools for resource discovery, reservation, provisioning, experiment control and monitoring, it would sufficient to get acquainted with one set of tools to be able to perform the different lab exercises belonging to the different courses on the curriculum ("Internet Technology", "Wireless Networks", "Distributed Systems", etc.). Additionally, if one common approach to authentication and authorization was adopted in the federation, students could execute all labs without losing time requesting different accounts for each of them. As a result of both the above assumptions, the Fed4FIRE federation would allow students to really focus on the content of the labs. Since, the students of today are the employees and entrepreneurs of tomorrow, this approach in teaching computer science could result in a higher demand for FIRE experimentation in the long run. This would both be beneficial for the FIRE community, and for the European Internet Industry for which innovation is a key aspect.
Picture	
Scenario description (Storyboard)	This storyboard only focuses on the example of teaching the basics of IP based networking. It is expected that all other cases will be very similar. A professor in communication networks wants to develop a lab exercise as described in the rationale above. The lab would start from a given required topology. In our example, the topology represents a desired deployment in an imaginary corporation. There are workstations and servers deployed over different subnets, connected by a router. A second router provides Internet access, and will also run the firewall software. Students are given a certain IPv4 address range, and have to make optimal use of this available address space. They also have to configure the firewall so that the websites of the main

Scenario Name	SCENARIO 1: Teaching computer science using FIRE facilities

suppliers and the SAP system of the headquarter can be reached. On the other hand, access to other websites, Facebook and MSN messenger should be prohibited.

The first step is to find appropriate infrastructure to support this experiment. The deployment of the corporate environment could be implemented using both virtual or physical machines running some flavour of Linux. However, it is important as an experimenter to control the way these machines are interconnected. This way the actual desired topology can be used. It does not matter if all nodes are located in the same FIRE infrastructure, as long as they appear to be part of the same LAN. To test the firewall rules, however, some fake sites and services should also be deployed somewhere else on the public Internet as part of the exercise.

Taking these constraints into account, the professor surfs to the Fed4FIRE portal. There, he/she can access a tool for resource discovery. After requiring the resources to be PCs running Linux, with at least one wired interface for experiments, and the possibility to control the actual topology, a list of possible nodes is returned. A large amount of them seems to be part of an infrastructure called “iMinds Virtual Wall”. The professor then goes to the facilities catalogue on the Fed4FIRE portal to have some more background information about this infrastructure, which seems to be an Emulab instance. After reading the corresponding flyer, it is clear that this is a suitable infrastructure to support the exercise. However, the fake services on the public Internet are still missing. These could be emulated on the Virtual Wall, but the professor would rather want these actually deployed on servers on the public internet. He/she therefore goes back to the resource discovery tool, omits the requirement for controllable topology, and adds the requirement that nodes should be directly connected to the public Internet, without any firewalling. The returned list of candidates clearly refers to an infrastructure called PlanetLab Europe (PLE). Again some more background information is retrieved from the facilities catalogue, and PLE is identified as the missing link for the lab exercise.

The next step is to setup one experiment that covers the entire lab exercise. On the Fed4FIRE portal, the professor uses the Fed4FIRE resource reservation tool to reserve 7 nodes on the Virtual Wall (3 workstations, 2 servers, and 2 routers), and 4 virtual machines on PLE, spread across 4 different locations (a fake supplier website, a fake website, a fake SAP system in the imaginary headquarter, and a fake MSN service). When selecting the desired PLE nodes, available information regarding geographical dispersion and node reliability is taken into account. For all nodes the same version of Ubuntu Linux is the operating system of choice. This should make it easier for students to configure the different nodes in the

Scenario Name	SCENARIO 1: Teaching computer science using FIRE facilities

exercise. For the professor it is also easier to configure the fake services, since Ubuntu is familiar to him/her. Another aspect that eases the setup of this experiment is the fact that all nodes could be added to the experiment using a single tool on the Fed4FIRE portal. There is no need to get familiar with two different tools corresponding with the two different underlying testbeds.

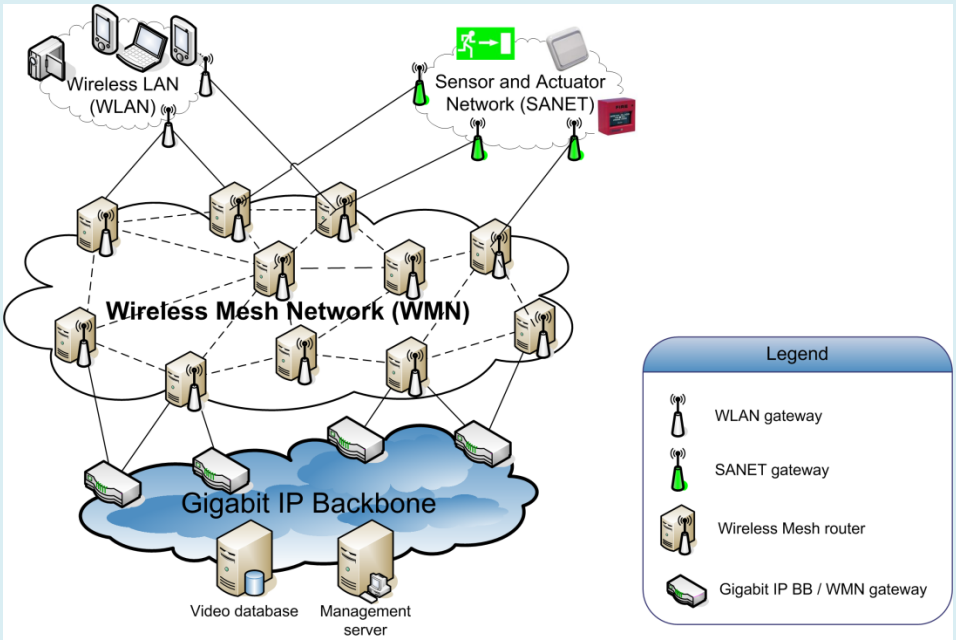
When the resources are provisioned, the professor wants to draw the correct topology between the Virtual Wall nodes. On the experiment monitoring page, a table is shown that displays the status of each node (in this case "up"), but there is an additional button "Configure topology" next to the Virtual Wall nodes. When clicked, this opens a screen where all 7 nodes are depicted, and links can easily be drawn between them. Once a link is established, characteristics such as bandwidth and delay can be configured.

Now that all required infrastructure is available, the professor installs and/or activates all needed software libraries on the nodes (webserver, firewall, etc.). To do so he/she logs in to the node using a standard SSH client. The IP addresses corresponding to the control interface of the different nodes are depicted in the experiment monitoring table on the Fed4FIRE portal. It is convenient that thanks to the Fed4FIRE authentication framework, the same login and password can be used on all nodes, no matter which testbed the node belongs to. In fact, these credentials are the same as the ones used to gain access to the Fed4FIRE portal in the first place.

Once this prototype of the entire setup is finished, it is permanently stored so that it can be used later in a new experiment. Since 10 groups of students will be working on the lab at the same time, 10 identical setups are deployed in parallel the day before the lab. The duration of the reservation is configured long enough so that they remain active until the end of the lab. The actual setup is entirely deployed by the professor. He/she creates specific user accounts on the Linux nodes of each separate setup. The corresponding credentials are handed out to the students during the actual lab. Just before the start of the lab, the professor has a quick look at the monitoring page to validate that the nodes of his 10 experiments are all still up and running. This model assumes that the lab is a typical physical lab, where students gather in one PC room during a pre-defined time, together with the professor and his/her assistants. Another possibility could be to organize the labs more as homework. In that case, students should be able to deploy a setup of their own when they see fit. As a result, they should have a student account on the Fed4FIRE platform. This type of account will most likely be more restricted or restrained than that of researchers or teachers, but it should allow students to deploy setups created by their teachers. Some mechanism is needed so that the teachers can annotate which student accounts

Scenario Name	SCENARIO 1: Teaching computer science using FIRE facilities
	receive the right to deploy a specific lab experiment. From the students' point of view, it is convenient that they can use the same Fed4FIRE account to execute the labs corresponding to the different courses on their curriculum. When finished with the exercise, the students have to upload their network interface configuration files and firewall configuration file to the permanent storage server where they remain available to the professor, even when the experiment is over. Later on, these files will be used to grade the students. Of course, it should be guaranteed that no user other than the professor or his assistants can access these files. The students can now start preparing themselves for another course for which they have a lab exercise tomorrow. Since this exercise will again be supported by facilities federated within Fed4FIRE, the students will be able to use the same account and tools as today. Hence they can focus their preparation on rehearsing the theory of the corresponding course, instead of focusing on practical aspects with limited education value. The professor, on the other hand, can evaluate the usefulness of the lab exercise, share it with some colleagues at other universities, and possibly collaborate with them to further enhance the educational value for the next year's students.
Technology involved	Physical PC/server, Virtual Machines, wired LAN/WAN access, topology configuration through Emulab, OpenFlow networking, wireless sensor networks, IEEE 802.11, cloud computing, high performance computing, mobile wide area data networks

