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**

D3.4/D4.4 – Third input from community to architecture

Work package	WP3, WP4
Task	T3.1, T4.1
Due date	30/09/2013
Submission date	31/10/2013
Deliverable lead	Ciro Scognamiglio (UPMC), Felicia Lobillo (Atos)
Version	1.0
Authors	Felicia Lobillo (Atos) Javier García (Atos) Mark Sawyer (EPCC) Gareth Francis (EPCC) Pablo Sotres (UC) Günter Thomas (FOKUS) Mikhail Smirnov (FOKUS) David Margery (INRIA) Ciro Scognamiglio (UPMC) Chrysa Papagianni (NTUA) Frederic Francois (Bristol) Carlos Bermudo (i2CAT) Brecht Vermeulen (iMinds)

Reviewers	Kostas Kavoussanakis (EPCC) Steve Taylor (IT Innovation) Tim Wauters (iMind) Peter Van Daele (iMinds)

Abstract	This document provides the third iteration of high level requirements to WP2 “Architecture”, WP5 “Experiment lifecycle” and WP6 “Measurement and Monitoring” from the Services and Applications community’s perspective
Keywords	Requirements, Services, Applications, Fed4FIRE

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

Disclaimer

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

Executive Summary

The purpose of this document "D3.4/D4.4 Third input from community to architecture" is to report the results of an exercise that has gathered the third set of requirements from both the Infrastructure and the Services and Applications communities' perspective in order to finalise the construction of federation of FIRE facilities. These communities have a strong link with Fed4FIRE, since all partners within the project are members of relevant (sub-)communities and participate in associated FIRE projects. Additionally, several Fed4FIRE partners are involved in the FI-PPP initiative (Atos, Fraunhofer, iMinds, IT-Innovation, UC, etc.).

For the third cycle, we have collected requirements from the following sources:

- ongoing experiments coming from the 1st Fed4FIRE Open Call – 8 experiments involving WP3 and WP4 testbeds
- proposals presented to the 2nd Fed4FIRE Open Call – 11 proposals from Industry and 8 from Academia
- proposals presented to the 1st Fed4FIRE SME Open Call – 4 proposals in total involving WP3 and WP4 testbeds .

The requirements have been mostly evaluated by our community by filling a survey and, in the case of the ongoing experiments from the 1st Open Call, which already have an important background as far as interaction with the federation is concerned, also by providing answers to an open questionnaire with answers in free text.

The requirement coverage map resulting from cycle 2 was reported in "D2.4 Second federation architecture". Since the survey was built out of requirements partially or not tacked in cycle 2, it only focuses on pending functionality or features. However, requirements have evolved and new requirements have been added during the process.

The survey scorings highlight some of the most important requirements for our community such as, for example, the need for uniformity of resource information for resources belonging to different testbeds as far as their presentation is concerned, the availability of orchestration tools to ease the interaction with the federation or the possibility to have detailed monitoring and eventual trouble information (events) during experimentation.

When looking at the answers provided to this open questionnaire, we find important insights concerning how our experimenters experience the interaction with the federation. For example, they seem to greatly appreciate support and guidance when setting up the experiments as well as available documentation and they would like the JFED tool to be extended with further functionality.

We have also taken the opportunity to analyse the full text of the proposals presented to the 2nd Open Call in order to match old requirements coming from or not tackled in cycle 2 and we have found that most of these requirements remain of low priority, except one requirement related to interconnectivity.

Work regarding requirements does not stop at this deliverable even if cycle 3 is the last one foreseen for Fed4FIRE architecture. We are currently distributing a simplified version of the survey among for proposals presented to the 2nd SME Open Call and will collect further inputs from ongoing experiments in order to gather further recommendations for the future operation of the federation.

Acronyms and Abbreviations

AM	Aggregate Manager
API	Application Programming Interface
CA	Certificate Authority
CPU	Central Processing Unit
EC	Experiment Controller
FI-PPP	Future Internet Public-Private Partnership
FIRE	Future Internet Research & Experimentation
FLS	First Level Support
FRCP	Federated Resource Control Protocol
GENI	Global Environment for Network Innovations
NEPI	Network Experiment Programming Interface
OCCI	Open Cloud Computing Interface
OF	OpenFlow
OMF	Control and Management Framework
OML	OML Measurement Library
OMSP	OML Measurement Stream Protocol
QoS	Quality of Service
PDP	Policy Decision Point
RSpec	Resource Specification
SFA	Slice Federation Architecture
SIM	Subscriber Identity Module
SNAA	Sensor Network Authentication and Authorization
VM	Virtual Machine
WSN	Wireless Sensor Network
XMPP	Extensible Messaging and Presence Protocol

Table of Contents

1 Introduction

The purpose of this document "D3.4/D4.4 Third input from community to architecture" is to report the results of an exercise that has gathered the third set of requirements from the Infrastructure and the Services and Applications communities' perspective in order to finalise the construction of federation of FIRE facilities.

This document follows D3.2 [7] and D4.2 [8], which contained requirements for the second cycle of the federation construction. Requirements in D3.2 and D4.2 were based on the following inputs:

- First cycle requirements: derived from generic use cases whose interest was the combined use - and the federation - of several facilities as the mechanism to enable "research by experimentation" over these complex scenarios. These scenarios were inspired by real ongoing and realistic needs coming from different sources, from current market and/or research trends to FIRE projects that run on these facilities or even on FI-PPP vertical projects. Some high-level priority requirements were postponed in the first iteration for several reasons – mainly feasibility within the project's time schedule – and also some medium and lower priority requirements were left for later cycles. Others are partially covered or totally covered.
- For the second cycle, the target was to broaden the scope of input sources used for the requirements analysis, especially for the Services and Applications community:
    - The services and application community identified the need for a greater orientation of Fed4FIRE towards the concept of "**software services**" or applications. The second cycle of the Fed4FIRE architecture has thus provided further utilities or software available for the experimenters in order to ease their interaction with the federated resources, which are heterogeneous as far as technology is concerned.
    - **New use cases**, most of them relying on the infrastructure and services provided by the Fed4FIRE testbeds belonging to the Services and Applications community at the end of the first cycle (i.e. BonFIRE, FuSeCo and SmartSantander). This exercise was influenced and inspired by the knowledge of their specific domains of expertise (i.e. Cloud Computing, LTE services and Smart Cities) and they represent typical experiments these communities would carry out. Concerning **FI-PPP**, although the new wave of vertical use cases was at an early stage, the interaction with these projects (e.g. FITMAN [3]) also produced a scenario which, inspired by one of the FI-PPP vertical use cases, could be built over current FIRE wireless and cloud facilities. These use cases were matched to the list of requirements, in order to confirm their validity, adding new ones or modifying others when required.
    - The practical experience and lessons learned by the **BonFIRE** community were also identified as valuable source of requirements for the second cycle in order to ease its integration in the federation.
    - Feedback from **external stakeholders** was also important. This feedback has a main source: the proposals presented to the first Fed4FIRE **open calls**. A total of 55

experimentation proposals were received. These proposals are real examples of the kind of experiments that could be run on the federated facilities and represent an important input as community requirements. All the material provided by these experimenters was analysed (taking the appropriate confidentiality implications into account) in order to confirm current requirements and to incorporate new ones.

In cycle 2, the approach for prioritisation of requirements was to weight them according to how popular they were (how many experiments and use cases where demanding them or would need them).

For **cycle 3**, Fed4FIRE has gathered requirements mostly from the several open calls experiments and proposals, since this is the first hand and most relevant input from our community.

As in previous cycles, the requirements have been classified according to the following functional areas:

- **Experiment lifecycle**: including discovery, reservation, resource usage and experiment control
- **Measurement and Monitoring**: covering metrics, instrumentation, data management
- **Trustworthiness**[1]: gathering federated identity management and access control, privacy, accountability.
- **Interconnection**: including access networks, routing, etc.

In order to gather our community's inputs, a survey was made out of requirements existing in cycle 2 as a starting point. This table was chosen as a first source because it contained many requirements from the different areas within Fed4FIRE (experiment control, monitoring, etc.) we needed to score. This survey was a transcription of the old requirements into a more natural language. The survey was given to our experimenters in order for them to express how important each item was for their experiment. The survey also provided the possibility for them to add new requirements not identified by us. Functional requirements for cycle 3 have been collected by requesting our community to answer this questionnaire with very precise questions. These inputs from our community arrived at Fed4FIRE through different mechanisms, which we have named after a letter as follows:

- As part of the feedback provided by ongoing experiments selected in the 1st Open Call (input a).
- As part of the open calls templates (all proposals presented to the 2nd Open Call) (input b).
- Filling the questionnaire during interviews with 1st SMEs Open Call winners (input c).
- A couple of proposals presented to the 1st SME Open Call (not selected) who filled in the survey (input d).

[1]Requirements related to SLA management, which have been gathered in dedicated sessions in the framework of WP7, will directly feed WP2 and will not be included in this deliverable but in the second iteration of D2.1 the "first federation architecture".

Additionally, we have analysed the following information:

- The full text of proposals (beyond the survey) presented to the 2nd Open Call (input e).
- Information gathered as feedback from ongoing experiments (1st Open Call), not restricted to the survey but rather their direct experience in their own words, through conversations after having used the federation testbeds and tools (input f).

Thus, this last iteration intends to gather final requirements to harvest all the conclusions and adjustments with the aim of achieving a sustainable federation.

This document is structured as follows:

- Section 2 describes the requirement elicitation methodology for cycle 3, including the different sources and the process to obtain the final table of requirements using requirements from cycle 2 as a starting point.
- Section 3 provides the actual requirements.
- Section 4 includes conclusions and next steps.

2 Requirement elicitation process

This section reports the methodology for requirement elicitation in cycle 3, starting from the sources in subsection 2.1and going through the whole process in subsection 2.2.

The following figure depicts the whole set of sources and the process followed and will be explained in these two subsections. These sources are actually different mechanisms through which WP4 has been able to gather requirements, taking advantage of the different Open Call processes (closed and open).

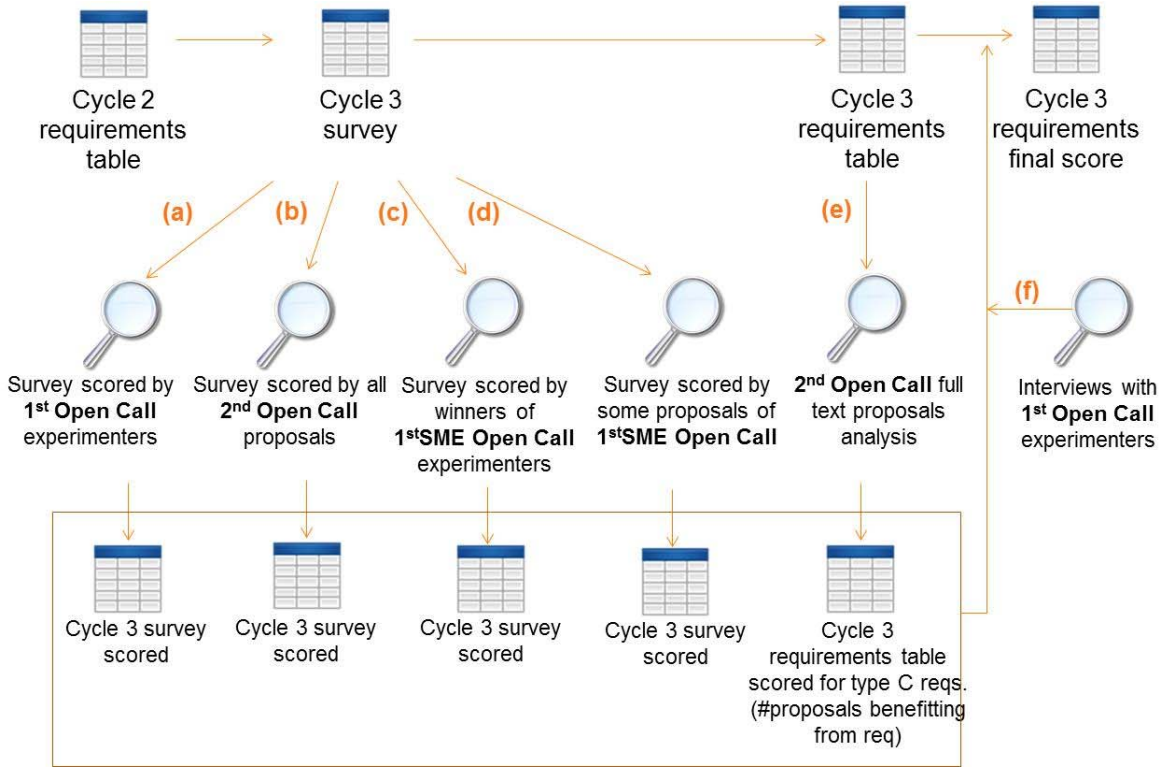

Figure 1: Requirement elicitation process for cycle 3 of Fed4FIRE

2.1 The sources

2.1.1 Pending requirements from cycle 2

Fed4FIRE produced a requirements matrix in cycle 2 that has been considered in the second iteration of the architecture, and analysis of these requirements has been one of the first steps towards the second set of requirements. Some high-level priority requirements such as interconnectivity between testbeds and storage of all kinds of information were kept for later cycles. Some other requirements are now partially fulfilled. All of these have been carried over to cycle 3.

