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**

D7.3 – Report on first cycle developments regarding trustworthiness

Work package	WP7
Task	Task 7.2, T7.3, T7.4
Due date	31/03/2014
Submission date	06/05/2014
Deliverable lead	Steve Taylor (IT Innovation)
Version	1.0
Authors	Georgios Androulidakis Loic Baron Pedro Rey Estrada Steve Taylor
Reviewers	Wim Vandenberghe (iMinds) Mark Sawyer (EPCC)

Abstract	This deliverable reports on the progress made in WP7 for the first cycle of Fed4FIRE. Most of the cycle 1 requirements (described in D7.1) were not demanding. This enabled us to concentrate on early developments for cycle 2, and this is where the bulk of effort was spent in the time from D7.1 to D7.2 (months 5 – 16). As a result, preparatory work has already been done for cycle 2, and this is described in D7.2. The other contribution that this deliverable makes is to provide answers to the questions raised at the end of D7.1. We now know answers to most of these, based on the work we have done in the period between delivery of D7.1 and the current time. Many of the questions are addressed by the PDP work implemented up to PM18 and described in D7.2. The main reason for this is that the questions actually represent issues that needed to be resolved in order to make progress in the area of federation authorisation. Thus the work done on the PDP, and the discussion in the project around this subject were the main sources of answers to the questions.
Keywords	Trust, Reputation, SLA, Security, Certificate, Credential, Access Control

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

This deliverable reports on the progress made in WP7 for the first cycle of Fed4FIRE. Most of the cycle 1 requirements (described in D7.1) were not demanding. This enabled us to concentrate on early developments for cycle 2, and this is where the bulk of effort was spent in the time from D7.1 to D7.2 (months 5 – 16). As a result, preparatory work has already been done for cycle 2, and this is described in D7.2.

The other contribution that this deliverable makes is to provide answers to the questions raised at the end of D7.1. We now know answers to most of these, based on the work we have done in the period between delivery of D7.1 and the current time. Many of the questions are addressed by the PDP work implemented up to PM18 and described in D7.2. The main reason for this is that the questions actually represent issues that needed to be resolved in order to make progress in the area of federation authorisation. Thus the work done on the PDP, and the discussion in the project around this subject were the main sources of answers to the questions.

Acronyms and Abbreviations

AM	Aggregate Manager
Authn	Authentication
Authz	Authorisation
CA	Certificate Authority
CMS	Content Management System
CRUD	Create, Read, Update, Delete
CSR	Certificate Signing Request
DN	Distinguished Name
FRCP	Federated Resource Control Protocol
GID	Globally Unique Identifier
GUI	Graphical User Interface
HRN	Human-Readable Name
IdP	Identity Provider
MOS	Mean Opinion Score
OMF	Orbit Management Framework – a reference implementation of FRCP
OML	ORBIT Measurements framework and Library
PDP	Policy Decision Point
PEP	Policy Enforcement Point
QoE	Quality of Experience
QoS	Quality of Service
RAG	"Red-Amber-Green" – status indicators for components operational state for first level support
RC	Resource Controller (in OMF)
REST	Representational State Transfer
ROCQ	Reputation, Opinion, Credibility, Quality
SA	Slice Authority
SFA	Slice Federation Architecture
SLA	Service Level Agreement
URN	Uniform Resource Name
UUID	Universally Unique Identifier

Table of Contents

1 Introduction

This document is a report on developments in cycle 1 of Fed4FIRE in WP7. It is necessarily short, as most of the cycle 1 requirements (described in D7.1) were not demanding, and this enabled us to concentrate on early developments for cycle 2, and this is where the bulk of effort was spent in the time from D7.1 to D7.2 (months 5 – 16). The result is work we have already done in preparation for cycle 2, and this has been described in D7.2. As a result, this report is as short as D7.2 is long, and in it we briefly describe work done in cycle 1:

- Certificate directory
- SLA management
- Trust and reputation

The other contribution of this deliverable is to provide answers to the questions raised at the end of D7.1. We now know answers to most of these, based on the work we have done in the period between delivery of D7.1 and the current time. This report contains a section containing the questions from D7.1, with answers.