Scenario Name	**SCENARIO 2: Testing a networking solution for wireless building automation on different platforms.**
Background / Rationale	Building automation systems lowers the total cost of ownership, increases the security level and raises the comfort of the people inside the building. Modern building automation systems build on bus systems such as EIB or its successors KNX, LON and BACNET. Replacing this wired bus with a wireless network, would result in significant cost savings, and would also enable new applications, such as indoor positioning. It would also allow the automation of hard to reach locations and historical buildings where it is difficult to install wires. However, the deployment of such wireless networks is hindered by a number of fundamental technological problems. The biggest challenge we face is the fact that the wireless network has to fulfil contradictory requirements. On the one hand, sensors and actuators have to be as energy-efficient as possible because they are battery powered, while on the other hand the network should be performant enough to support video surveillance and replace the wired backbone. In addition, the network should be able to serve IEEE 802.11 devices and it should be scalable to support the number of devices that can be found in a large building. A networking technology that simultaneously fulfils those requirements does not exist. Instead, heterogeneous technologies have to be combined into an appropriate networking solution. In this use case, an SME has developed a solution that consists of four different kinds of networks: a wired backbone, a wireless mesh network (WMN), wireless LANs (WLANs), and sensor and actuator networks (SANETs). The figure below depicts this cohesion of different technologies. • The wired backbone will typically be a gigabit Ethernet backbone, and can thus achieve a much higher throughput than the wireless technologies. • The WMN connects the WLAN and SANET to the wired backbone. The use of WMN nodes with multiple interfaces should enable high throughputs. By means of dynamic channel selection and power control, the topology can be controlled and interference can be reduced. The WMN should also be self-healing and self-organising. • The SANET is responsible for transmitting monitor and control information. This involves low data rate traffic, a domain where IEEE 802.15.4 is a suitable technology. The SANET should be self-healing, self-organising, and since most wireless sensors and actuators are battery powered, the SANET nodes should consume as little energy as possible. • The WLAN is responsible for serving IEEE 802.11 a/b/g/n devices, such as surveillance cameras, smartphones, laptops, speakers, and video screens. Those devices require a high bandwidth and a low latency (e.g., for video surveillance). The wired backbone and WLAN are fairly standard components which are of no real interest to the SME. However, both the WMN and the WSN are considered as interesting research topics. Innovation in these domains would open the doors

Scenario Name	**SCENARIO 2: Testing a networking solution for wireless building automation on different platforms.**
	for the SME to pioneer this novel market of wireless building automation. Due to the different characteristics of the WMN and SANET networks, they were developed as two separate solutions. To guarantee interoperability between them, they were both made IP compatible. Hence the WMN was designed as an IP-based ad hoc network relying on the AODV protocol. On top of this, the SME researchers developed novel distributed channel selection, power control, and load balancing protocols. The SANET, on the other hand, was based on the DYMO-low protocol, which is IP compatible since it adopts the 6LoWPAN specifications. On top of this, several optimizations on both the MAC and network layers were implemented that should significantly reduce the power consumption of each SANET node.
Picture	
Scenario description (Storyboard)	The SME wants to use the federated FIRE facilities to perform the following three different tests: 1. Quantify the effectiveness of the WMN optimizations. The most important metric will be the total throughput from a selection of WMN nodes representing the WLAN gateways (which would act as entry points for video data originating from wireless IP cameras connected to the WLAN) to one specific WMN node that represents the wired backbone gateway (where the video database will be connected to). 2. Quantify the effectiveness of the SANET optimizations. The most important metric will be the power consumption of the SANET nodes. 3. Test the compatibility between the WMN and SANET. For this test a ping test will be performed from a WMN node to a SANET node. The corresponding packet success rate and round-trip time will be evaluated here.

Scenario Name	**SCENARIO 2: Testing a networking solution for wireless building automation on different platforms.**
	The initial development of the WMN solutions is performed on a small dedicated setup at the SME premises, which consists of five standard PC's, each equipped with two IEEE 802.11 a/b/g/n interfaces. For the first larger scale experiment on the FIRE facilities, the SME researchers first want to test with as little external interference as possible, just to validate that the solution works. This would make it easier to debug the implementation before proceeding with performance evaluation in more realistic building automation environments. The researcher opens the Fed4FIRE portal and starts the resource discovery tool. He/she enters the following requirements: Intel x86 compatible CPU, minimum 128 MB of RAM, possibility to run Ubuntu Linux, minimum two .11a/b/g/n interfaces, at least 30 nodes present in the same location, and an outdoor environment. The returned table displays many nodes originating from the NITOS testbed. Next to each node, there is a button to display the specific testbed topology on a separate page. Using this functionality, the researcher learns that the NITOS testbed provides a suitable grid of WMN nodes on the roof of a building. The researcher decides to run the first experiment on these specific nodes on this specific testbed. To do so, the researcher can select all desired nodes in the table that was returned by the resource discovery tool, and make them part of a new experiment. He/she can also configure the version of Ubuntu Linux to be installed on all nodes. Using the scheduling functionality of the Fed4FIRE portal, the researcher starts the experiment, knowing that it is guaranteed to remain active for the next 5 hours. Since the implemented dynamic channel selection algorithm will try to use all channels in the specific 2.4 or 5 GHz band, he/she also reserves all corresponding channels for the duration of the experiment. Note that because the researcher is affiliated with an SME, the reservation systems will give his/her requests a higher priority. Once the nodes are provisioned, he/she logs in to one of the WMN nodes using a standard SSH client and his/her standard Fed4FIRE credentials. Additional libraries are installed on this node using the Ubuntu packaging system apt-get, and specific WMN networking software is copied to the node using the SCP protocol. Final configurations are done to ensure that at boot time the network interfaces will be configured appropriately, and that the WMN software will be automatically started. Once complete, the hard disk of this node is imaged and stored on the testbed for later reuse. The researcher changes the experiment description so that all nodes will be provisioned with this specific disk image from now on. He/she restarts the provision phase of his active experiment, and after a few minutes, all NITOS nodes on the roof of the building are running the specified WMN solution of the SME. To verify correct WMN interconnectivity, the SME researcher logs in to several nodes using SSH, and manually performs some basic networking tests

Scenario Name	SCENARIO 2: Testing a networking solution for wireless building automation on different platforms.

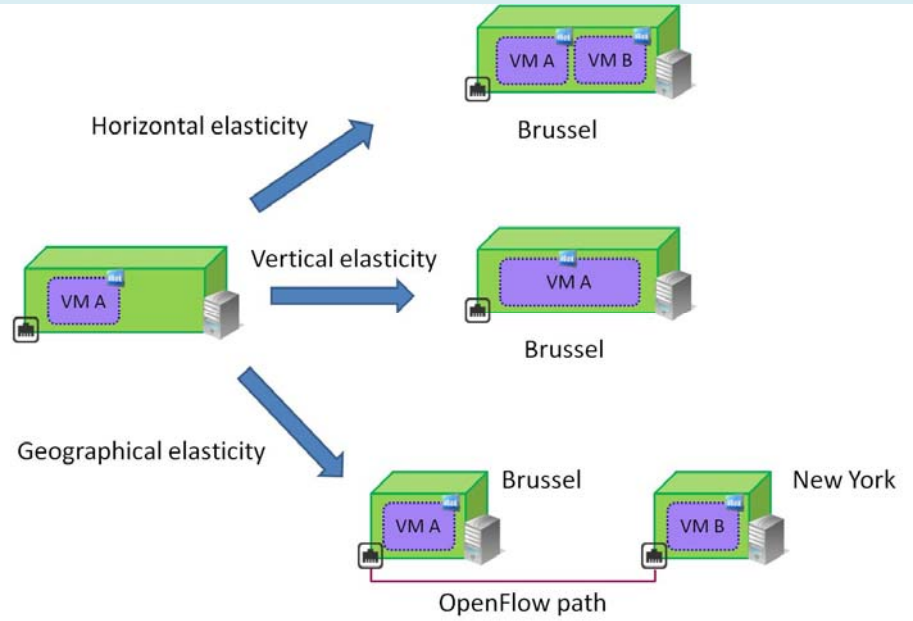
such as ping, iperf, and so on.

Since the code seems to work well, the next step is to perform actual performance tests. In this context of wireless building automation, indoor environments are more suitable for this type of tests. When changing the required environment from outdoor to office, the resource discovery tool returns nodes belonging to different testbeds: NORBIT, NETMODE, and w-iLab.t. After inspecting their topologies, connectivity maps, and background information on the facilities catalogue of the Fed4FIRE portal, is the researcher decides that all of them are interesting test environments. To strengthen the value of the test results, the researcher decides to evaluate the performance of the SME solution for all these testbeds. For each of them, he/she creates a specific experiment and an appropriate hard drive image. As a result, an operational version of his/her WMN solution can be easily deployed on all testbeds. However, to thoroughly quantify the effect of the WMN optimizations, the researcher designs a scenario in which 6 specific WMN nodes have to send streams of random data to a specific 7^{th} node. At well defined moments in time, the data rate of these streams has to be simultaneously increased. If the testbeds federated in Fed4FIRE would all adopt some common experiment control tools, the experimenter could easily run the same advanced scenario automatically on all envisaged testbeds. The actual received data rate at the 7^{th} node has to be logged in function of time for later analysis. Additionally, the amount of RF interference should also be logged in function of time. The latter allows the SME researchers to assess the scientific value of the results. For this purpose, one of the testbed software defined radios is included in the experiment, and provisioned with a default hard drive image that provides channel assessment functionality. To log both types of results, the researcher would like to use a common federation measurement tool. This would again reduce the implementation overhead when testing a single solution on several federated testbeds.