2.1.2 Our community: open calls analysis and surveys

The Fed4FIRE **open calls process** is the fundamental source of requirements in cycle 3. The proposals presented to Fed4FIRE for experimentation are real examples of the kind of experiments that could be run on the federated facilities and represent a first-hand input as community requirements.

2.1.2.1 Surveys and feedback (Open Calls proposals and ongoing experiments)

The set of requirements obtained in cycle 2 was transformed into a questionnaire in order to gather direct feedback from our community. The questionnaire, included in ANNEX 1, was filled by:

- ongoing experiments coming from the 1st Open Call (a),
- proposers as part of the procedure to apply to the 2nd Open Call (b) and
- the winners (c) – as well as some of the non winning proposals – of the 1st SME Open Call (d).

The whole process is depicted in Figure 1.

For experiments selected in the 1st Open Call (a), the survey was directly filled as part of the official feedback form.

Applicants to the 2nd Open Call were requested to fill in the survey as part of the application procedure (b).

For the 1st SME Open Call experiments, since these have joined the federation more recently and thus do not have the same background concerning Fed4FIRE, the survey has been filled during an interview in order to provide further context and explanations in case they were required (c, d).

Compared to the requirements table upon which it is based, we made an effort so that the questionnaire contained more natural language (less technical). Also, specific questions Fed4FIRE has required to have answers to in a more detailed manner were included. Moreover, some of the requirements have been modified in order to make them clearer to our community or simply disregarded (e.g. some low priority requirements) in order to have a shorter list of questions, easier to answer and less time consuming for our experimenters.

Moreover, we have had conversations with ongoing experiments coming from the 1st Open Call (f) in order to gather their direct feedback in their own words. This kind of information is very useful in order to understand how they have found the federation testbeds and tools when going through the process of setting up their experiments and using the resources.

As for quantification, the number of answers for each community group (which is important for the evaluation of the importance of a requirement), is as follows:

- a) and f) 6 experiments involve WP3 testbeds, 7 experiments involve WP4 testbeds
- b) 11 proposals from Industry and 8 from Academia
- c) 1 SME experiment using WP3 testbeds, 1 SME experiment using WP4 testbeds
- d) 2 proposals having answered the questionnaire

2.1.2.2 Analysis of proposals text

All the material provided by the 2nd Open Calls experimenters – respecting of course their confidentiality requirements – has been thoroughly analysed in order to better understand their needs (e).

Following a similar procedure to the one we applied for cycle 2, in cycle 3 all 2nd Open Calls proposals involving WP3 and WP4 testbeds have been reviewed following a common procedure, consisting of going through them as if the reviewer were in charge of helping/guiding these experimenters deploy their experiments across the testbeds mentioned in it and asking ourselves questions such as: "What do they need?" "What are they assuming?" "What do they expect Fed4FIRE to provide?"

As we did in the previous cycle, we have focused on federation instead of what these proposals require from individual testbeds. For example, some proposals might assume a certain testbed is offering functionality beyond their current capabilities (e.g. some wireless technology a testbed does not currently offer). In that case, each individual facility might retain this as a requirement for its own internal roadmap as a testbed, but this has not been considered a requirement for the federation (e.g. in the previous case of a wireless technology, the requirement for the federation would be to be able to expose the actual technology the testbed is offering and to allow the experimenter to get their experiment running on these nodes).

The process has consisted of comparing the content of each proposal and mapping each to as many requirements as possible, modifying the requirements when needed, and adding new ones when no match was possible. All submitted proposals involving WP3 and WP4 testbeds were analysed. Since some proposals were confidential, we do not indicate which proposal beneffited from which requirements. Instead, we determined an all-inclusive set of requirements, counted the number of proposals that had each requirement, and considered the total weight for each requirement as the sum of proposals benefitting from it. This exercise has mainly focused on requirements coming from cycle 2 that were not included in the survey, since for those included we did have answers to the survey provided by the experimenters themselves (input a).

At the time of writing, it is not known which of the proposals submitted to the 2nd Open Call will be finally funded, since this process is still ongoing. However, for the requirement gathering process we care about all of them, not only the winners. Later one, each of the selected proposals will analyse their own detailed requirements for implementation purposes when the time comes before experiment setup time, which is beyond the scope of this document. We intend to present and discuss the set of requirements resulting from our analysis of all the open call proposals with the winning proposers in order to check their validity, including any modifications arising from these discussions. However, we believe that all proposals (winners or not), are important for Fed4FIRE as a source of requirements since they represent real examples of what our community of experimenters

would like to do with the federation. Therefore, at the moment, we are treating each proposal equally.

The outcome of this analysis can be found in sections 3.1.2, 3.2.2 and 3.4.2.

2.2 The process

As a first step, the initial set of requirements coming from cycle 2 (cycle 2 requirements table) was turned into a survey meant to gather inputs from our communities. Cycle 2 requirements table was adapted it in order to make it easier for experimenters to answer the questionnaire (cycle 3 survey). Some of the requirements were simplified, adapted to more natural and less technical language, some were grouped, some were added, some were removed in order to make the process of answering shorter – and thus less time consuming -, etc.

The initial set of requirements (cycle 2 requirements table) was then evolved in order to incorporate aspects included in the survey (cycle 3 survey), which resulted in the cycle 3 requirements table (see Figure 1)**Error! Reference source not found.**. Rewording was included for the sake of clarity and modifications with further explanations are examples of this evolution. This methodology was meant to clearly show the evolution of the requirements' table established from the beginning of the project.

The table was made available to all partners involved in the process as a Google spreadsheet and this was used as the main collaboration tool to gather all the inputs coming from the different sources, since the work was split among all testbed providers and involved partners.

As a result, the table of requirements for cycle 3 contains three types of requirements for which we have different weighting mechanism, as represented in Figure 2. Some of them (the ones existing in the survey) are scored from 0-4 whereas some others simply sum the number of proposals benefitting from one requirements (the ones not existing in the survey).

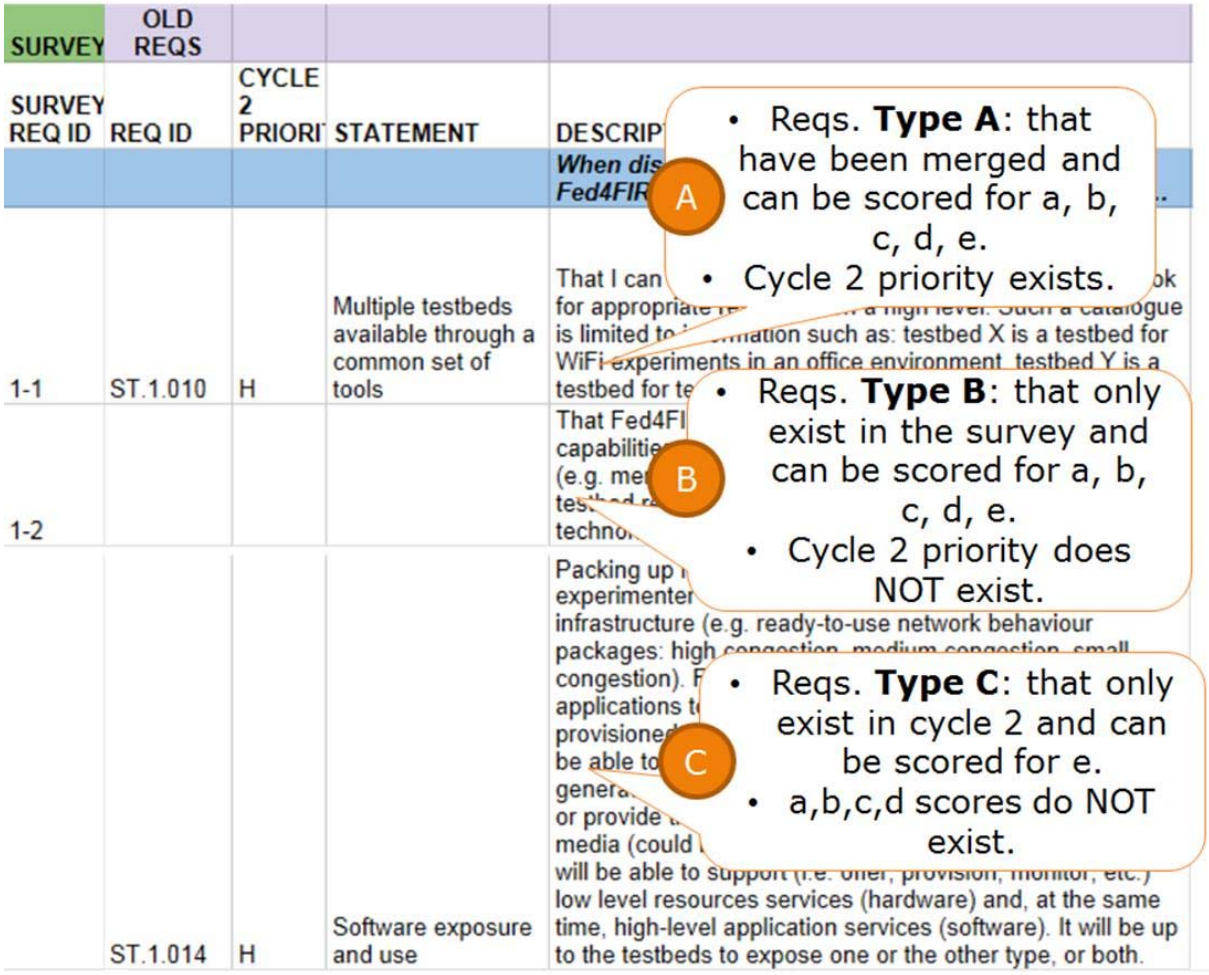

Figure 2: Different types of requirements for cycle 3 of Fed4FIRE

The threes types of requirements are explained below:

- **Requirements type A**: for these requirements, a match was found between cycle 2 requirements table and the cycle 3 survey (searching the original requirement from which each survey item derived). They have thus been merged and we have scoring for processes a, b, c, d and e. Moreover, a priority from cycle 2 also exists for these requirements.
- **Requirements type B**: these requirements only exist in the cycle 3 survey (they were added from scratch to the survey). They have scoring for processes a, b, c, d and e but a priority from cycle 2 does not exist for them.
- **Requirements type C**: these requirements only exist in cycle 2 requirements (they were not included in the survey because they were too specific). They have no scoring for processes a, b, c, d and e but only a priority from cycle 2.

Let us recall that the cycle 3 survey was used as a template to gather the feedback from our community, for:

- Ongoing experiments selected in the 1st Open Call: represented as (a) in **Error! Reference source not found.**.
- 2nd Open Call proposers: represented as (b) in Figure 1.
- Ongoing experiments selected in the 1st SME Open Call: represented as (c) in Figure 1.
- Several non-winning proposals presented to the 1st SME Open Call: represented as (d) in Figure 1.

The result of this exercise was a scoring (out of three different sources) for the requirements included in the survey, using the following weighting method:

- X = no opinion or not applicable to your experiment/ environment
- 1 = not required
- 2 = nice to have
- 3 = important
- 4 = must have

The process has thus produced three different types of requirements (A, B and C). Since the weights for A and B (X, 0, 1, 2, 3, 4) differ from those applied to C (where we count the number of proposals benefitting from one requirement), we have separated these requirements in two different blocks: A+B on one side and C on the other.

For block A+B, we have considered that the following inputs are equivalent:

- X = no opinion or not applicable to your experiment/ environment
- 1 = not required
- Null = no answer provided

Thus, we have converted all these values to '0', so that they do not sum in the average but they do compute as an entry. This means that we consider equally the fact that a requirement is not needed and the fact that experimenters mention that a requirement does not apply to a particular experiment, and the fact that they do not have an opinion on that (after all, we intend to highlight those requirements that our community considers most relevant).

The rest of values (from 2 to 4) have been subtracted 1 in order to have values from 0 to 3.

- 0 = not important
- 1 = nice to have
- 2 = important
- 3 = must have

In parallel, the information provided in the questionnaire, as in cycle 2, proposals presented to the 2nd Open Call were also thoroughly analysed by Fed4FIRE in order to find further insights beyond the questionnaires using the cycle 2 requirements table. Weights have been provided according to how many proposals benefit from a requirement (the requirement weights "1" every time a match is found), mostly for type C requirements. This is represented in Figure 1 as arrow (e).