2 Certificate Directory

The certificate directory is a federation-wide store of root certificates that can be consulted by users, testbed providers and tools. In many cases, testbed providers act as their own CA by having a self-signed certificate that is used to sign other certificates for users of the testbed. The certificate directory is a single place to store all testbed providers' "CA" certificates.

As described in D7.1, we decided to implement such a directory using a simple apache web server. The certificates of the different testbeds are stored in a folder and accessible through a URL on the Fed4FIRE portal.

https://portal.fed4fire.eu/certificates/

Index of /certificates

Name	Last modified	Size	Description
Parent Directory		-	
README	02-May-2013 17:37	40	
fed4fire.gid	20-Jun-2013 11:43	778	
fuseco.cer	06-Jun-2013 11:19	3.2K	
nitos.gid	20-Jun-2013 11:43	757	
ocf.gid	03-Jul-2013 11:31	1.0K	
ofam.gid	03-Jul-2013 11:31	2.1K	
omf.gid	03-Jul-2013 11:31	757	
ple.gid	06-Jun-2013 10:40	757	
smartsantander.gid	03-Jul-2013 11:31	802	
wall2.cert	03-Oct-2013 18:15	1.6K	

Apache/2.2.22 (Debian) Server at portal.fed4fire.eu Port 443

Any testbed's software, such as an Aggregate Manager (AM) can access this URL and add the relevant certificates to its trusted roots. Therefore any user presenting a credential issued by one of these roots of trust will be able to access federated testbeds. Which certificates to trust is a decision for individual testbeds.

3 SLA Management

During cycle 1, the main activities concerning SLAs have been centred around gathering requirements, identifying the most suitable implementation in the framework of Fed4FIRE (distributed/centralized) and studying the modifications required in order to adapt the SLA tool in order to meet Fed4FIRE's approach. The exercise of finding a suitable and simple SLA inspired in commercial cloud offerings (e.g. Amazon) was also done during this period.

The task of gathering requirements from the testbeds had a main objective: to understand what the SLA management should do in Fed4FIRE. For this, the participation and involvement of the testbeds was considered to be the most important factor since SLA management is usually associated to something beneficial for the customer and not for the providers (testbeds in our case). SLA Management provides an interesting source of information concerning self-performance which can be used by the testbeds to protect their facilities.

In order to retrieve information and gather these requirements, a round of interviews with every testbed provider took place to understand their expectations from Fed4FIRE SLA management.

The results of these interviews were that a proportion of testbeds will not operate or intend to operate commercially. These testbeds offer resources and/or services on a best effort basis and do not have SLA mechanisms currently in place. However, other testbeds are potentially interested in offering commercial services, and testbeds in general need to manage their resources – all of which SLA management can help with. A key finding was that the more well-used (i.e. successful) a testbed was, the greater the resource management challenges they faced because demand outstripped supply.

In relation to the architecture of the SLA, different approaches were analysed and they were:

- **Federated centralized SLAs**: An SLA is agreed between the experimenter and the federation as an entity. The federation deals with underlying testbeds as a sort of SLA broker.
- **Federated SLA setup**: An SLA is agreed between the experimenter and every single testbed involved in an experiment (supposing they all have adopted SLAs). The federation ensures there is an SLA established with every testbed before the experiment begins.
- **Federated SLA tool**: An SLA is agreed between the experimenter and every single testbed involved in an experiment. The federation does not ensure there is an SLA established with every testbed before the experiment begins. Every testbed having adopted SLAs will expose its SLAs and will manage its SLA commitments individually. With this approach, the federation only provides the set of tools for a testbed to implement SLA management. It's up to the testbed to use it or not.