After demonstrating the added value of the developed WMN solutions on the different FIRE facilities, the focus of the SME R&D team shifts towards the SANET implementation. These tests will be handled similar to the WMN tests. First, basic functionality will be tested in an interference-free environment. The resource discovery tool of the Fed4FIRE portal in this case will return nodes of the w-iLab.t Zwijnaarde testbed. When searching for office environments, the w-iLab.t office testbed will be returned. Again the common experiment control and measurement tools will be adopted, since the researchers are already familiar with them.

Scenario Name	**SCENARIO 2: Testing a networking solution for wireless building automation on different platforms.**
	Finally, to validate interoperability between WMN and SANET, the researcher searches for hardware that supports both IEEE 802.11a/b/g/n and IEEE 802.15.4 technology. On top, the nodes should be placed in an interference-free environment. The resource discovery tool returns nodes from the w-iLab.t Zwijnaarde testbed. Because the student is already familiar with this testbed now, he/she can easily define the required experiment.
Technology involved	IEEE 802.11a/b/g/n, IEEE 802.15.4, Software Defined Radio

Scenario Name	**SCENARIO 3: Researching the concept of geographical elasticity in cloud computing**
Background / Rationale	In cloud computing, horizontal and vertical elasticity are two techniques to handle increasing user demands. Horizontal elasticity allows rapid change on the number of separate VM instances. Vertical scalability allows you to rapidly change the size of VM instances themselves in a flexible way (e.g., increasing RAM size, dedicated CPU cores, etc.). The concept of elasticity allows quick down scaling and up scaling. This means that peaks in demand can be met while no excess capacity is wasted when not needed. In this use case, a researcher has developed the concept of geographical elasticity. Again, elasticity is used to increase the amount of resources under high demand, and to decrease them in case of low demand. The main idea here is that if users of a cloud service can be clustered in two distinct locations, then the corresponding VM could be split into two instances which are deployed close to these two locations. One VM remains the master server, containing all data and/or functionality. The other VM is only provided with the specific data needed to serve most users from its own region. However, if a user's request actually requires data that is only available on the master instance, a reliable path should be already established between them. For this purpose OpenFlow is used. The established OpenFlow path is continuously monitored, and automatically adjusted when needed. The development of such a flow controller is one of the key challenges for the researcher. Another important topic for him/her is the algorithm that decides when and how to geographically split and merge. The example that inspired the researcher to foster the idea of geographical elasticity is that of a personalized interactive form of sports events viewing. For instance, in case of motor sport events, several commentary audio streams could be made available to the user. One stream could be commented by a former motor sports engineer, focusing on many technical details. On the other stream the comment could be given by a former racing pilot, focussing on the driving aspects during the race. Two other streams could focus more on background information regarding specific teams. For instance, Ferrari fans would tune into the first stream, while McLaren fans would tune into the second one, and so on. The selection of which audio stream is sent to which user should be done automatically, based on the user's online profiles, previous viewing behaviour, and so on. The same approach should be followed for the video stream. If you are a neutral user, you would get a more generic montage with a balanced overview of the race, while if you are fan of a specific team, you would see their cars more often than in the generic montage. If you are interested in the driving aspects, you would see a lot more on-board clips than normal, and so on. Again the suitable stream for a certain user should be automatically determined. On top of the audio and video streams, interactive functions should also be provided. At the touch of a button, a digital map of the track could be displayed that shows the actual position of all cars in the race, or a table could be shown that enumerates the current lap times of all cars, or short replays of crashes and overtakes could be started, and so on.

Scenario Name	**SCENARIO 3: Researching the concept of geographical elasticity in cloud computing**
	Imagine that a new European start-up company would deploy such a commercial service. Initially it would deploy this cloud service on a single location in Europe. However, as the service gains momentum, the load on the service increases and some additional resources have to be initiated. In this case, geographical elasticity could prove to be a more efficient solution than horizontal or vertical elasticity. The automatic analysis tools could for example conclude that on one hand, the European users mainly watch Formula One races, while on the other hand the US users mainly watch NASCAR racing. Hence, the master service is kept in the EU, and a slave service is automatically initiated in the US. Only the files and services that are related to the NASCAR races are mirrored on the slave. Since both services in general will be hosted relatively close to the corresponding viewers, network aspects such as delay and jitter should be minimized. This should result in the best possible viewer's Quality of Experience. In case a US viewer wants to tune into a Formula One race, the required data is immediately transferred from the EU service instance to the US one using the OpenFlow path between both instances. When the load on the service reduces again, it can be automatically revert back to the state where there was only one service instantiation, which would be the master services in the EU.
Picture	
Scenario description (Storyboard)	The researcher wants to experimentally validate the possible added value of this novel concept. For this purpose, he/she wants to perform the following experiment: 1. A mock-up web service is deployed on a European cloud computing platform belonging to the Fed4FIRE federation. The platform should support both horizontal and vertical elasticity. 2. A relatively large amount of virtual machines is deployed both on EU and US based physical machines. These physical machines should be directly connected to the public Internet. On the VMs, a mock-up client is deployed. Gradually, more and more of the VMs activate this client,

Scenario Name	SCENARIO 3: Researching the concept of geographical elasticity in cloud computing

which automatically starts interacting with the mock-up web service.

3. As the load increases on the web service, so does the need to instantiate more resources. Once the adopted elasticity results in a specific upscale, all clients start measuring specific useful metrics such as throughput, delay and jitter of the incoming mock-up audio and video streams. Besides, mock-up requests to the interactive part of the service should regularly be triggered at the clients. In this case the same metrics as for the media streams should be measured.
4. This same experiment will be performed for horizontal, vertical and geographical elasticity. In the end, the goal of these three experiments is to demonstrate and quantify the added value of geographical elasticity compared to horizontal and vertical elasticity. To allow a fair and realistic comparison, the different client VMs are configured in such a way that the EU clients mainly call some mock-up service A, while the US clients mainly call some fake service B. Once in a while, a US client will also call service A. In case of geographical elasticity, the slave service will be deployed on a cloud computing platform in the US. The OpenFlow path between the two different service instances should be established automatically. To be able to dynamically influence this path to test different flow control algorithms, it should be automatically configured in such a way that it goes through an OpenFlow testbed.

The first step in the implementation of this experiment is to find the suitable resources. The researcher opens the resource discovery tool on the Fed4FIRE portal. He/she enters the following requirements: virtual machines running on a cloud computing environment (in the EU and in the USA) that supports both horizontal and vertical elasticity and which has a layer 2 connection to an OpenFlow testbed. It turns out that one of the BonFIRE islands is a suitable EU candidate. In the USA, a similar testbed is found that has a layer 2 connection to the same OFELIA OpenFlow testbed as the previously discovered BonFIRE island. Now that the cloud computing platforms for both services instances have been determined, together with the required OpenFlow infrastructure, the only resource left to discover are the client VMs in the EU and in the USA. When looking for VMs that are directly connected to the public Internet, and are deployed on physical machines located in the EU and in the USA, a large list of PlanetLab Europe and PlanetLab Central nodes is returned.

The researcher then uses the Fed4FIRE portal to requests 1 VM on the BonFIRE island, 1 VM on the USA OpenFlow testbed, 50 VMs on PlanetLab Europe and 50 VMs on PlanetLab Central. In this request it is mentioned that they should at least remain up and running for 30 days. On the Fed4FIRE portal, the researcher

Scenario Name	SCENARIO 3: Researching the concept of geographical elasticity in cloud computing

can also open a tool on which the topology of the applied OFELIA testbed is displayed. This allows the researcher to know what optical network resources are available, what topologies he/she can configure, and what capacities (bandwidth) can be utilized for the experiments. The researcher is also interested to know what compute nodes (VMs) could be interconnected via the network to provide link impairment. Based on this knowledge, the experimenter defines an appropriate flow space for the experiment, containing several links between the available OpenFlow switches, a virtual machine on which the flow controller (NOX server) will be deployed, and some additional VMs that will be used as impairment nodes. Note that throughout this entire procedure, no additional logins or passwords were requested, although resources have been requested on different testbeds. This effectively handles the first hurdle that FIRE experimenters are often faced with when testing new facilities: the challenge of getting an activated account on these facilities.

The next step is then to configure the entire experiment and run it. One approach could be to manually log in to all VMs, and run a small script that installs the required libraries, services, clients and the public SSH key of the experimenter. To perform the actual experiment, a bash script could then be written that runs on one of the VMs, and which sequentially executes all required commands to start the mock-up web services, gradually increase the load on them, and dynamically change characteristics of the OpenFlow path on the Ofelia site. However, if the federation would provide some common mechanism for experiment control, a much more efficient approach could be followed. In that case, it should be described only once how the client nodes should be configured. When feeding this description to the experiment controller, together with a list of IP addresses corresponding with all requested VMs on PlanetLab, the tool can automatically perform the installation of all 100 nodes. The same controller could also steer the execution of the experiment according to the predefined scenario. The approach with the bash script followed a sequential approach of going through a fixed list of commands with appropriate sleeping times between each command. A common experiment control tool could also trigger events based on measured metrics instead of purely based on time. Besides, such a control tool could also be able to send commands instantly to a group of nodes, allowing synchronized activation of certain clients or services.