Finally, the conversations held with ongoing experiments coming from the 1st Open Call (f) in order to gather their direct feedback in their own words have also been considered as an input for cycle 3. This kind of information is very useful to understand how they have found the federation testbeds and tools when going through the process of setting up their experiments and using the resources (represented as (f) in Figure 1). This information has been used as an additional input for the prioritisation of requirements (please refer to ANNEX 2 for further information concerning the list of questions).

3 Requirements

The process described in section 2 has produced two blocks of requirements that have been analysed and prioritised.
As explained in section 2.2, we have three types of requirements: A, B and C.

Requirements from **types A+B** (existing in the survey) have been considered in processes (a), (b), (c) and (d) (please refer to Figure 1). These requirements are presented using the same format, shown on Table 1 below:

Table 1: template for requirements description (A+B)

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) - b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE

The headings are explained below:

- Survey Req ID: Identifier of the question within the cycle 3 survey.
- Req. Id: Requirement Identifier to ease tracing. It has the format "ST.Area.number". The possible fields are as follows:
  - ST stands for Services and Tools
  - Areas :
    - 1: Experiment Workflow and Lifecycle Management
    - 2: Measurements and Monitoring
    - 3: Trustworthiness
    - 4: Facility Management
  - Number: Requirement number within each area, e.g: 001, 002, etc.
- Description: Descriptive text for the requirement.
- Comments: additional information regarding the requirement
- Cycle 2 priority: Priority the requirement had in cycle 2, if existing. H (High), M (Medium), L (Low)[2]
- Average 1st open Call survey (a): Average of survey scores among 1st open call experimenters (a) – 7 experiments involved
- Average 2nd Open Call survey results (Industry) – (b): Average of survey scores among 2nd Open Call proposers coming from for Industry (b)
- Average 2nd Open Call survey results (Academic) – (b): Average of survey scores among 2nd Open Call proposers coming from for Academia (b)
- Average 1st SME Open Call winners (c): Average of survey scores among SME Open Call winners (c)
- Average 1st SME Open Call survey (d): Average of survey scores among 1st SME Open Call proposers (d)
- GLOBAL AVERAGE: Average of survey scores including all inputs

[2] Please refer to Fed4FIRE 318389 D4-2 Second input from Services and Applications community to the architecture for further details

As explained in section 2.2, all averages have been calculated using the following scale:

- 0 = not important
- 1 = nice to have
- 2 = important
- 3 = must have

All requirements are shown in order of global average value, meaning that those with higher value to experimenters are shown first. Moreover, the community having placed the highest value to each particular requirement is highlighted in a different colour. This is useful in order to provide a context for technical workpackages in charge of the implementation in order to see who each requirement is benefitting the most. The number of experiments within each community is also interesting (please refer to section 2.1).

For requirements resulting from the merge of cycle 2 requirements table and cycle 3 survey (type A), the old priority from cycle 2 is also recorded in order to provide further context to technical workpackages.

Requirements from **type C** (not existing in the survey) have been considered only in process (e) (please refer to Figure 1) and they have been gathered in a different table as shown on Table 2.

Table 2: template for requirements description (C)

REQ ID	Description	Comments	CYCLE 2 PRIORITY	N of proposals benefitting from req

The headings are explained below:

- Req. Id: Requirement Identifier to ease tracing as in the previous table.
- Description: Descriptive text for the requirement.
- Comments: additional information regarding the requirement
- Cycle 2 priority: Priority the requirement had in cycle 2, if existing. H (High), M (Medium), L (Low)[3]
- N of proposals benefitting from req Number of experiments requesting this requirement. For requirements derived from proposals presented to the 2nd open calls (e), this represents the number of proposals that would benefit from this requirement.

All of these requirements are presented in two different tables (A+B and C) for each functional domain in the following subsections (3.1-3.4).

The feedback provided by ongoing experiments coming from the 1st Open Call (f), this information has been collected and reported in section 3.5.

[3] Please refer to Fed4FIRE 318389 D4-2 Second input from Services and Applications community to the architecture for further details

3.1 Experiment Workflow and Lifecycle Management

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

3.1.1 Cycle 3 survey requirements (A+B)

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) -b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE
When discovering the different resources that Fed4FIRE can offer me for my experiment, I require …										
'1-3'	ST.1.003	That the above view on node capabilities is the same across the different testbeds of the federation. This means that when describing the characteristics of resources, all testbeds should adopt the same units (e.g. represent RAM always in MB, and not sometimes in MB and sometimes in GB) and use the same parameter names for aspects that mean the same (e.g. always talk about "RAM", and not "RAM" on some testbeds, "working memory" on some others and just "memory" on a third group of testbeds). Resources must be described in a homogeneous manner so that the experimenter can compare the resources in different testbeds, giving the experimenter a view into the internal steps the provisioning of his resources go through.	Fed4FIRE must able to deploy an experiment on multiple testbeds selected by the experimenter.	H	2.86	2.00	2.38	1.50	2.50	**3.00**
'1-1'	ST.1.001	That I can browse some kind of resource catalogue to look for appropriate resources on a high level. Such a catalogue is limited to information such as: testbed X is a testbed for WiFi experiments in an office environment, testbed Y is a testbed for testing cloud applications, etc.	Data sharing, connectivity and experiment management across multiple testbeds - Data management & migration of data	H	3.00	2.55	2.38	1.00	2.50	**2.50**

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) -b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE
'1-2'	ST.1.002	That Fed4FIRE provides a detailed view on what node capabilities are available on every testbed of the federation (e.g. mentioning information for every resource of a testbed regarding CPU speed, RAM, supported 802.11 technology, optical networking interfaces, etc), including if possible detailed information about the full set of configurable QoS parameters and QoS interfaces and information such as a detailed view on each hardware or software component on every available node on every testbed of the federation (e.g. modem radio model, modem radio driver and interoperability with linux kernel modules like pktgen)			3.00	2.27	2.25	1.50	2.00	**2.37**
'1-6'	ST.1.006	That for nodes that have static network connections to other nodes in the same testbed, that it should be possible to identify the corresponding physical topology. In the wired domain this means that you can know how the nodes are connected to each other. For wireless resources this means that you know which resources are in transmission range of each other.			1.71	2.00	1.88	1.50	3.00	**1.93**
'1-5'	ST.1.005	That I know the location of the site where resources are located. Per site, this location information can be exactly the same for all resources.			2.14	1.55	1.50	1.00	1.00	**1.60**
'1-4'	ST.1.004	That next to browsing through information about what is available, that I can actively search for the existence of resources with certain characteristics by defining a specific query (e.g. something that is similar to an SQL query, e.g. select resources from all testbeds where RAM >= 8 GB)			2.29	1.64	1.38	0.00	1.00	**1.57**

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) -b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE
'1-9'	ST.1.009	That I can assess which testbeds/resources are more reliable then others (both in terms of provided hardware, software, and wireless interference, possibly based on historical health information about the resources and their environment)			1.71	1.55	1.88	0.00	1.50	**1.57**
'1-8'	ST.1.008	For virtual resources, that I know their physical host and the actual location.			1.86	1.09	1.38	1.50	1.50	**1.40**
'1-7'	ST.1.007	That I have location information about the actual resources that I will use. For example , in wireless nodes the accuracy is particularly important (1 m accuracy)			1.14	1.36	1.00	0.00	1.50	**1.13**
When selecting and reserving resources that I want to include in my Fed4FIRE experiment, I require …										
'2-11'	ST.1.020	That I can easily reserve resources across multiple testbeds using the same common tools. These should also be as user-friendly as possible, abstracting the complexity of the underlying infrastructures for me as much as possible. This way I can focus on the experiment design itself instead of learning how to work with numerous testbed-specific tools.	Scheduling simultaneous access to multiple testbeds. This requirement is also related to resource management on testbeds, SLA management, enforcing limits, making experimenters accountable for the resources they request and use.	H	3.00	2.27	2.38	1.50	0.00	**2.27**
'2-1'	ST.1.010	That when browsing through the resource descriptions, that I can manually select every node that should be added to my experiment. Think of an experience similar to online shopping and putting resources in your shopping cart.			2.57	1.91	2.38	1.00	1.00	**2.07**

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) -b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE
'2-13'	ST.1.022	That I can use a single Fed4FIRE account to select and reserve resources at all different testbeds of the federation. So even when using one common tool for reservation at the different testbeds, I don't want to remember a different username/password combination for every testbeds, and I also don't want to register again at every testbed that I want to use. Of course, registering for that one Fed4FIRE account should also be straightforward. Experimenters will use a single set of credentials for all operations on the platform (experimenting, monitoring, etc.) regardless of the tool used (handover). The authentication and authorisation processes to grant access to resources should be as light as possible for the experimenter. Finally, any Identity information must be protected within Fed4FIRE.	It should enable the creation of an account on different testbeds and/or services and link them to a unique Fed4FIRE user. ID management and mapping. Privacy and data protection are also important. For example, passwords must never be sent by email.	H	2.57	1.55	2.63	1.50	0.50	**2.00**
'2-5'	ST.1.014	That I can reserve resources. It is OK for me that they are shared with others (soft reservation, e.g. requesting a virtual machine that will be deployed on a physical server that is used by other experiments also), as long as I know that I will also have guaranteed access to them.			2.29	1.91	2.25	0.50	0.00	**1.87**
'2-8'	ST.1.017	That situations are avoided where a have to wait days or weeks before being able to use the testbed because of long reservations of others.			3.00	1.36	1.75	1.50	1.50	**1.87**
'2-12'	ST.1.021	That when reserving resources across multiple testbeds, that there is guidance in finding the first appropriate time when all the resources that I want across the testbeds would all be available.			2.86	1.09	2.38	1.50	0.50	**1.83**
'2-10'	ST.1.019	That a reservation is approved or rejected quickly (within a few minutes).			3.00	1.18	1.88	1.50	1.00	**1.80**

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) -b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE
'2-15'	ST.1.024	That the testbeds and/or the federation guarantee a certain Service Level to me regarding the execution of my experiment (availability of resources, reliability of resources (uptime/downtime), responsiveness of support services, privacy guarantees, etc).			2.71	1.18	1.75	1.50	1.50	**1.73**
'2-16'	ST.1.025	That I can dynamically scale my resources up and down according that what my experiment needs during its execution. For instance if a server deployed on a VM gets overloaded, I should be able to assign more resource (RAM, CPU cores, etc.) to that running VM, and/or should be able to add a second VM to my running experiment on which I deploy a second instance of that server.	This could imply scaling resources - adding and removing them from testbeds or adding / removing a whole testbed	H	2.29	1.73	1.88	0.50	0.00	**1.70**
'2-2'	ST.1.011	That I can select suitable resources for inclusion in my experiment by defining a specific query (e.g. something that is similar to an SQL query, e.g. select all resources from Virtual Wall where nr_ethernet_cards >= 6)			2.43	1.73	1.63	0.00	0.50	**1.67**
'2-7'	ST.1.016	That next to adding resources to my experiment right now (instant reservation), that I can also define a reservation for any moment in the future (future reservation, e.g. tomorrow from 9AM-5PM).			2.00	2.09	1.00	0.50	1.50	**1.63**
'2-6'	ST.1.015	That I can reserve resources. They have to be exclusively assigned to me (hard reservation, e.g. reserving a virtual machine that will be deployed on a physical machine that is dedicated to your experiment only)			2.71	1.27	1.00	1.50	1.50	**1.57**
'2-14'	ST.1.023	That if testbeds decide to assign me a certain reservation quota (e.g. based on my profile such as student, post-doc, professor, paying customer, etc), that I can request a temporary increase of my quota if really need it (e.g. before a paper deadline)			2.29	1.27	1.13	0.50	1.50	**1.43**

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) -b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE
'2-3'	ST.1.012	That I can temporarily install my own equipment at a Fed4FIRE testbed for testing, and select it to be included in my experiment. My own equipment might be for example mobile devices in a wireless experiment. The fact that this kind of resource appears and disappears might not alter the course of the experimentation. When these devices appear, they might become part of the infrastructure (e.g. as sensing nodes and thus producing information). This also applies for VMs, for example, since experimenters should be able to dynamically create VM (resources), name them and use that name to interact with them.	It relates to registration of resources in testbeds, specification of resources, advertising	H	2.00	1.55	1.25	0.50	0.00	**1.40**
'2-4'	ST.1.013	That the mechanism for registering my own equipment at a testbed is standardized, allowing me to register that equipment at different testbeds in exactly the same manner.			2.29	1.36	1.00	1.00	0.00	**1.37**
'2-17'	ST.1.026	That if I reserved a number of resources at a testbed, that I can divide them over different independent experiments that I am doing at the same time. Experimenters can have more than one experiment running on a given testbed at the same time. He should be able to easily address/group the resources from one experiment. This means that experimenters should be able to use reserved resources for more than one experiment	Could be solved if experimenters are allowed to create slices. Provisioning must be done more than one in this case, so that each experiment starts from a clean state	M	1.86	0.82	1.50	0.50	1.00	**1.23**
'2-9'	ST.1.018	That I can reserve nodes exclusively for myself for a longer period (days or weeks)			2.57	0.64	0.38	0.00	0.00	**0.93**