The preferred option was "Federated SLA tool" since it involved a thin federation layer, more flexibility to the testbeds to adopt SLA management.

Finally, in terms of technology baseline, WS-Agreement specification was selected as the most appropriate for creating and enforcing SLAs and REST interfaces were selected for the communication between the SLA elements for their simplicity, flexibility and control.

4 Trust and Reputation

The reputation service in Fed4FIRE aims to provide mechanisms and tools towards building trustworthy services based on the combination of Quality of Experience (QoE) and monitoring data. The developed mechanisms and tools will reflect the end users' (experimenters') perspective with the objective of empowering the users/experimenters to select testbeds based on dynamic performance metrics. These metrics will offer a "smart" user support service that provides a unified and quantitative view of the trustworthiness of a facility. In order to achieve that, the service will mainly focus on building reputation-based trust utilizing:

1. Raw monitoring data (e.g. information to experimenters on up-time, usage etc. that results into site popularity) and SLA information
2. Users' feedback regarding their Quality of Experience (QoE) and service received.

During the first cycle of the Fed4FIRE project, the work performed mainly concentrated on the identification of:

a) a suitable framework to be used in a federated environment,
b) the modifications and enhancements necessary for the successful adoption of such a framework in order to meet Fed4FIRE's approach and goals, and
c) the components that have to be developed as well as the main interactions of the Reputation service with the other Fed4FIRE components.

As a first step, in order to gather requirements and testbed owners' expectations and needs of the reputation service, the testbeds were surveyed using a questionnaire. The questionnaire contained questions about the metrics that testbeds are planning to monitor and whether the existence of such a service was in the interest of them. The results revealed that most testbed owners wish to have their testbed evaluated and that the reputation service provides an added value with regards to its trustworthiness. Moreover, from the questionnaire, we were able to gather information about the monitoring data that will be provided to the reputation service.

As documented in D7.1 and D7.2, the ROCQ algorithm was selected as the basis of the reputation-based trust mechanism, to be implemented and adopted within the Fed4FIRE framework. In order to test its applicability, we initially simulated a single testbed environment and conducted several simulated experiments. Some experiments also included an injected sample from different categories of malicious users providing false feedback. Subsequently, we identified the necessary modifications for the single testbed reputation mechanism to be applied in the federated environment of testbeds. Moreover, we designed the template for the feedback that will be requested from the user upon the completion of an experiment and decided on the technical and non-technical questions from which the template will be comprised of.

Moreover, the interactions of the Reputation Service module with the components of the Fed4FIRE architecture have been decided, namely with the:

- Portal
- Future reservation broker

- Infrastructure Monitoring Data Broker[1]
- SLA Service

[1] Described in D2.4

5 Answers to Questions from D7.1

This section contains the list of questions from the conclusions of D7.1, together with answers based on discoveries and lessons learned up to the current time. The questions from D7.1 are repeated here, interspersed with answers.

As a side note, it can easily be seen from the questions and answers that many of the questions are addressed by the PDP work implemented up to PM18 and described in D7.2. The main reason for this is that the questions actually represent issues that needed to be resolved in order to make progress in the area of federation authorisation. Thus the work done on the PDP, and the discussion in the project around this subject were the main sources of answers to the questions.

1. **How will the user actually use the X.509 certificates? Will they need client side software? Alternatively can they use their web browser?**

The SFA uses certificates to identify all entities – not only users, but also servers, resources, slices and authorities. A user registers with a membership authority, who creates a public private key pair and certificate for them.

There are different answers to this question, depending on what the users are trying to do, and what tools they are using, and these options are discussed below. Fed4FIRE has deliberately provided compatibility with a number of client-side tools, so as to give the user a choice, and the relevant tools are discussed here below.