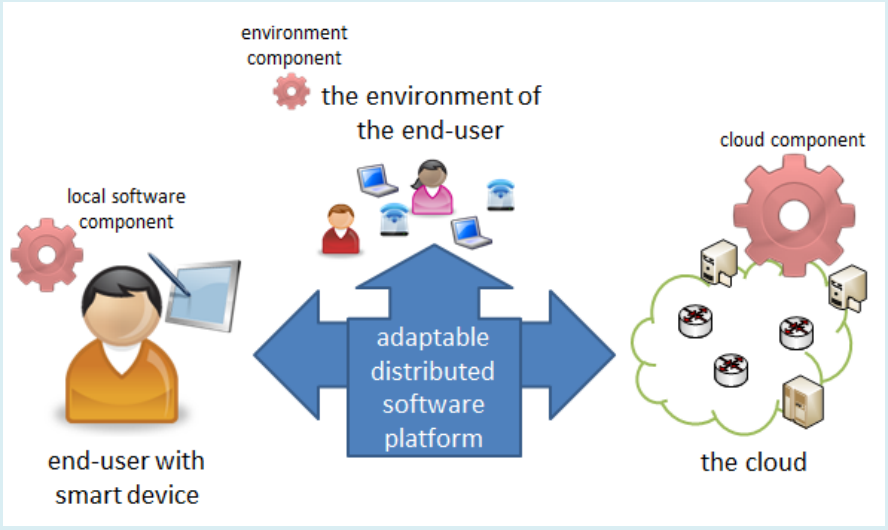
In the following it is assumed that such an experiment control tool is available. First of all, during the execution of the experiment, the researcher continuously wants to be informed about the status of all VMs. If too many clients would go down for some reason, it might be needed to postpone the experiment, or

Scenario Name	SCENARIO 3: Researching the concept of geographical elasticity in cloud computing
	configure it with other resources. As mentioned before, specific metrics will also have to be recorded on all 100 client nodes. It would be very convenient for the experimenter if some common measuring tool would be available on all nodes in the experiment. This tool should make sure that when a client wants to store some specific measured metric, that it automatically gets pushed to a central database or repository for later analysis. On this facility for permanent storage, the link between any dataset and the corresponding experiment run ID should be clearly identifiable. For each run ID, background information (execution start and end time, experimenter comments, etc.) should also be kept. Together, the datasets and experiment information provide all necessary information to perform scientifically correct performance analysis. Besides, the platform for permanent storage should also allow to save experiment descriptors, experiment control scenarios, software to be tested, and so on. This way experiments can easily be repeated later.
Technology involved	Virtual Machines, Cloud computing platforms, OpenFlow

Scenario Name	**SCENARIO 4: Benchmarking a service platform for information retrieval**
Background / Rationale	The scenario described here involves information retrieval from a document archive. An example archive could be that of a large newspaper, containing all articles ever released by this paper. This means that 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. This document processing would be based on text analysis and training documents, not on matches with existing lists of words. This means that in this case, processing of a document is a very calculation intensive task. Most likely such an information retrieval platform would therefore be a distributed solution. For instance in case of the example newspaper archive, it would contain a great amount of data which could only be processed in parallel in order to finish the task in an acceptable time frame (e.g., 10 million documents that could be processed by 10 machines during the period of one week). At the end of a national research project focusing on such an information retrieval service, it was decided to establish a spin-off that would further develop the existing proof-of-concept implementation, benchmark it thoroughly, and bring it to the market. The intention is to use the Fed4FIRE facilities for the benchmarking of the developed service platform. The researchers of the spin-off are interested in the following questions: • Does our solution scale? Does it still operate adequately when running in a highly distributed fashion? • How can it cope with deployments on heterogeneous environments with different types of machines and virtualization?
Scenario description (Storyboard)	To test the service, the researchers create an experiment with 1 storage server and some worker nodes which fetch the data from the storage server, process it, and send it back. On the machines (on all worker nodes) java version 7 needs to be installed, together with the experimenter's own software and dictionaries (about two gigabytes in size). The experimenter already has an account on the iMinds Virtual Wall (an Emulab based testbed) for some years now. He/she copied the software and dictionaries to his/her NFS exported home directory. With the help of a script the user deploys the software and the dictionaries automatically on each worker node after the experiment has started. For testing new information retrieval techniques, it is needed to rerun and analyze all the documents several times. Therefore the researchers need a lot of machines. As mentioned, they also want to see the influence of different hardware to optimize the parallelization techniques (faster/slower cpus, more/less cores, etc.). For this, the researchers want to run this experiment also

Scenario Name	SCENARIO 4: Benchmarking a service platform for information retrieval
	on Grid'5000. This is a FIRE facility belonging to the Fed4FIRE federation, and provides multiple hardware architectures on resources which are very similar to the Virtual Wall. When testing if the developed solution still operates adequately when running in a highly distributed fashion, the researchers are planning an experiment where worker nodes collaborating in a single experiment would be provisioned both on the Virtual Wall and on Grid'5000 at the same time. It should be easy for the spin-off researchers to switch between the Virtual Wall (which they already used a long time in their previous career at the university), and Grid'5000. If there would be a high corresponding learning curve when working with a previously unknown testbed, the researchers would lose valuable time. This would be unacceptable in this case, since the spin-off only has a small period of time available before they run out of venture capital. The Fed4FIRE federation framework should therefore pay great attention to common authentication and authorization functionalities, and to common tools for resource discovery, reservation, experiment control and monitoring. It should also be possible for the researcher to share his/her centralized and always available home directory which contains the software and dictionaries needed for the experiment. This way, the experimenter would not have to upload all the gigabytes to the testbed each time he/she starts a new experiment. This would be a non-practical approach, especially in case of a poor Internet uplink (e.g., if the experimenter would want start some experiments from a trade fair hotel room).
Technology involved	Linux servers, Emulab, OAR, Java, Artificial Intelligence

Scenario Name	SCENARIO 5: Mobile cloud service platform
Background / Rationale	During the past two decades, technological advances in the chip manufacturing industry have opened the door to a new generation of low-power portable embedded computing devices: laptops, tablets and smartphones are omnipresent in the everyday private and professional lives of many people. At the same time –and driven by the increasing mobility expectations of the users of these devices- the evolutions in wireless techniques and technologies now make it possible to be connected to the Internet (nearly) anytime and anywhere in large parts of the world, be it through a Wi-Fi connection or via a cellular network. Moreover, the Internet of today is capable of transporting massive amounts of information between servers anywhere in the world. Interestingly, these evolutions cause the "old" concept of "network terminals" to regain an increased attention of the research community. While today's portable end-user devices can indeed be impressively powerful (e.g., compared to the personal desktop computers of just ten years ago), their performance and available memory obviously can in no way be compared to the information and processing power that is available in the cloud today. Moreover, the battery in these devices is still a weak point: intensive computing tasks and sending/receiving large amounts of data quickly drain the battery, forcing the end-user to give up his/her mobility while being connected to a power socket. As such, depending on the type of task that needs to be executed on such portable end-user device, it might be more efficient (in terms of service quality, energy efficiency, lifetime of the end-user device, etc.) to execute simple computing tasks such as editing a text file locally, while offloading CPU intensive tasks such as graphics rendering to a remote server located in the cloud. Additionally, it is clear that the portable computing devices of today and in the future are not only used to carry out tasks locally or on a remote server. Many different interactions with the environment are possible; For example: the device (in whatever shape or form) can be used as part of a localization service, may connect to a large screen to display information in a more convenient way, can interact with other similar devices of people in the neighbourhood, can be used to make payments or open electronically locked doors. For all of these cases and for all of these tasks, an adaptable distributed software platform can be imagined that determines where and how tasks are run by different interacting software components that run either (partly) on the device, and/or (partly) in the network, and/or (partly) on other neighbouring devices. To obtain the maximum user satisfaction, the specific workload split between these different software components should be influenced by several criteria such as the available compute/storage/power resources of the device and its environment, the wireless connection technique and quality, the specific application requirements (e.g., "result is needed asap vs. result can wait for some time"), wireless interference, and many others.

Scenario Name	SCENARIO 5: Mobile cloud service platform
Picture	
Scenario description (Storyboard)	A developer of an adaptable distributed software platform, as described in the rationale above, wants to test the performance of his platform on a testbed. In a first step, the developer needs to know whether executing an experiment involving this large amount of different components is possible. For that, he or she first needs to discover what infrastructures are available. In this case, the scenario is quite complex, and the developer needs to find out what 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. As many components are involved, this is not such an easy task. The developer might go through a large amount of documentation on the different facilities in the federation, but it is clear that information on the individual facilities will not be enough; If, for example, the experiment is to include different wireless access techniques (including licensed LTE and/or WiMax) as well as deployments on cloud infrastructures, it might be possible for the developers to estimate the feasibility of running a small part of the experiment on a single facility in the federation. However, the possibility of real-time interconnection and cooperation between facilities will have to be clearly described as well. Additionally, the developer may want to contact people from different facilities or someone with a more high-level view to sort out some specific questions. Based on this first discovery phase, the developer decides to build his solution on top of several facilities; In the experiment, two “main” end-users will be involved. Main end-user 1 will be located in the w-iLab.t testbed in Zwijnaarde. Here, the end-user device is mimicked by an embedded PC which is mounted on a mobile robot. In this way, repeatable mobility will be introduced in the experiment.