When using the resources that I included in my Fed4FIRE experiment, I require …										
'3-1'	ST.1.027	That I can SSH to my nodes. Experimenters should be able to use different tools to prepare and run their experiment. For example, have access to CLI tools to perform standard actions	A portal and an API is not enough for experimenters	M	3.00	2.55	2.63	1.50	1.50	**2.53**
'3-2'	ST.1.028	That I have root access to my nodes. This allows me to perform any action on the nodes that I want (install new applications, device drivers, load additional kernel modules, etc).			3.00	2.55	2.50	1.50	1.50	**2.50**
'3-3'	ST.1.029	That I can use a single public/private SSH key pair to access my resources on all the different testbeds			3.00	2.18	2.25	1.00	0.00	**2.17**
'3-7'	ST.1.033	Fed4FIRE must provide the means for experimenters and third parties to develop and/or deploy applications on top of Fed4FIRE infrastructure. This should be done in such a way that these external providers find it attractive to eventually join the federation. An experimenter might be able to install own pieces of software (or software downloaded from the internet) on resources of the facilities involved in his experiment. The experimenter will also have the possibility to manually complete the experiment data and to setup initial data-sets over different facilities. An interface through which the experimenter can enter these data must be provided for this and the data entered must be stored in the selected facility to be used at runtime.	This involves suppliers which might become part of the core federation players	H	2.71	2.36	1.75	1.50	1.50	**2.17**
'3-12'	ST.1.038	That I can easily use my resources across multiple testbeds using the same common tools. These should be as user-friendly as possible, abstracting the complexity of the underlying infrastructures for me as much as possible. This way I can focus on the experiment itself instead of learning how to work with numerous testbed-specific tools. In general, Fed4FIRE tools should be user friendly to the experimenter		H	2.43	1.82	2.38	1.00	2.00	**2.07**
'3-5'	ST.1.031	That I can choose to have a specific Linux distribution on my nodes (e.g. latest Ubuntu LTS release)			2.43	2.00	1.88	1.00	0.50	**1.90**

'3-8'	ST.1.034	That I can take a binary image of the hard drive of my nodes, and that I can store these for later re-use (so flashing the image back later on)			2.57	1.64	1.63	0.50	1.50	**1.77**
'3-11'	ST.1.037	That I can allow other people of my work team that are involved in the experiment to use the resources that I have reserved and deployed. I should be able to specify which resources should be shared, and which not. Resource groups should be shared by default with a group of people the experimenter belongs to. Resources might be shared among several experimenters or used exclusively, depending on the testbed offer and on the experimenters' needs.	Could be solved if users from the same sub-authority can get slice credentials for slices created by others	M	2.00	1.73	1.63	0.00	2.00	**1.67**
'3-10'	ST.1.036	That during the deployment of my resources over different facilities, that my initial data sets can be automatically loaded to all these resources.			2.71	1.18	1.63	0.00	1.50	**1.60**
'3-9'	ST.1.035	That I can define what a node should automatically do at start-up (bootstrap scripts).		H	3.00	1.27	1.38	0.00	0.00	**1.53**
'3-6'	ST.1.032	That I can choose to use a custom Linux kernel on my nodes (e.g. with my own performance upgrade patches to the kernel)			2.29	1.64	1.00	0.50	0.00	**1.43**
'3-4'	ST.1.030	That I can choose to have Windows installed on my nodes			2.14	1.45	0.88	0.00	1.50	**1.37**

When controlling the execution of my experiment in an orchestrated manner, I require …										
'4-3'	ST.1.041	That the description of the above orchestration is described in a human-readable way. This description should also be uniform across the different testbeds.			2.57	1.91	1.75	0.50	2.00	**1.93**
'4-1'	ST.1.039	That I can define the behaviour over time of a distributed experiment in a single script, which can be started automatically at any desired moment, and will be automatically translated to the corresponding triggers at the nodes at the appropriate time. So e.g. describing in a single script that the 5 client nodes in an experiment should gradually increase their load on the server that they are testing in the experiment. This will be done automatically, without the experimenter login in to these 5 nodes and gradually increasing this load manually. It should be possible to write a workflow or other description of an experiment which allows it to be run from start to finish including resource provisioning, resource control, service interaction, monitoring and data collection.	Proposed workflow system could offer this if well integrated with existing / lower level components	L	2.43	2.00	1.63	0.50	2.00	**1.90**
'4-2'	ST.1.040	That I can define the behaviour of a distributed experiment in a single script, based on events (e.g. value above threshold). This can be started automatically at any desired moment, and will be automatically translated to the corresponding triggers at the nodes at the appropriate moment. So e.g. describing in a single script that a server should scale up to a VM with more CPU power and RAM when the load of the clients on the server becomes higher than a certain threshold. Orchestration should involve different testbeds, for example using the results from one testbed as inputs or triggers for the other			2.14	1.82	1.63	0.50	2.00	**1.77**
'4-4'	ST.1.042	That the above description of the orchestrated control of the experiment can also include other aspects that will be performed automatically. This includes selection, reservation and deployment of resources; monitoring of the resources and collection of measurement data during the experiment.		L	2.43	1.27	1.38	0.50	2.50	**1.60**

3.1.2 Cycle 2 pending requirements (C)

3.1.2.1 Infrastructure community

REQ ID	Description	Comments	CYCLE 2 PRIORITY	N of proposals benefitting from req
I.1.103	The different available wireless channels should be considered to be an infrastructure resource.	If channels are a resource, they can easily become discovered and reserved later on.	M	**3**
I.1.107	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.		H	**4**
I.1.108	Resource virtualization (VMs, flows) must provide information about the supporting physical devices and their location		M	**8**
I.1.110	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.		M	**10**
I.1.203	A Federated API is needed which would enable specifying the desired virtualized topologies that will be deployed over the existing physical topology	Examples are drawing a topology on the Virtual Wall that will be automatically translated to a correct selection of machines and VLAN configuration on all ports. Another example is defining the topology of a Flowspace on an OFELIA infrastructure.	M	**14**
I.1.204	The experimenter should be able to specify the need for additional experimentation devices (such as spectrum analyzers or Smartbits measurement devices). These can be devices of which the experimenter knows that the testbeds could optionally provide them, although that they are not exposed as available resources. But these could also be additional equipment (both supporting equipment but also devices under test) owned by the experimenter himself which he temporarily installs at the testbed	E.g., in the discovery phase wireless resources were searched that have 2 IEEE 802.11n interfaces per node. Since there is a large amount of suitable nodes, the experimenter has the luxury to define additional requirements. In this case, the experimenter does not only wants to define the node requirements, but also the fact that a spectrum analyzer should be present at the same site.	L	**5**

I.1.302	A means to enforce fairness with hard reservations is required. Situations where a few users reserve all nodes for too long should be avoided. Similarly, situations where a large amount of users reserve a few resources for a very long time should also be avoided. This kind of reservations is typically done to develop new solutions on the testbed, but makes it harder to schedule other large-scale experiments.	This could be achieved by specifying an expiration date for reservations via calendar or a scheduler, or through the usage of reservation quota. It could be interesting to look at existing techniques applied in high performance computing clusters.	H	**9**
I.1.303	Fed4FIRE must provide a reservation system with adequate security to provide assurance to industrial users		H	**9**
I.1.304	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.		H	**12**
I.1.306	When an experimenter has defined all resources that it wants to include in its experiment, and the desired duration of the experiment, then he should be shown a screen which gives information about all the oncoming timeslots in which all the nodes of the experiment are still available. Selecting the desired timeslot and reserving all of them should be possible with the click of a single button.	Could be considered to be a tool requirement instead of an architectural requirement.	L	**0**
I.1.401	APIs are required to enable direct instantiation of both physical and virtualized resources for experiments. Ideally, these APIs would be compatible with provisioning APIs already supported by the infrastructures and/or existing uniform tools. This would decrease the development costs for the infrastructure providers and tool builders.	Instantiation of physical node would involve powering the node on, and appropriately steering the node boot process. Instantiation of virtualized resources can be related to the setup of virtual machines, OpenFlow flows, etc. For such virtualized resources the API should support an annotation mechanism to define the actual physical resource on which the virtual one should be instantiated.	H	**3**
I.1.404	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.		H	**12**
I.1.406	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.		M	**1**
I.1.408	In case some of the resources cannot be provisioned successfully, you should be informed so that you can decide to continue the experiment, or change it to another testbed or reserve again later on. Such information coming from all the different testbeds involved in a single experiment should be presented to the experimenter in a single location			**3**
I.1.504	The experiment control engine should be general enough to support the control of all possible kinds of Future Internet technology: wireless networks, optical networks, OpenFlow devices, cloud computing platforms, mobile robots, etc. If it is not feasible to control all these resource types with a single tool, you should at least be able to use the same tool per domain (e.g. one tool for all cloud computing tasks, one for all wireless tasks, and one to control the robots movements).		H	**1**

I.1.507	It can be necessary that an experimenter require manual live participation by a person present (e.g. walking around an office environment with a smartphone to validate indoor positioning solutions). For this one should be able to contact the actual support staff of the testbed to make practical arrangements (local staff does it for the experimenter, or the experimenter travels to the testbed for one day)		L	**1**
I.1.501	This is most valuable during the development phase or for debugging purposes.		H	**6**
I.3.101	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.	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.	H	**13**
I.3.103	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.		M	**3**
I.3.104	Access to the Fed4FIRE APIs (discovery, reservation, provisioning, etc.) should also be protected by an authentication mechanism		H	**14**
I.3.105	It should be easy for new experimenters without any affiliation to the federation to create their Fed4FIRE identity		H	**13**
I.3.201	It should be possible for infrastructures to dynamically decide which resources they should make available to a certain Fed4FIRE experimenter, and which experimentation quota that will be appropriate. This can be bases on a set of possible experimenter roles, on specific attributes, etc.	Example of roles could be: master student, PhD student, post-doc, professor, paying customer, etc. Example of attributes could be: affiliation, years of experience, credit card limit, etc.	M	**8**
I.3.202	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.		M	**10**

3.1.2.2 Services and Applications community

REQ ID	Description	Comments	CYCLE 2 PRIORITY	N of proposals benefitting from req
ST.1.043	After launching an experiment, the experimenter must be informed of the required time until the experiment is provisioned and starts running	This requirement can be extended so that Fed4FIRE provides time information until results are available but this could dependent on the experiment than on the federation itself.	L	**0**
ST.1.044	Packing up resources in services offered to the experimenter so that he/she is isolated from the real infrastructure (e.g. ready-to-use network behaviour packages: high congestion, medium congestion, small congestion). Fed4FIRE must provide the means for these applications to be exposed, discovered by experimenters, provisioned and used. For example, experimenters should be able to generate load (from thin clients, using traffic generators, etc.), using existing tools available at testbeds or provide the possibility to gather information from social media (could be simulated). Very importantly, Fed4FIRE will be able to support (i.e. offer, provision, monitor, etc.) low level resources services (hardware) and, at the same time, high-level application services (software). It will be up to the testbeds to expose one or the other type, or both.		H	**1**
ST.1.045	Experimenters should be able to manage resources they have created (share/unshare/destroy)	Experiment replicability might benefit from ability to publicly share VM images	H	**0**
ST.1.046	Fed4FIRE will be able to produce registers for billing purposes and provide them to involved parties		L	**0**
ST.1.047	Fed4FIRE will be able to trace all usage for accountability purposes (testbeds, services, quantity, time, service delivery completion,…). Testbed owners should be able to trace the origin of outgoing traffic to the experimenter responsible for it	Audit and accountability.	M	**0**
ST.1.048	Resources should be able to keep providing information when in offline mode. This information will be sent to the data storage centres once the device (service user) is connected again. When experiment data cannot be uploaded to their final destination for connectivity issues, they must be kept and uploaded when the connection is re-established. Platform should not impose any restriction on providing past information.		M	**0**
ST.1.049	Service providers (experimenters) should be able to access real time information, historical information or a combination of both of them.	Historical data access of monitoring information	M	**1**
ST.1.050	Possibility of comparing different monitoring options (Nagios, Zabix, OML) in order to test applications' behaviour based on this information (monitoring applications)		L	**0**

3.1.3 Cycle 3 new requirements

3.1.3.1 Infrastructure community

REQ ID	Description	Comments	N of proposals benefitting from req
I.1.111	That Fed4FIRE provides a detailed view on each hardware or software component on every available node on every testbed of the federation (e.g. modem radio model, modem radio driver and interoperability with linux kernel modules like pktgen)		**5**
I.1.416	Important to have all data plane connected and 'globally reachable' within the federation.		**1**

3.2 Measurement & Monitoring

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

3.2.1 Cycle 3 survey requirements (A+B)

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) -b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE
When capturing the results of my experiment (monitoring and measuring data), I require …										
'5-11'	ST.2.011	That the overhead of any monitoring and measurement tool is minimal. These tools should have a negligible impact on the results of my experiment.			2.86	2.36	2.25	1.50	2.50	**2.40**

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) -b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE
'5-9'	ST.2.009	That other aspects related to the successful execution of my experiment are continuously monitored, and that I am automatically informed in case of any errors. Examples are: when a selected resource could not be instantiated, when there is a problem with the interconnectivity between the used testbeds, when a used testbed goes down during the experiment, when there is a sudden peak of wireless interference, etc. This might be important when analysing anomalies in the experiment results.			3.00	2.27	1.88	1.50	1.00	**2.20**
'5-15'	ST.2.15	That I can store experiment configurations in order to repeat experiments and compare results of different runs. The experiment setup should be made easy in order not to repeat the same procedure every time (same tools, same user, etc.)		M	2.86	1.73	2.38	0.50	2.50	**2.13**
'5-7'	ST.2.007	That the overall health status of the different testbeds (testbed up or down, has free resources left, etc.) is continuously monitored by the federation, and that in case of issues I am informed of this. Fed4FIRE will deal with errors and exceptions occurred during an experiment and raise alerts providing comprehensive information back to the experimenter, including information from several facilities. Capability to collect information from different testbeds in order to furnish an open platform with all the gathered product data.		M	2.14	1.91	2.25	1.50	2.00	**2.03**

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) -b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE
'5-3'	ST.2.003	That by default some common characteristics of my resources are stored automatically for later analyses during experiment runtime (CPU load, free RAM, Tx errors, etc).			2.43	1.45	2.50	1.50	1.00	**1.93**
'5-2'	ST.2.002	That Fed4FIRE makes it easy for me to retrieve and store data that I measured during the runtime of the experiment. This means that it should be easy to store my measurement somewhere in a way that the data is clearly related to the experiment ID, but without needing to establish connections to certain databases manually from within my code, and without needing to know the specific experiment ID that belongs to my current experiment.			2.57	1.73	2.13	0.50	1.00	**1.90**
'5-8'	ST.2.008	That the overall health status of the different testbeds (testbed up or down, has free resources left, etc.) is continuously monitored by the federation, and that in case of issues the corresponding testbeds try to solve them asap.			2.00	1.73	2.38	0.50	1.50	**1.87**
'5-12'	ST.2.012	That I can store and access my experiment monitoring data and other measurements on a data service on the federation, which is accessible during the experiment (temporarily data storage by the federation). Fed4FIRE must provide storage facilities for the experimenter to retrieve historical data concerning an experiment data-set and results. Data centres have to keep log of data stored and access to external resources. It is	This requirement involves data management and archiving. Questions arise about the long-term storage of experiment data - access control,	M	2.29	1.64	2.13	0.50	2.00	**1.87**

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) -b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE
		important for audits and historical data tracking	management, curation, who pays for long term storage.							
'5-10'	ST.2.010	That when an error requiring manual intervention is reported to me as part of the previous step, that I am guided through the process for recovery.	Operational errors, such as failure of he infrastructure, impossibility to store or retrieve data are some examples of these errors	L	2.29	1.91	1.75	0.50	1.50	**1.83**
'5-14'	ST.2.014	That access to my stored data is properly secured. Experiments must be kept confidential if required, the privacy of experiments, data sets and results should be guaranteed.		L	2.71	1.09	1.88	1.50	2.50	**1.80**
'5-17'	ST.2.017	That I am made aware if my storage capacity is running out.			2.86	1.18	2.00	0.50	2.00	**1.80**
'5-4'	ST.2.004	That for the above monitoring, that I can select and configure how this data should be collected (always at a specified interval, only after a certain event or alarm, define some specific filters, etc). Fed4FIRE must provide tools to create, view, update, and terminate monitoring configurations related to shared resource types or experiments in real time. Monitoring data should be reportable for visualisation and analysis purposes with several reporting strategies (only alarms, all data, filters, etc.) in real time in order to provide accurate information and ease	As most of the monitoring in cycle 1 is done by using OML streams, experimenters should be able to create and/or configure new/existing OML Measurement Points in real		2.43	1.36	1.88	0.50	1.50	**1.70**

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) -b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE
		the analysis process. The experimenter might create own aggregated/composite monitored elements out of the available ones when designing the experiment, deciding what to monitor, defining some filtering possibilities as well, and providing the destination endpoint to send the information to. Monitoring information must cover information from different facilities and services. Monitoring metrics should be compatible across different facilities. For example, Monitoring computing & network resources' capacity. The monitored data during an experiment runtime will be available to the experimenter for all components involved in the experiment. If not aggregated, at least monitoring information from all involved testbeds must be provided. Testbeds must be able to publish infrastructure status through an API	time (maybe via FRCP)							
'5-13'	ST.2.013	That Fed4FIRE makes it easy for me to retrieve and store data that I measured during the runtime of the experiment. This means that it should be easy to store my measurement somewhere in a way that the data is clearly related to the experiment ID, but without needing to establish connections to certain databases manually from within my code, and without needing to know the specific experiment ID that belongs to my current experiment.		M	2.14	1.45	2.13	0.50	1.00	**1.70**

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) -b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE
'5-1'	ST.2.001	That the internal clocks of resources across multiple testbeds are synchronized very accurately; Comment from MobileTrain: "Different experiments need different time accuracy. Sometimes NTP is enough, whereas other times it is necessary a GPS module on each node."			1.71	1.45	1.13	1.50	3.00	**1.53**
'5-5'	ST.2.005	That I might create own aggregated/composite monitored elements out of the available ones when designing the experiment, deciding what to monitor, defining some filtering possibilities as well		H	1.86	1.45	1.38	0.00	1.00	**1.40**
'5-16'	ST.2.016	That I can share my stored data with specific others (individuals and/or groups), or even make them publically available			1.43	1.64	1.38	0.00	1.00	**1.37**
'5-6'	ST.2.006	That information about external wireless interference during the execution of my experiment is automatically provided for me.			0.57	0.82	1.13	1.50	2.50	**1.00**

3.2.2 Cycle 2 pending requirements (C)

3.2.2.1 Infrastructure community

REQ ID	Description	Comments	CYCLE 2 PRIORITY	N of proposals benefitting from req
I.2.205	Fed4FIRE should give the possibility to store metadata about the data of the experiments.	This allows easy lookup of experiment results, and eases the assessment of the meaning of the data. This seems most valuable when looking for shared data of other experimenters.	M	**0**
I.2.107	The user must be able to request on-demand measurements. In order to do so, they will need to express that they want agents with such on-demand polling capacities	The same information can be retrieved by looking into the output of the monitoring and measurement tools that will continuously provide measurements during the experiment run-time. However the on-demand measurement is more convenient during experiment development and debugging.	H	**0**
I.2.108	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.	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.	M	**0**
I.3.401	Fed4FIRE should provide a method for querying and reporting the reliability of a testbed in terms of provided hardware, software and present wireless interference.	A possible approach could be to monitor facility resources to observe service experience. Regular questioning of the experimenters about their experience could be another possibility. In this case attention should be given to	M	**5**

REQ ID	Description	Comments	CYCLE 2 PRIORITY	N of proposals benefitting from req
		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.		
I.3.402	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.		M	**0**
I.3.403	Fed4FIRE should provide the possibility to trace network traffic back to the originating experiment. This is useful when misuse of the infrastructure has been detected and the corresponding experimenter should be sanctioned (e.g., by revoking his/her account). The fact that accountability mechanisms are in place will automatically increase the level of trust that infrastructure providers can have in Fed4FIRE experimenters which are unknown to them.	FIRE facilities can be powerful tools, and misuse should most definitely be handled adequately.	M	**1**

3.2.2.2 Services and Applications community

REQ ID	Description	Comments	CYCLE 2 PRIORITY	N of proposals benefitting from req
ST.2.018	Different services in different testbeds should be used with different identities (e.g. a shopping profile can be integrated with a supporter profile but waste cans require different identifiers)		L	0
ST.2.019	There has to be a mechanism that assures service users and experimenters that the providers are trusted.	Information providers must be identified and Fed4FIRE must ensure that they are who they say they are	M	0
ST.2.020	In addition to basic authorisation information, additional profiles related to a specific application must be definable for one service user		L	0
ST.2.021	The real id or owner of the device, who decides whether the information is sent, collected, etc. (external resource involved in an experiment) has to be kept hidden for other Service Users in order to avoid any legal problems. An alias or hash code should be used instead. A mechanism to unbind data from owner or real device identity for experimenters or service users is required.	This requirement is about the privacy of service users	L	0
ST.2.022	Provide the means to get end user opinion on service deployed on top of the facilities provided on Fed4FIRE in order to measure the reputation of the facilities and services provided. Facilitate interaction between experimenters and federated facilities.	This is linked to reputation	M	0
ST.2.023	Service providers can register to notifications on changes on information of their interest		L	0

REQ ID	Description	Comments	CYCLE 2 PRIORITY	N of proposals benefitting from req
ST.2.024	Fed4FIRE will make the distinction between requests of local users, PhD students from other institutes (research), students (practical exercises), in order to know what kind of experimenter is logging in and from where and apply policies accordingly. Fed4FIRE must provide secure mechanisms to retrieve information produced by experimenters during experiments or to use resources privately. This access should be controlled according to these profiles. Moreover, Fed4FIRE must provide the means to authenticate different users belonging to several organisations that might collaborate in the same experiment. Of course, within the limits accepted by all parties in the federation, testbed owners should be able to control access to their testbed. Finally, within the limits accepted by all parties in the federation, testbed owners should be able to grant credentials to experimenters outside the scope of the federation for local usage.	Identification of user roles – policies could define rights users may have on resources - e.g. PhD students have less rights than a principal investigator. If they have complementary expertise, which is usually the case, they might be accessing different testbeds (not the same) for the same experiment, but this should not be restricted.	H	0
ST.2.025	Testbed owners must present to funding parties data about: * scientific relevance of their testbed, probably by tracking publication linked to past usage and success stories * socio-economic impact, in particular usage of the platform by industry and SMEs * type of experiments run using their testbed * user provenance (geographic and type of users)		M	0

3.3 Trustworthiness

Trustworthiness gathers tools and services targeted to increase the trust in FIRE considering community views on trust (e.g. academic vs. commercial), id management, SLA management[4], etc.
Trustworthiness requirements will be included in the next issue of the architecture document (cycle) since they have not been provided by our community of experimenters but rather internally by the testbeds. Experimenters have not expressed strong technical or functional preferences as far as SLAs are concerned, beyond the wish of their existence.

3.4 Interconnection

3.4.1 Cycle 3 survey requirements (A+B)

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) -b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE
When focusing on the connectivity of the resources that will be included in my Fed4FIRE experiment, I require …										
'6-1'	ST.4.001	That resources at different testbeds are interconnected on layer 3 (IP)			2.86	1.64	1.38	1.50	1.50	**1.83**
'6-3'	ST.4.003	That I can know the type of interconnections that are available between the testbeds (layer 2 and/or layer 3, NAT or VPN included, dedicated direct link, connected through Géant with or without bandwidth reservation, connected over the public Internet, …)			2.57	1.36	2.13	0.50	1.50	**1.80**

[4] In this cycle, requirements related to SLA management, which have been gathered in dedicated sessions in the framework of WP7, will directly feed WP2 and will not be included in this deliverable but in the third iteration the "Thirdfederation architecture".