Scenario Name	SCENARIO 5: Mobile cloud service platform
	Other wireless devices inside w-iLab.t are configured as Wi-Fi access points, to which end-user 1 will connect. Still other devices will be configured so they represent the environment of the end-user: some of these "environment" devices will do nothing more than causing intermittent interference which should impact the Wi-Fi connectivity and thus the distribution of software components. Other "environment" devices will be configured similarly to the "main" end-users and represent other users in the environment, with whom end-user 1 can directly interact over Wi-Fi. The developer thus needs to have a good view on the location of the different nodes (which are configured as Wi-Fi access points, interferers or end-users in the environment). End-user 2 will be located in the NITOS testbed, and will at times connect via Wi-Fi and at times via LTE. Again the details on the node locations are important. This is also true for the LTE part of the experiment: a clear view on the topology between the LTE base station and the connected nodes is needed, preferably through an easy-to-understand GUI. The Bonfire infrastructure is identified as the cloud infrastructure, which will host the cloud components of the distributed software. To reduce the complexity, in a first version of the experiment, only the Virtual Wall (part of Bonfire) will be used during the experiment), and only the ecosystem (end-user, environment, cloud) around end-user 1 is considered. Different servers representing servers in the cloud are configured on the Virtual Wall. The links between these "servers in the cloud" are also emulated by the Virtual Wall, as such mimicking server locations very close to the end-user and distant servers, located on another continent. The developer then sets up a similar experiment inside the NITOS testbed to test the impact of a different wireless environment and the impact of LTE on his solution. Since the Virtual Wall is also part of this experiment, it must be clear to the developer on how to interconnect the Virtual Wall an NITOS, which are –in contrast to the w-iLab.t and Virtual Wall- on different networks. Obviously, the more uniform the w-iLab.t and NITOS descriptions on the portal and the needed resource discovery and reservation tools are, the easier for the developer. In all of the basic experiments, the developer also wants to define measurements (metrics) that can be compared in a meaningful way between the two different locations. In a final set of experiments, each of the main end-users will not only be interacting with its environment, there will also be a more advanced experiment in which an collaborative meeting is set up between the two main end-users over the Internet. During this meeting, files are shared over the Internet, and an end-to-end video stream is set up. In this phase, the developer also changes its cloud servers: there will not longer only be cloud servers with emulated links in the Virtual Wall: different servers in the Bonfire cloud will be used.

Scenario Name	SCENARIO 5: Mobile cloud service platform
	Before, during and after the experiment, the developer would like to use as uniform tools as possible to plan, schedule and execute the experiments, and to collect the data.
Technology involved	Physical PC/server, Virtual Machines, Wi-Fi access, cellular access, wireless sensor network, Cloud infrastructure

3 Requirements

Source	Req. id	Req. Statement	Req. description	Priority	Comments

- Source: Use case that has provided the requirement.
  - 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
  - GEN: When the requirement is shared among the majority of the 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 first development cycle of Fed4FIRE
  - Medium: Should be implemented in later iterations
  - Low: nice to have but not essential.
- Comments: additional information regarding the requirement

3.1 Experiment workflow and lifecycle management

3.1.1 Resource discovery

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
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	High	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).
WIRELESS MOBILE-CL	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.)	High	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.
WIRELESS MOBILE-CL	I.1.103	Wireless spectrum as a resource	The different available wireless channels should be considered to be an infrastructure resource.	High	If channels are a resource, they can easily become discovered and reserved later on.
GEO-ELAS	I.1.104	Site location information	For all nodes the location of the site where they are physically deployed should be known.	High	Per site the location might be the same value for all nodes, no need for the accuracy level of I.1.002 here.
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	High	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

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
			infrastructures and/or existing uniform tools. This would decrease the development costs for the infrastructure providers and tool builders.		be well documented.
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.	High	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.
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.	High	
GEO-ELAS MOBILE-CL	I.1.108	Background info virtualized resources	Resource virtualization (VMs, flows) must provide information about the supporting physical devices and their location	High	

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
GEN	I.1.109	Query search	It should be able to build tools that allow a query search for suitable infrastructures/nodes	High	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.
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 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.	High	

3.1.2 Resource requirements

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
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).	High	

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
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 PlanetLab Europe sites in the Benelux.), and let the reservation /provisioning tools select a suitable node at experiment runtime, based on availability.	Medium	This is more convenient for experimenters to make sure that they do not need to select free node IDs for specific reservation times. However, this functionality is not a prerequisite to be able to actually run experiments, so it gets the medium priority.
TEACHING GEO-ELAS	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	High	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.
WIRELESS	I.1.204	Relation between different required resource types	The experimenter should be able to specify the need for additional experimentation devices (such as spectrum analyzers or Smartbits measurement devices).	Medium	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.

3.1.3 Resource reservation

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
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).	High	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.
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.	Medium	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.
WIRELESS BENCHM	I.1.303	Secure reservation	Fed4FIRE must provide a reservation system with adequate security to provide assurance to industrial users	High	

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
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.	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.	High	

3.1.4 Resource provisioning

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
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.	High	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.
GEN	I.1.402	Customizing Linux	Fed4Fire must provide the ability to install a specific custom Linux kernel or distribution on the nodes	Medium	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.
GEN	I.1.403	Root access	Fed4FIRE must provide the possibility to access a node as root user	High	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

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
					experimentation needs. This also requires root access.
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.	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.	Medium	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. Therefore the priority is set to medium. 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.
GEO-ELAS MOBILE-CL	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.	High	

3.1.5 Experiment control

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
GEN	I.1.501	SSH access	Nodes must be accessible via SSH.	High	This is most valuable during the development phase or for debugging purposes.
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.	High	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.
GEO-ELAS MOBILE-CL	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.	Medium	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 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, etc.	High	This is a high priority requirement in Fed4FIRE, since the intention is to federate a very heterogeneous collection of infrastructures, services and applications.
GEN	I.1.505	Ease of use	The experimenter should be able to describe the whole experiment in a human readable and uniform way.	Medium	More a convenience than a necessity. Hence the medium priority.

3.2 Measurement and monitoring

3.2.1 Monitoring

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
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.	High	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.
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.)	Medium	Users could also manually configure their experiments to measure these metrics. Hence this requirement is more related to convenience, and receives the medium priority.
WIRELESS MOBILE-CL	I.2.103	Wireless interference	Information about external wireless interference during the execution of the experiment should be provided.	High	The interference can be detected using monitor interfaces in dedicated nodes and/or spectrum analyzers that offer more exact results. This functionality should be easily provided to every experimenter who is not “spectrum analyzing” expert.
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	High	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.

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
			actually trying to use these resources at that moment, they should also be informed.		
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.	High	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.
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.	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	Medium	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. Hence the medium priority.
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 experimenters should be informed about this.	Medium	If the measurement is not available, the returned zero or random values will most likely be noticeable by the experimenter. However, a formal notification of missing measurements (e.g., because a given metric is not applicable in all domains) is more convenient. Hence the medium priority.

3.2.2 Permanent storage

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
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	High	
GEN	I.2.202	Data security	Access to the data should be properly secured	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.	High	Experiment configurations can contain deployment descriptors, experiment control scripts, etc.
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.	Low	This could be of interest for the research community as a whole, but is considered not that important for the experimenter that performed the actual initial experiments. Therefore this requirement is given a low priority.
GEN	I.2.205	Stored metadata	Fed4FIRE should give the possibility to store metadata about the data of the experiments.	Low	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. Hence it receives the same low priority.
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	Medium	This requirement is not a prerequisite for Fed4FIRE operation, but it will increase reliability on the long run. Hence it is considered as medium priority.

3.3 Trustworthiness

3.3.1 Dynamic federated identity management

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
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 authentication framework would be compatible with those already supported by some of the infrastructures. This would decrease the development costs for the infrastructure providers.	High	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. Regarding the compatibility with existing solutions, this of course first of all has to be technically feasible.
GEN	I.3.102	Public keys	Fed4FIRE should also provide authentication by the use of public SSH keys.	High	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.
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.	High	
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	High	

3.3.2 Authorization

Source scenario	Req. Id	Req. statement	Req. description	Priority	Comments
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.	High	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. An example tool to implement such behaviour is to dynamically set IPTables rules to enable/disable resource access. This is however not such a scalable solution.
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.	Medium	

3.3.3 SLA management

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
WIRELESS BENCHMARK	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.	Medium	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. Therefore this requirement gets a medium priority. In the case of academic research, the demand for SLAs is rather low.

3.3.4 Trust and user experience

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
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.	Medium	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. Anyway, both the monitoring and the feedback approaches would need specific functionality to be in place in the Fed4FIRE federation. In all use cases the experimenter was looking for previously unknown facilities to implement the

Source scenario	Req. id	Req. statement	Req. description	Priority	Comments
					experiment. Reliability reports as mentioned here could be a valuable additional source of information. But it is not a key functionality that is expected of Fed4FIRE. Therefore it is given a medium priority.
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.	High	Funding and sustainability is a key issue for all infrastructures. Therefore this is a high-priority requirement.
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.	High	FIRE facilities can be powerful tools, and misuse should most definitely be handled adequately. Therefore this is a high priority requirement.

3.4 Interconnectivity

Source scenario	Req. Id	Req. statement	Req. description	Priority	Comments
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.	High	Ideally all infrastructures are connected to high-capacity research Internet backbones such as Géant.
GEO-ELAS MOBILE-CL	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.	High	
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.	High	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.
GEO-ELAS BENCHM	I.4.004	Per-slice bandwidth reservation	Per experiment, Fed4FIRE should provide the possibility to reserve bandwidth on the links that interconnect specific infrastructures.	Medium	It is expected that non-reserved interconnections over high-capacity research backbones will be sufficient in most cases. Therefore this is not a high priority requirement.
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.	High	Many infrastructures offer a large number of virtual or physical resources at one site. Their available IPv4 range is not sufficient to provide a public IPv4 address to all these resources. Hence NAT mechanisms are typically put in place. But this conflicts with the

Source scenario	Req. Id	Req. statement	Req. description	Priority	Comments
					transparency requirement. IPv6 support would be the solution for this problem.
GEN	I.4.006	Information about testbed Interconnections	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.	High	

4 Requirements matrix

In this section the most important requirements for the first release cycle of Fed4FIRE are enumerated. It gathers all requirements which cover the majority of the use cases (annotated with "GEN" in the source scenario column) AND have a high priority. These requirements are considered essential and their implementation should be included in the first cycle developments as much as possible and as long as this is feasible in terms of other constraints (e.g. effort).

Federation aspect	Req. id	Req. statement
Resource discovery	I.1.101	Node capabilities
Resource discovery	I.1.105	Discovery through federation-wide APIs
Resource discovery	I.1.106	Intra-infrastructure topology information
Resource discovery	I.1.107	Inter-infrastructure topology information
Resource discovery	I.1.109	Query search
Resource discovery	I.1.110	Catalogue search
Resource requirements	I.1.201	Manually extract requirements from discovery query results
Resource reservation	I.1.301	Hard resource reservation
Resource reservation	I.1.304	Automated reservations handling
Resource reservation	I.1.305	Reservation information
Resource provisioning	I.1.401	Provisioning API
Resource provisioning	I.1.403	Root access
Resource provisioning	I.1.404	Internet access to software package repositories
Experiment control	I.1.501	SSH access
Experiment control	I.1.502	Scripted control engine
Experiment control	I.1.504	Generality of control engine
Monitoring	I.2.101	Measurement support framework
Monitoring	I.2.104	Monitoring resources for operational support
Monitoring	I.2.105	Monitoring resources for suitable resource selection and measurement interpretation

Federation aspect	Req. id	Req. statement
Monitoring	I.2.106	Minimal impact of monitoring and measuring tools
Permanent storage	I.2.201	Data storage
Permanent storage	I.2.202	Data security
Permanent storage	I.2.203	Stored experiment configuration
Dynamic federated identity management	I.3.101	Single account
Dynamic federated identity management	I.3.102	Public keys
Dynamic federated identity management	I.3.103	OpenVPN handling
Dynamic federated identity management	I.3.104	Authentication of API calls
Authorization	I.3.201	Per-experimenter restrictions
Trust and user experience	I.3.402	Experiment descriptions
Trust and user experience	I.3.403	Accountability
Interconnectivity	I.4.001	Layer 3 connectivity between testbeds
Interconnectivity	I.4.003	Transparency
Interconnectivity	I.4.005	IPv6 support
Interconnectivity	I.4.006	Information about testbed Interconnections

Appendix A: Overview of experiment lifecycle management

Function		Description
Resource discovery (1)		Discovery of the facility resources (e.g., uniform resource description model).
Resource requirements (2)		Specification of the resources required during the experiment, including compute, network, storage and software libraries. E.g., 5 compute nodes, 100Mbps network links, specific network topology, 1 TB storage node, 1 IMS server, 5 measurement agents.
Resource reservation (3)		How can you reserve the resources? Examples: (1) no hard reservation or best-effort (use of a calendar that is loosely linked to the facility), (2) hard reservation (once reserved, you have guaranteed resource availability). Other options: (1) one should reserve sufficient time in advance or (2) one can do instant reservations
Resource provisioning (4)	Direct (API)	Instantiation of specific resources directly through the facility API (responsibility of the experimenter to select individual resources).
	Orchestrated	Instantiation of resources through a functional component, orchestrating resource provisioning (e.g., OpenNebula or PII orchestration engine) to decide which resources fit best with the experimenter's requirements. E.g., the experimenter requests 10 dual-core machines with video screens and 5 temperature sensors.
Experiment control (5)		Control of the experimentation facility resources and experimenter scripts during experiment execution (e.g., OMF experiment controller, ssh). This could be predefined interactions and commands to be executed on resources (events at startup or during experiment workflow). Examples are: startup or shutdown of compute nodes, change in wireless transmission frequency, instantiation of software components during the experiment and breaking a link at a certain time in the experiment. Real-time interactions that depend on unpredictable events during the execution of the experiment are also considered.
Monitoring (6)	Resources	Monitoring of the infrastructure health and resource usage (e.g., CPU, RAM, network, experiment status and availability of nodes).
	Experiment	Monitoring of user-defined experimentation metrics (e.g., service metrics, application parameters) of the solution under test. Examples include: number of simultaneous download sessions, number of refused service requests, packet loss and spectrum analysis.
Permanent storage (data, experiment descriptor) (7)		Storage of the experiment descriptor or experimentation data beyond the experiment lifetime (e.g., disk images and NFS).

Appendix B: Overview of trustworthiness

Function	Description
Dynamic Federated Identity Management	Tools and services for authentication of individuals and programmes including the processes for establishing roots of trust and issuing identities within a federation.
Authorization	Capabilities to allow access to data and other resources only to authorized individuals, including mechanisms for delegation and revocation of rights to experimenters and services operated within an organization or by 3rd party organizations. Mechanisms to deal with how user attributes (e.g., name, email, id, and roles) can be incorporated into policy enforcement and decision making using approaches such as role-based, attribute-based and process-based access control, along with more advanced techniques such as delegation logic.
SLA Management	Specifications, tools and services for SLAs that can support the formal definition of the relationship between providers and customers as a mechanism to increase trust in providers by encoding security and dependability commitments and ensuring the level of Quality of Service is maintained to an acceptable level.
Trust and User Experience	Mechanisms and tools for building trustworthy services based on the combination of reputation and monitoring data that reflects the experimenters' experience of service and empowers them with a "smart" service that provides a unified and quantitative view of the trustworthiness of a facility.

Appendix C: Questionnaire as distributed to the WP3 partners

Platform

Platform type: (If this is a type of platform for which there exists multiple instances, state the type, such as PlanetLab or Emulab. If there is only one instance, give the platform name, such as FEDERICA.)
Platform developers: (Which institutions contribute to the development of this type of platform?)
Instances in Fed4FIRE: (What are the instances of this type of platform that are present in Fed4FIRE? Mention the project partners who are responsible.)
Instances elsewhere in Europe: (List the known instances of this type of platform that are not in Fed4FIRE but that are present in Europe. Mention the institutions that are responsible.)
Instances elsewhere in the world: (List the known instances of this type of platform that are present elsewhere in the world. Mention the institutions that are responsible.)

1. Resource discovery

Functionality: Discovery of the facility resources (e.g., uniform resource description model).
Character: SPECIFIC/GENERIC (Keep only one word: SPECIFIC, meaning that this functionality is available in a way that is specific only to the individual platform, or GENERIC, meaning that this functionality is available in a generic way, that crosses platforms.) Explanatory text. (Describe.)
Priority: LOW/MED/HIGH (Keep only one word: LOW, MED, or HIGH, describing the priority for further development of this functionality.) Explanatory text. (Describe.)
Status: MISSING/TO IMPROVE/WORKING (Keep only one phrase: MISSING, TO IMPROVE, or WORKING, describing the status of this functionality.) Explanatory text. (Describe.)
Example experiments: Description of experiment. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
User needs: With the experiment in mind, describe future developments that are required, from a user's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
Provider needs: With the experiment in mind, describe future developments that are required, from the provider's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)

2. Resource requirements

Functionality: Specification of the resources required during the experiment, including compute, network, storage and software libraries. E.g., 5 compute nodes, 100Mbps network links, specific network topology, 1 TB storage node, 1 IMS server, 5 measurement agents.
Character: SPECIFIC/GENERIC (Keep only one word: SPECIFIC, meaning that this functionality is available in a way that is specific only to the individual platform, or GENERIC, meaning that this functionality is available in a generic way, that crosses platforms.) Explanatory text. (Describe.)
Priority: LOW/MED/HIGH (Keep only one word: LOW, MED, or HIGH, describing the priority for further development of this functionality.) Explanatory text. (Describe.)
Status: MISSING/TO IMPROVE/WORKING (Keep only one phrase: MISSING, TO IMPROVE, or WORKING, describing the status of this functionality.) Explanatory text. (Describe.)
Example experiments: Description of experiment. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
User needs: With the experiment in mind, describe future developments that are required, from a user's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
Provider needs: With the experiment in mind, describe future developments that are required, from the provider's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)