SURVEY REQ ID	REQ ID	DESCRIPTION	COMMENTS	CYCLE 2 PRIORITY	Average 1st open Call survey (a)	Average 2nd Open Call survey results (Industry) - b	Average 2nd Open Call survey results (Academic) -b	Average 1st SME Open Call winners (c)	Average 1st SME Open Call survey (d)	GLOBAL AVERAGE
'6-5'	ST.4.005	That my resources are directly reachable, without any network address translation (NAT) or virtual private network (VPN) in between. So actually I require that all resources have a public IPv4 or IPv6 address.			2.14	2.09	1.38	0.50	1.00	**1.73**
'6-6'	ST.4.006	That if an issue arises with the interconnection between my used testbeds, that I am automatically informed about this.			2.29	1.55	1.88	0.50	1.00	**1.70**
'6-4'	ST.4.004	That I can configure a specific bandwidth on the interconnections between the different testbeds used in my experiment. As long as the links behave as configured, I don't really care what the testbed has to do behind the curtains to implement this (reserve guaranteed bandwidth in case of limited capacity on the interconnecting link, or limit the bandwidth in case of a high capacity on that same link).			1.71	1.73	1.50	0.50	1.00	**1.53**
'6-2'	ST.4.002	That resources at different testbeds are interconnected on layer 2, or that such a layer 2 connection can be automatically created for me (in a way that all the underlying technical details are abstracted for me)			1.57	0.82	1.38	0.00	0.00	**1.03**

3.4.2 Cycle 2 pending requirements (C)

3.4.2.1 Infrastructures community

REQ ID	Description	Comments	CYCLE 2 PRIORITY	N of proposals benefitting from req
I.4.007	Since many of the F4F testbeds are connected to géant, it can be expected that the interconnectivity between testbeds is not representative of the actual interconnections of a realistic deployment using the normal public internet. Therefore experimenters should be able to easily limit the bandwidth available between sites to emulate more realistic settings.		M	**0**
I.4.003	Providers must be able to offer in a transparent way the resources of all the federated testbeds. Interconnectivity solutions should not introduce unneeded complexity in the experiment.	"Solutions based on VPN or other tunnels require that the experimenter is aware of the corresponding configurations when developing the experiment, while he/she should be concentrating on the content of the experiment, and not these practical preconditions. Besides, VPN tunnels will work initially, but they will not scale when a larger number of infrastructures has to be interconnected, due to conflicts in address spaces."	H	**4**

3.4.2.2 Services and Applications community

REQ ID	Description	Comments	CYCLE 2 PRIORITY	N of proposals benefitting from req
ST.4.007	Experimenters must be able to find testbeds whose network connectivity is compatible with the experiment envisioned. In particular, addressing scheme (public/private IPv4, IPv6) and firewall policies must be taken into account.	This is also linked to facility advertising (the testbed providers being able to specify and advertise resources they are offering.)	M	**0**
ST.4.008	Experimenters must be able to use DNS tricks for load balancing. Bandwidth throttling strategies that you can choose from for the given interconnections.		L	**0**
ST.4.009	WAN links/public internet access between testbeds with detailed monitoring for accountability (for testbed provider) so that during an experiment, experimenters can invoke services inside and outside a testbed, moving data between testbeds without it going though his personal machine. Running an experiment on resources provided by different hosting organisations. Information must be retrieved from several testbeds and processed in another one. Also, replicate data between different clouds and redirect requests to the appropriate data centre to retrieve the information.		H	**3**
ST.4.010	That I have monitoring information about networks connections (latency among nodes, number of hops, capacity, ...) before selecting the resources			**0**
ST.4.011	Full mesh of L2 or L3 links between all testbeds			**1**

3.4.3 Cycle 3 new requirements

3.4.3.1 Infrastructure community

REQ ID	Description	Comments	CYCLE 2 PRIORITY	N of proposals benefitting from req
	That I have monitoring information about networks connections (latency among nodes, number of hops, capacity, ..) before selecting the resources			**2**
	Full mesh of L2 or L3 links between all testbeds			**1**

3.5 Feedback from ongoing experiments

The following questions were asked to ongoing experiments:

- How was your experience in general with the federation tools and testbeds?
- What was the hardest part for you in your process of experimenting?
- What would you have liked to have had and was missing? (documentation ok, support ok, tools working, user friendliness ok?)
- Which of the tools/procedures require changes in your opinion and how would you like them changed?

The following table gathers the answers provided by Fed4FIRE experiments:

Table 3: Feedback from ongoing experiments (f)

	#1 How was your experience in general with the federation tools and testbeds?
Experiment 1	Pretty good - testbeds were mostly reliable and suitable for our experiments. In the event of a testbed being unusable, it was documented/made known in advance. JFed was advertised as the only tool we'd need, but its coverage did not include several of the testbeds. As a result, we resorted to using Omni. Two particular partners from University of Bristol were extremely helpful with regards to our questions and incidents when experimenting on the testbeds. Support was in general adequate but these two individuals were particularly helpful.
Experiment 2	JFED is very easy to use and resources booking and experiment setup is made very easy via the direct access on the testbed nodes.
Experiment 3	We found JFed to be a really nice and intuitive tool for setting up the layout of an experiment and to allocate the necessary resources. The Fed4Fire project members were really responsive and helped solving all our issues as soon as possible. Beside automated experiments (simulations) running in the testbeds, we also had on-site experiments in Santander, which required special arrangements (eg. SIM cards with data plans). University of Cantabria partners (Pablo and Jorge) were there to help us with this and we also had the opportunity to get a deeper understanding of the SmartSantander testbed during the on-site experiments.
Experiment 4	The experience is good. That because federation tools and testbeds have provided the most capabilities needed to make our experiment. The highlights capabilities are: - Use of several testbeds at the same time, allows the exploitation of their capabilities. - The creation of the experiments using several testbeds an easy way. - One login (user) allows work with all capabilities of the federation.
Experiment 5	In general, the Fed4FIRE tools provide good and easy experimentation. In our project, the testbeds used in our experiment were BonFIRE, Virtual Wall and PlanetLab. **BonFIRE** • The BonFIRE testbed has the following advantages: o Interconnection BonFIRE- Virtual Wall. o Tools for provisioning, deployment and execution very easy to use and with features that facilitate experimentation.

- The implementation of the designed experiment in BonFIRE is direct. The architecture designed in the design stage has not relevant changes with the architecture truly implemented.
- The interface provided by BonFIRE is user-friendly. Graphical user interface would facilitate more the implementation and experimentation.
- It permits the customization of the resources used.
- The documentation is well organised.

- It has the following disadvantages:
  - EaaS component does not work. This cancels the experimentation of cloud features that should be tested.
  - Cluster does not work.
- After the execution of the experiment, the following results were obtained:
  - The virtual machines worked with its local storage and the obtained performance was good (52Mbps). By using a shared storage, the performance was much worse (2.3Mbps). To deploy applications using large sets of IO operations implies a bottleneck in the shared storage.
  - Using the current shared storage, main cloud computing features cannot be tested.

Virtual Wall

The advantages of using VW are the following:

- It is possible to create network topologies.
- It is possible to customize the network features of the created virtual networks.
- Interconnection with BonFIRE is performs well.
- Reservation and deployment of the nodes is easy and user-friendly.
- It offers the possibility of executing scripts when the requested node starts.
- Good support.
- JFed is a good tool for experimenting.

The main disadvantages encountered are listed as follows:

- It is not possible to customize and to create natural links using JFed between Virtual Wall and the BonFIRE machines.
- Bridge component is not available.
- It is not possible to create and customize virtual machines for reuse.

PlanetLab

This testbed provides a distributed network around the world which facilitates the experimentation in real networks.

- The main advantages of PlanetLab are the following:
- It offers nodes distributed around the world.
- It offers virtualized and exclusive nodes to be used in experimentation.
- It is compatible with the tool NEPI, which allowed us to control and monitor the experiment done in PlanetLab.

	The main disadvantages of this platform are the following: • We think that there is lack of maintenance of the platform since some nodes are labelled as available but they do not really work.
Experiments 6 & 8 (joint feedback in the same interview)	The overall experience within the Fed4Fire testbed is good. Though a good choice of resources is provided in Fed4Fire testbeds, but the collection of resources is required to be increased to have a supermarket effect. The environment is definitely beyond the lab environment. The mailing list has provided a good support to get valuable feedback and resolve problems
Experiment 7	Our experiment relies on the use of more than one Fed4FIRE testbeds, Thus Fed4FIRE tools are required for simplify performance and operation. jFed (v5.1) is an intuitive, user-friendly and easy to use tool. It does a good job at providing a transparent layer between users and testbeds. However, users still need to know before hand about the underlying testbeds in terms of their structures and capabilities in order to configure each testbed correctly. jFed could include more clear descriptions towards defining or pointing towards test beds capabilities. In the positive side as long as you have a good understanding of how different testbeds work, jFed is a great tool to manipulate a small, cross-testbed cluster. OMF looks like a powerful tool but due the running time of our experiment set up we didn't had time to dig into its documentation and tutorial. BonFIRE works eventually fine and the power of Virtual Wall is enough to offer some capacities that are not easy to find on federated tools. In the other hand PlanetLab is not very intuitive to work with. Our Fed4FIRE experiments initially started with BonFIRE and immediately the use experience was like it was really similar to the way I'd been working with Amazon AWS before, which is simple, yet powerful enough for someone who just needed to set up a cluster of dozens of nodes. Virtual Wall is indeed a very good platform with consistent performance and gives you Ubuntu, the most supported Linux distribution by many big data platform providers. Virtual Wall is functional and operative tool except for its by design IPv6 external communication, which is really limited (even Twitter API doesn't support yet IPv6!). PlanetLab on the other the hand, is just capable of replacing local virtual machines with a real network. However the downsides are that almost no control over VM configuration and occasionally get disconnected, which this innestaility transform PlanetLab on a not very reliable platform. It's also about 50-50 chance to get a working VM since not all listed sites are actually available.
	#2 What was the hardest part for you in your process of experimenting?
Experiment 1	Experimental setup and configuration. Documentation for Rspecs vary between islands; some islands were well documented, others not so much. There are subtle differences between Rspecs at different islands too: consistency would help in this case. Broken aggregate managers delayed our experimentation, often for days. AMs are a vital part of the testbed and should be extensively tested prior to experimenters using them.
Experiment 2	We had difficulties in booking nodes with same computational power as well as to do concerted management of wireless nodes resources for specific purposes required in

	some applications, e.g. real time measurement figures for signal strength, background traffic, interference from adjacent access points. That would greatly improve importance of measurements taken on experiment traffic.
Experiment 3	The initial information / documentations did not cover all use cases of jFed, and the special settings necessary for specific testbeds (eg. Bonfire, OMF). However, by communicating with project members via the Fed4Groups experimenter list we could resolve all our problems. We also had some problems with the REST service to SmartSantander for collecting sensor data (some of the REST calls were performing too slow), but we could work around this. As we learnt, the REST service has been reimplemented and the new version performs much better. The new version was in development while we were executing our experiments, so we stayed with the old version and for our purposes (and with the workarounds) it was OK.
Experiment 4	The first contact with testbeds and learn how work each tool. That has conditioned to select the best group of testbed for our experiment and also the design of the full experiments.
Experiment 5	**BonFIRE** • The aggregator interface is not user-friendly. • The experiment was designed to use EaaS and Clusters and those characteristics are not available. Working with EaaS being partially implemented or not-functional at all is having consequences on the experiment planning and scope. **Virtual Wall** • The adaptation of the designed topology in Virtual Wall was complex because the hardware does not provided more than 5 network physical interfaces per node and we had to develop a software solution to emulate it. • To customize the impairments of the network between Virtual Wall and BonFIRE we had to create an alternative solution because this feature is not available. **PlanetLab** • The selection of the nodes because many of them are labelled as available and they are not.
Experiments 6 & 8 (joint feedback in the same interview)	The experiments involve deploying agents which are dependent on specific OS and kernel version. At the moment only the Virtual Wall 1, Virtual Wall 2 and Wi-Lab supported the operating system (such as Ubuntu) required for the installation of the Intelligent protection agents. On the Bonfire test bed we have specifically created a disk image of Ubuntu for the purpose of these experiments
Experiment 7	Our experiment works with big data and cloud processing, by this mean deploying and monitoring a big cluster has always been the most challenging part. Testbed capabilities providing computing capacity and extended storage is always our first choice when choosing where to deploy what for our experiments. At our experiment primarily was use BonFIRE however what is missing is to see some kind of enabler federated platform-as-a-service for popular big data platforms like Hadoop, Spark, Cassandra, etc… so that we as experimenters can start working right away, out of the box. It could be a long way to go to support all kind of platforms out there (not just big data!) fed4FIRE still has a wonderful future to look into these extensions. At our experiments, when doing distributed processing experiments and extraction

	operations, it is needed a common interface offering infrastructure comparison scale so that results comparison between different testbeds can be done, for example, a comparison in between infrastructures like or to Amazon, Rackspace or even to the performance of physical hardware. When it comes to how to correctly measure and evaluate a result, at our experiments it is necessary to know what was exactly the cause of a bad/good result, e.g. the network bandwidth, CPU, disk IO (dedicated or shared in reference to virtual machine)…. As a comparative result for Fed4FIRE we already experienced variations in performance when we deployed our experiments on BonFIRE-Inria and was not possible to identify/know about why the experiment was so slow compared with running the experiments on BonFIRE-EPCC.
	#3 What would you have liked to have had and was missing? (documentation ok, support ok, tools working, user friendliness ok?)
Experiment 1	Bristol's documentation was extensive and very helpful (http://univbrisofeliaf4f.blogs.ilrt.org/user-manual-introduction/). i2Cat's wasn't as good, and in one case incorrect. This was promptly fixed, however. Communication was good between partners, and everyone we spoke to was supportive. JFed needs further work to make it truly useable, especially as it seems to be designed as the main point of interaction between the user and the testbed(s). In our experimentation, Omni had the most support, and thus is the tool we used.
Experiment 2	Some new wireless interface technologies: BLE, BLEmesh, IEEE802.11ac
Experiment 3	As mentioned, documentation was not always up to date or cover all topics, but with the support of project members we could resolve all problems. One specific problem (and a possible future feature of jFed) we had with jFed is that once the testbed resources have been layed out and the experiment started there was no indication when we can expect the experiment to be fully up and running. For example, if there were not enough VMs in Bonfire we had no indication when the necessary number of Vms will be available. Also, it would have been helpful to know from jFed what are the actual capabilities of the testbeds. For example, how many VMs are actually available and so how many of them should we reserve without effecting the experiments of others (i.e. not to allocate too many VMs at once).
Experiment 4	• Full example of the creation and execution of experiment using several testbeds at the same time. • Availability of an incident management where the experimenters can create and know the state of the issues detected in Fed4FIRE. • Chat to resolve urgent doubts and share the know-how with the rest of the experimenters and the technical team of Fed4FIRE
Experiment 5	**BonFIRE** • Virtual Machines compatibility in all the nodes of BonFIRE. For example, Virtual machines created in a node at INRIA cannot be copied and used in other node at EPCC. This is very limiting. • EaaS tested and fully functional since the beginning • More examples in the documentation **Virtual Wall** • The network between the BonFIRE and the Virtual Wall nodes cannot be customized. It would be nice to have this feature. **PlanetLab** • An updated list of available nodes to facilitate their selection and provisioning.

Experiments 6 & 8 (joint feedback in the same interview)	The granular control required for the testbeds such as execution of the scripts on startup is not available through the Fed4FIRE interface. This may constrain the experiment to automatically install the protection agents on the testbed nodes. • Currently we use jFed interface to provision VMs and configure the Rspec to automatically install the Intelligent protection agents on the Fed4Fire nodes. This works well on the Virtual Wall 1, Virtual Wall 2 and Wi-Lab. Bonfire does not use RSpec yet. jFed currently internally converts the RSpec and the SFA calls to bonfire OCCI calls. This conversion is limited, and some features, such as startup commands are not translated, and thus not supported yet in the jFed implementation of OCCI for Bonfire. This limits the experiment to automatically install the Intelligent protection agents on the Bonfire nodes, but we are able to manually install the agents on the Bonfire nodes and manage the security of these nodes.
Experiment 7	BonFIRE is the one that has the most detailed documentation and tutorial so far. It is helpful that it gives enough knowledge to start working, as well as detailed information; the documentation is intuitive and detailed. The way it organizes is also good, easy to read, easy to follow and nothing to overwhelm first-time users. Virtual Wall isn't as good as BonFIRE but still provides solid materials to get me started. PlanetLab's documentation is hard to follow. However, since it's a simple platform, working with it via jFed is pretty straight-forward. jFed (latest v5.3, announced), hasn't been tested, however on the previous version the experience so far was that there were still some levels of inconsistency between different testbeds (for instance, BonFIRE prompted a separate authentication and configuration interface). That testbed-dependent layer puzzles users who don't know what is 'image', 'instance type' or 'sites' in a BonFIRE-specific terminology. The same thing happens to other testbeds. In the positive side jFed usually works smoothly as expected. Bugs happened sometimes but the good backup from support team is excellent. BonFIRE is a special case where it comes with its own tools and Web Portal, which are both user-friendly and powerful. OMF is one of the tools is very promising however by the time of running the experiment there were no time to look into, further investigation can be performed form our experiments in case there is time to do so. The support teams from BonFIRE, Virtual Wall and Fed4FIRE in general have been supportive and continuously in communication and fast in response. Technical experts always provided advices and quickly response to every message/report, both via forums and emails.

	# 4 Which of the tools/procedures require changes in your opinion and how would you like them changed?
Experiment 1	As mentioned previously, it would be beneficial to see JFed developed further, and fulfil its role as the main experimental interface. Documentation can sometimes be hard to find, particularly for some islands (EPCC for example). A well-developed tutorial would aid in getting acquainted with the testbed. This process otherwise consumes a huge amount of time which could be better use for experimentation rather than familiarisation.
Experiment 2	Maybe an integrated way of extracting measurements. Today we resort on user-supplied scripts.
Experiment 3	We consider jFed as the central tool for experimenters to access Fed4Fire, so naturally jFed should be (and we guess is) developed further. It would be great if the whole lifecycle of an experiments (or most of it) could be controlled and supervised from within a single tools. One specific feature that would be really handy is some way to hand over control from experiment setup to actual experiment execution. One problem we had, for example, is that in jFed we could only specify name for an experiment, however other tools, like OMF need an "experiment ID". This ID is not available from jFed. In our case, we had to use specific Bonfire API calls to translate experiment names to experiment IDs, so that we could start the experiments via OMF. It would also be nice if in jFed we could define setup scripts that are run on the VMs after they are started as part of the experiment. These scripts could configure the VMs, download/update necessary packages, etc.
Experiment 4	The documentation should have version control and a automatic mechanism to inform to experimenters of new version and a summary of changes.
Experiment 5	To provide an API to communicate with JFED. This would be nice to control and monitor an experiment without having to manually select, provision and deploy the nodes of Virtual Wall. With this solution, a complete experiment using different testbeds could be automatically controlled, for example with a script using NEPI. Sometimes, when the experiment is Terminated, it can't be deleted from the Portal, although the message "Experiment deleted" appears. Sometimes the VMs created in an experiment are always in pending state (can be because the testbed is currently full, or because we can't deploy more resources, for example). We can see the availability of the testbeds via URL, but it would be very useful to show a popup or alert after creating it to explicit why is not yet active.
Experiments 6 & 8 (joint feedback in the same interview)	• The resource catalogue currently provided is too high level and more granular catalogue may be useful • Reservation of nodes is essentially required when the scale-up test within the experiment is to be performed where all the resources across the testbeds will be required • Allow to define the storage capacity for the Nodes.
Experiment 7	Because we use different Fed4FIRE testbeds mostly to replace our limited in-house servers to deploy a large scale cluster, our primary concern is the way to scalability: deploying software, monitoring indicators, setting up network, rescaling/reconfiguring cluster… Performance has a trade off between friendly user interface and efficiency on command line APIs. (BonFIRE CLI for instance is more efficient than GUI tools. However it would be great if testbeds support popular tools such as Puppet, Chef or Vagrant out of the box. These tools already have huge communities and great documentations. Experimenters don't need to learn 'yet another cluster management tool/language' so that they can focus on their own work.

	Speaking of Fed4FIRE as federated environment, to see a more unified interface between different testbeds in such a way that user don't need to actually know about how each testbed works would be an asset. An overview understanding of capabilities and limitations of each testbed would be sufficient for any user to get started in this respect.
	# 5 Any other comments you may have
	• Improving the federated identity management aspects of Fed4FIRE service layer by techniques designed to be federated identity management protocol agnostic, so that different FIM systems can be plugged in such as OpenID, Oauth, SAML, PKI, Kerberos etc.

4 Conclusions and next steps

This document gathers the requirements for the third iteration of the Fed4FIRE federation architecture. Requirements have been derived from a combination of the requirements table existing in cycle 2 and a survey answered by our experimenters' community. The requirement coverage map resulting from cycle 2 was reported in "Fed4FIRE 318389 D2-4 Second federation architecture". Since the survey was built out of requirements partially or not tacked in cycle 2, it only focuses on pending functionality or features. The results are organised in two different blocks of requirements evaluated differently:

- For requirements included in the survey we have an evaluation from different experiments and proposals presented to our open calls.
- For requirements coming from old cycle 2 and not existing in the survey, we have analysed which proposals presented to the 2nd Open Call would benefit from them.

We have presented the results in section 3.

The result is a total of:

- 75 requirements for Experiment Workflow and Lifecycle Management,
- 31 requirements for Measurement and Monitoring and
- 15 requirements for Interconnection

The first set of requirements has been ordered according to their average value including all the answers gathered from our community. However, we have also highlighted with a different colour which segment within our community has valued each requirement most. This is important for technical workpackages to evaluate the importance of each requirement, also taking into account that each segment has a different number of participants having answered the survey (as explained in the introduction of section 3).

We conclude that, in general, the community of experimenters coming from the 1st Open Call, who already have an important background as far as interaction with the federation is concerned since they have been using the federation services and tool for some time, have provided the highest marks when evaluating the requirements included in the survey. However, the most important community in terms of the number of experimenters having answered the survey would be the experimenters having presented proposals to the 2nd Open Call, which has been separated in Industry and Academia.

The survey results highlight some of the most important requirements for our community such as, for example, the uniformity of resource information for resources belonging to different testbeds as far as their presentation is concerned, the availability of orchestration tools to ease the interaction

with the federation or the possibility to have detailed monitoring and eventual trouble information during experimentation.

When looking at the answers provided to this open questionnaire by these experienced experimenters, we find important insights concerning how they find the interaction with the federation. For example, they seem to greatly appreciate support and guidance when setting up the experiments as well as available documentation and they would like the JFED tool to be extended with further functionality.

As for requirements coming from the old table of requirements existing in cycle 2 not included in the survey (mostly Medium or Low priority requirements in cycle 2), we confirm that these remain low priority requirements in general, except for ST.4.009, which was and still is a high priority item.

The survey used in cycle 3 has been modified recently in order to make it shorter. This survey will be circulated among the winners of the 2nd Open Call and 2nd SME Open Call in order to gather more feedback from which we will obtain future recommendations for the federation.

Work regarding requirements does not stop at this deliverable even if cycle 3 is the last one foreseen for Fed4FIRE architecture. We are currently distributing a simplified version of the survey among for proposals presented to the 2nd SME Open Call and will collect further inputs from ongoing experiments in order to gather further recommendations for the future operation of the federation.

5 References

[1] VCOC experiment. http://www.bonfire-project.eu/innovation/virtual-clusters-on-federated-cloud-sites

[2] SILICON experiment. http://www.bonfire-project.eu/innovation/silicon

[3] http://www.fitman-fi.eu/

[4] Simon Vocella, Álvaro Monje, Celia Velayos, Fabio Farina, Chrysa Papagianni, Mauro Campanella and Vassilis Maglaris." Complex federated slices provisioning through SFA. The FEDERICA and NOVI joint experience ".https://tnc2013.terena.org/core/presentation/33

[5] Fed4FIRE "D4-1 First input from community to architecture"

[6] Fed4FIRE D2.2 "First sustainability Plan"

[7] Fed4FIRE "D3-2 Infrastructures community federation requirements, version 2"

[8] Fed4FIRE "D4-2 Second input from community to architecture"

ANNEX 1: Functional requirements questionnaire

The goal of this part of the survey is to get a feeling of the requirements that your experiment imposes on the Fed4FIRE federation of testbeds. For the listed requirements we are mainly trying to prioritize requirements that are already on our radar, based on what our potential experimenters really need. Next to those requirements, we are very keen to receive any new requirement that you can think of that also needs to be fulfilled when supporting your experiment. For this we have created the possibility to add as many new requirements as you see fit.

The questions of this part of the survey are presented in different tables, clustered around the different steps that an experimenter has to go through when running an actual experiment. In every of those tables, the Priority column should be filled in as follows:

X = no opinion or not applicable to your experiment/ environment

1=not required 2=nice to have 3=important 4=must have

Requirements related to resource discovery

The requirements listed in this table are all related to the very first thing that an experimenter does: learning about the different testbeds, and about which specific resources that they can offer.

	When discovering the different resources that Fed4FIRE can offer me for my experiment, I require …	Priority (X or 1-4):	Comments and further details
1-1	That I can browse some kind of resource catalogue to look for appropriate resources on a high level. Such a catalogue is limited to information such as: testbed X is a testbed for WiFi experiments in an office environment, testbed Y is a testbed for testing cloud applications, etc.		
1-2	That Fed4FIRE provides a detailed view on what node capabilities are available on every testbed of the federation (e.g. mentioning information for every resource of a testbed regarding CPU speed, RAM, supported 802.11 technology, optical networking interfaces, etc).		
1-3	That the above view on node capabilities is the same across the different testbeds of the federation. This means that when describing the characteristics of resources, all testbeds should adopt the same units (e.g. represent RAM always in MB, and not sometimes in MB and sometimes in GB) and use the same parameter names for aspects that mean the same (e.g. always talk about “RAM”, and not “RAM” on some testbeds, “working memory” on some others and just “memory” on a third group of testbeds).		
1-4	That next to browsing through information about what is		

	available, that I can actively search for the existence of resources with certain characteristics by defining a specific query (e.g. something that is similar to an SQL query, e.g. select resources from all testbeds where RAM >= 8 GB)		
1-5	That I know the location of the site where resources are located. Per site, this location information can be exactly the same for all resources.		
1-6	That for nodes that have static network connections to other nodes in the same testbed, that it should be possible to identify the corresponding physical topology. In the wired domain this means that you can know how the nodes are connected to each other. For wireless resources this means that you know which resources are in transmission range of each other.		
1-7	That I have accurate location information about the actual resources that I will use (1 m accuracy), typically important for wireless nodes.		
1-8	For virtual resources, that I know their physical host and the actual location.		
1-9	That I can assess which testbeds/resources are more reliable then others (both in terms of provided hardware, software, and wireless interference, possibly based on historical health information about the resources and their environment)		
1-10	*If you have any additional requirements regarding resource discovery, please insert them here. Create as many new rows in this table as needed.*		
1-11			
1-12			
1-13			

Requirements related to resource selection and reservation

Once an experimenter has learned which resources are available at every testbed, he/she can then design its experiment appropriately. When setting up the corresponding experiment, the first thing that needs to be done is selecting resources to be included in the experiment, and reserving them for the experiment for a certain moment in time.