3. Resource reservation

Functionality: How can you reserve the resources? Examples: (1) no hard reservation or best- effort (use of a calendar that is loosely linked to the facility), (2) hard reservation (once reserved, you have guaranteed resource availability). Other options: (1) one should reserve sufficient time in advance or (2) one can do instant reservations.
Character: SPECIFIC/GENERIC (Keep only one word: SPECIFIC, meaning that this functionality is available in a way that is specific only to the individual platform, or GENERIC, meaning that this functionality is available in a generic way, that crosses platforms.) Explanatory text. (Describe.)
Priority: LOW/MED/HIGH (Keep only one word: LOW, MED, or HIGH, describing the priority for further development of this functionality.) Explanatory text. (Describe.)
Status: MISSING/TO IMPROVE/WORKING (Keep only one phrase: MISSING, TO IMPROVE, or WORKING, describing the status of this functionality.) Explanatory text. (Describe.)
Example experiments: Description of experiment. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
User needs: With the experiment in mind, describe future developments that are required, from a user's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
Provider needs: With the experiment in mind, describe future developments that are required, from the provider's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)

4. Resource provisioning

Functionality: Direct (API): Instantiation of specific resources directly through the facility API (responsibility of the experimenter to select individual resources). Orchestrated: Instantiation of resources through a functional component, orchestrating resource provisioning (e.g., OpenNebula or PII orchestration engine) to decide which resources fit best with the experimenter's requirements. E.g., the experimenter requests 10 dual-core machines with video screens and 5 temperature sensors.
Character: SPECIFIC/GENERIC (Keep only one word: SPECIFIC, meaning that this functionality is available in a way that is specific only to the individual platform, or GENERIC, meaning that this functionality is available in a generic way, that crosses platforms.) Explanatory text. (Describe.)
Priority: LOW/MED/HIGH (Keep only one word: LOW, MED, or HIGH, describing the priority for further development of this functionality.) Explanatory text. (Describe.)
Status: MISSING/TO IMPROVE/WORKING (Keep only one phrase: MISSING, TO IMPROVE, or WORKING, describing the status of this functionality.) Explanatory text. (Describe.)
Example experiments: Description of experiment. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
User needs: With the experiment in mind, describe future developments that are required, from a user's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
Provider needs: With the experiment in mind, describe future developments that are required, from the provider's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)

5. Experiment control

Functionality: Control of the experimentation facility resources and experimenter scripts during experiment execution (e.g., OMF experiment controller, ssh). This could be predefined interactions and commands to be executed on resources (events at startup or during experiment workflow). Examples are: startup or shutdown of compute nodes, change in wireless transmission frequency, instantiation of software components during the experiment and breaking a link at a certain time in the experiment. Real-time interactions that depend on unpredictable events during the execution of the experiment are also considered.
Character: SPECIFIC/GENERIC (Keep only one word: SPECIFIC, meaning that this functionality is available in a way that is specific only to the individual platform, or GENERIC, meaning that this functionality is available in a generic way, that crosses platforms.) Explanatory text. (Describe.)
Priority: LOW/MED/HIGH (Keep only one word: LOW, MED, or HIGH, describing the priority for further development of this functionality.) Explanatory text. (Describe.)
Status: MISSING/TO IMPROVE/WORKING (Keep only one phrase: MISSING, TO IMPROVE, or WORKING, describing the status of this functionality.) Explanatory text. (Describe.)
Example experiments: Description of experiment. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
User needs: With the experiment in mind, describe future developments that are required, from a user's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
Provider needs: With the experiment in mind, describe future developments that are required, from the provider's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)

6. Monitoring

Functionality: Resources: Monitoring of the infrastructure health and resource usage (e.g., CPU, RAM, network, experiment status and availability of nodes). Experiment: Monitoring of user-defined experimentation metrics (e.g., service metrics, application parameters) of the solution under test. Examples include: number of simultaneous download sessions, number of refused service requests, packet loss and spectrum analysis.
Character: SPECIFIC/GENERIC (Keep only one word: SPECIFIC, meaning that this functionality is available in a way that is specific only to the individual platform, or GENERIC, meaning that this functionality is available in a generic way, that crosses platforms.) Explanatory text. (Describe.)
Priority: LOW/MED/HIGH (Keep only one word: LOW, MED, or HIGH, describing the priority for further development of this functionality.) Explanatory text. (Describe.)
Status: MISSING/TO IMPROVE/WORKING (Keep only one phrase: MISSING, TO IMPROVE, or WORKING, describing the status of this functionality.) Explanatory text. (Describe.)
Example experiments: Description of experiment. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
User needs: With the experiment in mind, describe future developments that are required, from a user's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
Provider needs: With the experiment in mind, describe future developments that are required, from the provider's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)

7. Permanent storage (data, experiment descriptor)

Functionality: Storage of the experiment descriptor or experimentation data beyond the experiment lifetime (e.g., disk images and NFS).
Character: SPECIFIC/GENERIC (Keep only one word: SPECIFIC, meaning that this functionality is available in a way that is specific only to the individual platform, or GENERIC, meaning that this functionality is available in a generic way, that crosses platforms.) Explanatory text. (Describe.)
Priority: LOW/MED/HIGH (Keep only one word: LOW, MED, or HIGH, describing the priority for further development of this functionality.) Explanatory text. (Describe.)
Status: MISSING/TO IMPROVE/WORKING (Keep only one phrase: MISSING, TO IMPROVE, or WORKING, describing the status of this functionality.) Explanatory text. (Describe.)
Example experiments: Description of experiment. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
User needs: With the experiment in mind, describe future developments that are required, from a user's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
Provider needs: With the experiment in mind, describe future developments that are required, from the provider's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)

8. Dynamic federated identity management

Functionality: Tools and services for authentication of individuals and programmes including the processes for establishing roots of trust and issuing identities within a federation.
Character: SPECIFIC/GENERIC (Keep only one word: SPECIFIC, meaning that this functionality is available in a way that is specific only to the individual platform, or GENERIC, meaning that this functionality is available in a generic way, that crosses platforms.) Explanatory text. (Describe.)
Priority: LOW/MED/HIGH (Keep only one word: LOW, MED, or HIGH, describing the priority for further development of this functionality.) Explanatory text. (Describe.)
Status: MISSING/TO IMPROVE/WORKING (Keep only one phrase: MISSING, TO IMPROVE, or WORKING, describing the status of this functionality.) Explanatory text. (Describe.)
Example experiments: Description of experiment. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
User needs: With the experiment in mind, describe future developments that are required, from a user's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
Provider needs: With the experiment in mind, describe future developments that are required, from the provider's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)

9. Authorization

Functionality: Capabilities to allow access to data and other resources only to authorised individuals, including mechanisms for delegation and revocation of rights to experimenters and services operated within an organisation or by 3rd party organisations. Mechanisms to deal with how user attributes (e.g., name, email, id, and roles) can be incorporated into policy enforcement and decision making using approaches such as role-based, attribute-based and process-based access control, along with more advanced techniques such as delegation logic.
Character: SPECIFIC/GENERIC (Keep only one word: SPECIFIC, meaning that this functionality is available in a way that is specific only to the individual platform, or GENERIC, meaning that this functionality is available in a generic way, that crosses platforms.) Explanatory text. (Describe.)
Priority: LOW/MED/HIGH (Keep only one word: LOW, MED, or HIGH, describing the priority for further development of this functionality.) Explanatory text. (Describe.)
Status: MISSING/TO IMPROVE/WORKING (Keep only one phrase: MISSING, TO IMPROVE, or WORKING, describing the status of this functionality.) Explanatory text. (Describe.)
Example experiments: Description of experiment. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
User needs: With the experiment in mind, describe future developments that are required, from a user's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
Provider needs: With the experiment in mind, describe future developments that are required, from the provider's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)

10. SLA Management

Functionality: Specifications, tools and services for SLAs that can support the formal definition of the relationship between providers and customers as a mechanism to increase trust in providers by encoding security and dependability commitments and ensuring the level of Quality of Service is maintained to an acceptable level.
Character: SPECIFIC/GENERIC (Keep only one word: SPECIFIC, meaning that this functionality is available in a way that is specific only to the individual platform, or GENERIC, meaning that this functionality is available in a generic way, that crosses platforms.) Explanatory text. (Describe.)
Priority: LOW/MED/HIGH (Keep only one word: LOW, MED, or HIGH, describing the priority for further development of this functionality.) Explanatory text. (Describe.)
Status: MISSING/TO IMPROVE/WORKING (Keep only one phrase: MISSING, TO IMPROVE, or WORKING, describing the status of this functionality.) Explanatory text. (Describe.)
Example experiments: Description of experiment. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
User needs: With the experiment in mind, describe future developments that are required, from a user's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
Provider needs: With the experiment in mind, describe future developments that are required, from the provider's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)

11. Trust and User Experience

Functionality: Mechanisms and tools for building trustworthy services based on the combination of reputation and monitoring data that reflects the experimenters' experience of service and empowers them with a "smart" service that provides a unified and quantitative view of the trustworthiness of a facility.
Character: SPECIFIC/GENERIC (Keep only one word: SPECIFIC, meaning that this functionality is available in a way that is specific only to the individual platform, or GENERIC, meaning that this functionality is available in a generic way, that crosses platforms.) Explanatory text. (Describe.)
Priority: LOW/MED/HIGH (Keep only one word: LOW, MED, or HIGH, describing the priority for further development of this functionality.) Explanatory text. (Describe.)
Status: MISSING/TO IMPROVE/WORKING (Keep only one phrase: MISSING, TO IMPROVE, or WORKING, describing the status of this functionality.) Explanatory text. (Describe.)
Example experiments: Description of experiment. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
User needs: With the experiment in mind, describe future developments that are required, from a user's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
Provider needs: With the experiment in mind, describe future developments that are required, from the provider's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)