	When selecting and reserving resources that I want to include in my Fed4FIRE experiment, I require …	Priority (X or 1-4):	Comments and further details
2-1	That when browsing through the resource descriptions, that I can manually select every node that should be added to my experiment. Think of an experience similar to online shopping and putting resources in your shopping cart.		
2-2	That I can select suitable resources for inclusion in my experiment by defining a specific query (e.g. something that is similar to an SQL query, e.g. select all resources from Virtual Wall where nr_ethernet_cards >= 6)		
2-3	That I can temporarily install my own equipment at a Fed4FIRE testbed for testing, and select it to be included in my experiment.		
2-4	That the mechanism for registering my own equipment at a testbed is standardized, allowing me to register that equipment at different testbeds in exactly the same manner.		
2-5	That I can reserve resources. It is OK for me that they are shared with others (soft reservation, e.g. requesting a virtual machine that will be deployed on a physical server that is used by other experiments also), as long as I know that I will also have guaranteed access to them.		
2-6	That I can reserve resources. They have to be exclusively assigned to me (hard reservation, e.g. reserving a virtual machine that will be deployed on a physical machine that is dedicated to your experiment only)		
2-7	That next to adding resources to my experiment right now (instant reservation), that I can also define a reservation for any moment in the future (future reservation, e.g. tomorrow from 9AM-5PM).		
2-8	Those situations are avoided where a have to wait days or weeks before being able to use the testbed because of long reservations of others.		
2-9	That I can reserve nodes exclusively for myself for a longer period (days or weeks)		
2-10	That a reservation is approved or rejected quickly (within a few minutes).		

2-11	That I can easily reserve resources across multiple testbeds using the same common tools. These should also be as user-friendly as possible, abstracting the complexity of the underlying infrastructures for me as much as possible. This way I can focus on the experiment design itself instead of learning how to work with numerous testbed-specific tools.		
2-12	That when reserving resources across multiple testbeds, that there is guidance in finding the first appropriate time when all the resources that I want across the testbeds would all be available.		
2-13	That I can use a single Fed4FIRE account to select and reserve resources at all different testbeds of the federation. So even when using one common tool for reservation at the different testbeds, I don't want to remember a different username/password combination for every testbeds, and I also don't want to register again at every testbed that I want to use. Of course, registering for that one Fed4FIRE account should also be straightforward.		
2-14	That if testbeds decide to assign me a certain reservation quota (e.g. based on my profile such as student, post-doc, professor, paying customer, etc), that I can request a temporary increase of my quota if really need it (e.g. before a paper deadline)		
2-15	That the testbeds and/or the federation guarantee a certain Service Level to me regarding the execution of my experiment (availability of resources, reliability of resources (uptime/downtime), responsiveness of support services, privacy guarantees, etc).		
2-16	That I can dynamically scale my resources up and down according that what my experiment needs during its execution. For instance if a server deployed on a VM gets overloaded, I should be able to assign more resource (RAM, CPU cores, etc.) to that running VM, and/or should be able to add a second VM to my running experiment on which I deploy a second instance of that server.		
2-17	That if I reserved a number of resources at a testbed, that I can divide them over different independent experiments that I am doing at the same time. It should be possible to easily address/group the resources from one experiment.		
2-18	*If you have any additional requirements regarding resource selection and reservation, please insert them here. Create as many new rows in this table as needed.*		
2-19			

Requirements related to using the resources (deployment and basic usage)

Once an experimenter has added the resources to the experiment, the next step is the deployment of those resources for that experiment, and basic usage of the resources. This section tries to capture the corresponding requirements.

	When using the resources that I included in my Fed4FIRE experiment, I require …	Priority (X or 1-4):	Comments and further details
3-1	That I can SSH to my nodes.		
3-2	That I have root access to my nodes. This allows me to perform any action on the nodes that I want (install new applications, device drivers, load additional kernel modules, etc).		
3-3	That I can use a single public/private SSH key pair to access my resources on all the different testbeds		
3-4	That I can choose to have Windows installed on my nodes		
3-5	That I can choose to have a specific Linux distribution on my nodes (e.g. latest Ubuntu LTS release)		
3-6	That I can choose to use a custom Linux kernel on my nodes (e.g. with my own performance upgrade patches to the kernel)		
3-7	That my nodes can download and install software from the Internet (e.g. using a package manager)		
3-8	That I can take a binary image of the hard drive of my nodes, and that I can store these for later re-use (so flashing the image back later on)		
3-9	That I can define what a node should automatically do at start-up (bootstrap scripts)		
3-10	That during the deployment of my resources over different facilities, that my initial data sets can be automatically loaded to all these resources.		
3-11	That I can allow other people of my work team that are involved in the experiment to use the resources that I have reserved and deployed. I should be able to specify which resources should be shared, and which not.		
3-12	That I can easily use my resources across multiple testbeds using the same common tools. These should be as user-friendly as possible, abstracting the complexity of the underlying infrastructures for me as much as possible. This way I can focus on the experiment itself instead of learning how to work with numerous testbed-specific tools.		
3-13	*If you have any additional requirements regarding resource usage, please insert them here. Create as many new rows in this table as needed.*		
3-14			

Requirements related to orchestrated control of the experiment

In the previous step resources were deployed, and the experiment can manually log in on them and control what they should do. However, when aiming to perform more advanced scenarios, where many resources are included and all of them should be triggered to perform certain task at the appropriate time, more orchestrated experiment control is needed. The corresponding requirements are captured in this section.

	When controlling the execution of my experiment in an orchestrated manner, I require …	Priority (X or 1-4):	Comments and further details
4-1	That I can define the behaviour over time of a distributed experiment in a single script, which can be started automatically at any desired moment, and will be automatically translated to the corresponding triggers at the nodes at the appropriate time. So e.g. describing in a single script that the 5 client nodes in an experiment should gradually increase their load on the server that they are testing in the experiment. This will be done automatically, without the experimenter login in to these 5 nodes and gradually increasing this load manually.		
4-2	That I can define the behaviour of a distributed experiment in a single script, based on events (e.g. value above threshold). This can be started automatically at any desired moment, and will be automatically translated to the corresponding triggers at the nodes at the appropriate moment. So e.g. describing in a single script that a server should scale up to a VM with more CPU power and RAM when the load of the clients on the server becomes higher than a certain threshold.		
4-3	That the description of the above orchestration is described in a human-readable way. This description should also be uniform across the different testbeds.		
4-4	That the above description of the orchestrated control of the experiment can also include other aspects that will be performed automatically. This includes selection, reservation and deployment of resources; monitoring of the resources and collection of measurement data during the experiment.		
4-5	*If you have any additional requirements regarding orchestrated experiment control, please insert them here. Create as many new rows in this table as needed.*		
4-6	*Real-time results*		
4-7			
4-8			

Requirements related to the results of the experiment (monitoring and measuring data)

The motivation for every experiment is to learn something. For this it is needed that the appropriate monitoring data and experiment measurements are captured. This section grasps the corresponding requirements.

	When capturing the results of my experiment (monitoring and measuring data), I require …	Priority (X or 1-4):	Comments and further details
5-1	That the internal clocks of resources across multiple testbeds are synchronized very accurately		
5-2	That Fed4FIRE makes it easy for me to retrieve and store data that I measured during the runtime of the experiment. This means that it should be easy to store my measurement somewhere in a way that the data is clearly related to the experiment ID, but without needing to establish connections to certain databases manually from within my code, and without needing to know the specific experiment ID that belongs to my current experiment.		
5-3	That by default some common characteristics of my resources are stored automatically for later analyses during experiment runtime (CPU load, free RAM, Tx errors, etc).		
5-4	That for the above monitoring, that I can select and configure how this data should be collected (always at a specified interval, only after a certain event or alarm, define some specific filters, etc).		
5-5	That I can request the monitoring solutions to provide me specific additional on-demand measurements of node characteristics to ease experiment development and debugging		
5-6	That information about external wireless interference during the execution of my experiment is automatically provided for me.		
5-7	That the overall health status of the different testbeds (testbed up or down, has free resources left, etc.) is continuously monitored by the federation, and that in case of issues I am informed of this.		
5-8	That the overall health status of the different testbeds (testbed up or down, has free resources left, etc.) is continuously monitored by the federation, and that in case of issues the corresponding testbeds try to solve them asap.		
5-9	That other aspects related to the successful execution of my experiment are continuously monitored, and that I am automatically informed in case of any errors. Examples are: when a selected resource could not be instantiated, when there is a problem with the interconnectivity between the used testbeds,		

	When capturing the results of my experiment (monitoring and measuring data), I require …	Priority (X or 1-4):	Comments and further details
	when a used testbed goes down during the experiment, when there is a sudden peak of wireless interference, etc. This might be important when analysing anomalies in the experiment results.		
5-10	That when an error requiring manual intervention is reported to me as part of the previous step, that I am guided through the process for recovery.		
5-11	That the overhead of any monitoring and measurement tool is minimal. These tools should have a negligible impact on the results of my experiment.		
5-12	That I can store and access my experiment monitoring data and other measurements on a data service on the federation, which is accessible during the experiment (temporarily data storage by the federation)		
5-13	That I can store and access my experiment monitoring data and other measurements on a data service on the federation, which is also accessible after the experiment (archiving of historical data by the federation)		
5-14	That access to my stored data is properly secured. Experiments must be kept confidential if required, the privacy of experiments, data sets and results should be guaranteed.		
5-15	That I can store experiment configurations in order to repeat experiments and compare results of different runs		
5-16	That I can share my stored data with specific others (individuals and/or groups), or even make them publically available		
5-17	That I am made aware if my storage capacity is running out.		
5-18	*If you have any additional requirements regarding monitoring and measuring data, please insert them here. Create as many new rows in this table as needed.*		
5-19			
5-20			
5-21			

Requirements related to the interconnectivity of the different testbeds

Fed4FIRE facilities are intended to allow experimentation with Future Internet techniques. And because Fed4FIRE is a federation of testbeds that enables experiments that included resources from different testbeds, the interconnectivity between the different testbeds is very important. This section enumerates the corresponding requirements.

	When focusing on the connectivity of the resources that will be included in my Fed4FIRE experiment, I require …	Priority (X or 1-4):	Comments and further details
6-1	That resources at different testbeds are interconnected on layer 3 (IP)		
6-2	That resources at different testbeds are interconnected on layer 2, or that such a layer 2 connection can be automatically created for me (in a way that all the underlying technical details are abstracted for me)		
6-3	That I can know the type of interconnections that are available between the testbeds (layer 2 and/or layer 3, NAT or VPN included, dedicated direct link, connected through Géant with or without bandwidth reservation, connected over the public Internet, …)		
6-4	That I can configure a specific bandwidth on the interconnections between the different testbeds used in my experiment. As long as the links behave as configured, I don't really care what the testbed has to do behind the curtains to implement this (reserve guaranteed bandwidth in case of limited capacity on the interconnecting link, or limit the bandwidth in case of a high capacity on that same link).		
6-5	That my resources are directly reachable, without any network address translation (NAT) or virtual private network (VPN) in between. So actually I require that all resources have a public IPv4 or IPv6 address.		
6-6	That if an issue arises with the interconnection between my used testbeds, that I am automatically informed about this.		
6-7	*If you have any additional requirements regarding monitoring and measuring data, please insert them here. Create as many new rows in this table as needed.*		
6-8	Bandwidth throttling strategies that you can choose from for the given interconnections.		
6-9			
6-10			

ANNEX 2: Additional feedback from ongoing experiments (f)

The following questions were informally asked to the ongoing experiments in order to get their impression concerning the testbeds, tools and services offered by the federation:

1. How was your experience in general with the federation tools and testbeds?
2. What was the hardest part for you in your process of experimenting?
3. What would you have liked to have had and was missing? (documentation ok, support ok, tools working, user friendliness ok?)
4. Which of the tools/procedures require changes in your opinion and how would you like them changed?