12. Interconnectivity

Functionality: Interconnectivity includes: (a) the ability to reserve bandwidth per-slice between instances of this platform or between this platform and other platforms; (b) the ability to communicate between this platform and others across the public Internet, without reserved bandwidth.
Character: SPECIFIC/GENERIC (Keep only one word: SPECIFIC, meaning that this functionality is available in a way that is specific only to the individual platform, or GENERIC, meaning that this functionality is available in a generic way, that crosses platforms.) Explanatory text. (Describe.)
Priority: LOW/MED/HIGH (Keep only one word: LOW, MED, or HIGH, describing the priority for further development of this functionality.) Explanatory text. (Describe.)
Status: MISSING/TO IMPROVE/WORKING (Keep only one phrase: MISSING, TO IMPROVE, or WORKING, describing the status of this functionality.) Explanatory text. (Describe.)
Example experiments: Description of experiment. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
User needs: With the experiment in mind, describe future developments that are required, from a user's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)
Provider needs: With the experiment in mind, describe future developments that are required, from the provider's point of view, in this functionality. (The text should explain, from an experimenter's point of view, the importance of this functionality in order to be able to carry out a specified experiment.)

Appendix D: Future Internet Research Facilities

The table below is an update of Appendix A of the Fed4FIRE description of work. It lists the infrastructure facilities (related to WP3) that Fed4FIRE's original partners bring to the Infrastructures community, along with examples of well-known facilities from outside Fed4FIRE. It is organized by the underlying facility technology (PlanetLab-based, optical, switching, emulation, wireless LAN, etc.) or the facility's focus (Internet measurements). If a given facility has several technologies, we list it more than once; for instance, w-iLab.t, offers both wireless LAN and sensing capabilities. For each technology, we designate a Fed4FIRE lead partner who is the principal partner responsible for reaching out to the larger community that shares that technology.

Facilities that share a common codebase with Fed4FIRE's Infrastructures facilities can benefit directly from the project, by upgrading their code to include its developments. Other facilities listed here are known to be participating in federation technologies that are coherent with the approach taken in Fed4FIRE (notably, those that use variants on the emerging SFA interface), and so they, too, are ready candidates for cooperation. The list is not exhaustive, and will continue to be revised as the project progresses.

Facilities in Fed4FIRE (Fed4FIRE lead partner)	**Similar facilities in the larger community (non-exhaustive list)**
PlanetLab-based facilities (INRIA) • PlanetLab Europe (PLE): based at UPMC and INRIA (France)	**In Europe** • EmanicsLab: based at U Zurich (Switzerland) • G-Lab: based at U Kaiserslautern (Germany) • Commercially-deployed PlanetLab facilities at Orange Group (France, Poland, and elsewhere) **Worldwide** • PlanetLab Central (PLC): based at Princeton U (USA) • PlanetLab Japan (PLJ): based at U Tokyo and NICT (Japan) • Private PlanetLab Korea (PPK): based at KAIST (Korea) • PlanetLab NZ+: based in New Zealand
Open network measurement infrastructures (UPMC) • TopHat: run over PlanetLab Europe and based at UPMC (France)	**In Europe** • ETOMIC: based at ELTE (Hungary) and UAM (Spain) • RIPE TTM: based at RIPE

	(Netherlands) **Worldwide** • PerfSONAR: created and run by the global NREN community • DIMES: based at U Tel Aviv (Israel) • Scriptroute: based at U Washington (United States)
Optical testbeds (UNIVBRIS) • Experimenta: i2CAT (Spain) • ADVA ROADMs: UNIVBRIS (UK) • Distributed dark fibre in the UK: UNIVBRIS (UK)	**In Europe** • Cross Border Dark Fibre triangle: between ACOnet (Austria), CESNET2 (Czech Republic), and SANET (Slovakia) **Worldwide** • BEN, the Breakable Experimental Network: based at RENCI (USA) • ALICE2: linking Latin America and Europe, based at RedCLARA (Chile/Uruguay) • JGN-X: NICT (Japan)
Switching testbeds (i2CAT) • iMinds OpenFlow island: iMinds (Belgium) • i2CAT OpenFlow island: i2CAT (Spain) • Essex OpenFlow island: UNIVBRIS (UK) • NITOS OpenFlow island: UTH (Greece) • NICTA OpenFlow island: NICTA (Australia) • KOREN (NIA) OpenFlow Island (Korea)	**In Europe** • CNIT OpenFlow island: CNIT (Italy) • CREATE-NET OpenFlow island: CREATE-NET (Italy) **Worldwide** • OpenFlow islands in the US: Clemson University, Rutgers University, Georgia Tech, Indiana University, University of Wisconsin Madison, University of Washington, Stanford University, Kansas State University • FIBRE OpenFlow islands in Brazil: Federal University of Pará (UFPA), Salvador University (UNIFACS), Federal University of Goiás (UFG), Federal University of Rio de Janeiro (UFRJ), Fluminense Federal University (UFF), Brazil's National Education and Research Network (RNP-RJ), Federal University of São Carlos (UFSCar), Telecommunications Research and Development Centre (CPqD), University of São Paulo (USP) • OpenFlow testbed in Japan: JGN-

	X (Japan)
Emulation testbeds (iMinds) • iMinds Virtual Walls 1, 2, and 3 (all Emulab-based): iMinds (Belgium) • FEDERICA: PoPs at NTUA (Greece), i2CAT (Spain) as well as non-Fed4FIRE institutions: CESNET (Czech Republic), DFN (Germany), GARR (Italy), PSNC (Poland), KTH (Sweden), NIIF (Hungary), GRNET (Greece), SWITCH (Switzerland), RedIRIS (Spain), FCCN (Portugal), HEAnet (Ireland)	**In Europe** • HEN facility: University College London, University of Hannover, LAAS **Worldwide** • ProtoGENI facilities (Emulab-based, several in the USA: Wisconsin, Deter at USC, Kentucky, BBN, Georgia Tech, Utah, Louisiana, Florida, UMass, etc.) • TWISC (Taiwan) • KISTI (Korea)
Wireless LAN testbeds: Wi-Fi, Bluetooth, etc. (UTH) • w-iLab.t Office: iMinds (Belgium) • w-iLab.t Zwijnaarde: iMinds (Belgium) • NITOS (ORBIT-based): UTH (Greece) • NORBIT (ORBIT-based): NICTA (Australia) • FIT-UPMC (ORBIT-based): UPMC (France) • FIT-INRIA-Sophia (ORBIT-based): UPMC (France) • Netmode: NTUA (Greece)	**In Europe** • DOTSEL: ETH Zurich (Switzerland) • FIT-IT-Evry: Institut Telecom (France) • Testbeds not publicly available but based on ORBIT technology at: Thomson (France), IT Aveiro (Portugal), CINI (Italy) **Worldwide** • At Rutgers University (USA): ORBIT and other indoor testbeds • At NYU Poly (USA) • Testbeds not publicly available but based on ORBIT technology at: GIST (Korea), UCLA (USA), WiCO (China)
Software defined radio testbeds (UTH) • w-iLab.t Office: iMinds (Belgium) • w-iLab.t Zwijnaarde: iMinds (Belgium) • NITOS: UTH (Greece)	**In Europe** • FIT-INSA: INSA (France) • TWIST: TU Berlin (Germany) • Dublin: Trinity College Dublin (Ireland) **Worldwide** • GENI initiative on WiMAX
Sensor networking / embedded object testbed (iMinds) • w-iLab.t Office: iMinds (Belgium) • w-iLab.t Zwijnaarde: iMinds (Belgium) • SmartSantander: UC (Spain)	**In Europe** • FIT-INRIA-Grenoble: INRIA (France) • FIT-INRIA-Rocquencourt: INRIA (France) • FIT-INRIA-Lille: INRIA (France) • FIT-LSIIT: U Strasbourg (France) • FIT-IT-Paris: Institut Telecom

	(France) • TWIST: TU Berlin (Germany) **Worldwide** • ORBIT: Rutgers University (USA) • Motescope: Berkeley (USA) • Motelab: Harvard (USA) • Kansei: Ohio (USA) • Mirage: Intel (USA) • Wymanpark: John Hopkins (USA) • Tutornet: USC (USA)
Cellular wireless testbeds: LTE, 3G, WiMax (UTH) • NITOS: UTH (Greece) • NICTA outdoor testbed: NICTA (Australia)	**In Europe** • LTE/LTE advanced: TUDresden (Germany) **Worldwide** • WiMAX at Columbia University • WiMAX at Polytechnic Institute of NYU • WiMAX at UCLA • WiMAX at University of Colorado at Boulder • WiMAX at UMass Amherst • WiMAX at University of Wisconsin • WiMAX at BBN Technologies • Winlab outdoor testbed: Rutgers U (USA)