- If the user is trying to request resources, they can use e.g. myslice and this permits them to identify themselves using their certificate. To request resources, a user needs a slice, which is issued by a Slice Authority.
- A user can use their certificate to request a slice from a Slice Authority. Often (as in the case of the Virtual Wall) the membership authority and slice authority are co-located, so the user can login and request a slice. The tool for this is a web-portal provided by the Slice Authority (for example the portal offered by the Virtual Wall, where users can request slices and manage them).
- The Fed4FIRE portal simply requires a login and it is anticipated that in future, a lot of the complexity of using Fed4FIRE will be hidden by this.
- When the user is using resources to run an experiment, they can do this in mainly two ways:
    - Use SSH keys – this requires the user to pass their public key to the AM on the service provider, the AM to link this with the certificates representing the user, and the AM to put the public key on the correct resource, so the user can SSH login.
    - Use the OMF Experiment Controller (EC) to interact with OMF Resource Controllers (RCs) at testbeds. The EC provides the opportunity to specify the certificate and private key they want to use when connecting to the RC.

Rights on slices are represented by signed XML documents, using the schema of the so-called “slice credential”[2]. The slice itself is identified with a certificate, as is the owner of the slice, and textual

[2] http://groups.geni.net/geni/wiki/GeniApiCredentials

representations of the certificates are included within the slice credential. The whole credential is signed by the authority that issued the slice.

A slice owner can delegate access to resources in their slice to another user, and they do this by submitting a request for a delegated slice credential to the slice authority and including the delegate's certificate. The result is a signed slice credential that names the delegate as an authorised user of the slice.

2. It is not clear how the X.509 certificates will map to SSH logins used in many testbeds.

This is done by the SFA-compliant Aggregate Manager (AM) at the testbed. The user requests resources using a certificate to identify themselves (and supplying their SSH public key), and the AM decides whether to give them any resources. If the AM grants the request, it puts the user's SSH public key onto the resource allocated to the user, and the user can login.

3. What happens at the testbeds with the SFA X.509 certificates? How are they authenticated? What SFA support is there to do this?

At resource request time, authentication of certificates at the SFA level is done by standard cryptographic mechanisms – based on verification of the validity of the certificate (i.e. does the issuance of the certificate go up to a root Certificate Authority trusted by the testbed?).

At experiment control time, SFA does not provide support for using certificates to access resources. The project has provided a federation authorisation system that links the technologies used for resource request and experiment control (described in D7.2) and this supports the use of certificates to assert the identity of the user.

4. An authorisation decision about whether a user should get access to resources in a testbed should only be made by the gatekeeper protecting that testbed. It is unlikely any testbed will wish to permit any other party to have the power to make an authorisation decision on behalf of a testbed.

Rather than a question, this is an assertion made at the time of writing D7.1, and the core principle of giving testbeds full control of their resources in a federated situation has guided the development of the PDP (Policy Decision Point – described in D7.2). This aims to protect FRCP resources, and each testbed that wants to use FRCP can choose to install their own PDP, and each PDP will be under the full control of the testbed's administrator.

5. What authorisation decisions will need to be made by testbeds and how will this relate to "federation authorisation"?

The work on the PDP (described in D7.2) provides one way of answering this question. It is based on the realisation that resources acquired using SFA need to be accessible securely using FRCP, and that there was a gap between the two. The PDP section of D7.2 describes how bridging this gap was achieved, and the chosen mechanism for this was the SFA slice credential. The slice credential is chosen because it is a well-known mechanism already existing in the SFA domain, so is well understood and supported by SFA people and tools. The idea in designing and implementing the PDP was to enable users to use this same token to use resources as well as reserve them – if a user could

get resources allocated to a slice via a slice credential, then they should be able to use this same slice credential when they want to use the resources via FRCP.

The actual access policy is based on the AM at a testbed being trusted to tell the PDP which resources it has allocated to a slice, and that the owner of the slice can use those resources. When a user requests access to a resource, they present a slice credential, and the PDP checks which slice the resources should be in, whether the slice credential is actually for that slice, and whether the user is the owner of that slice. If all this is true, then the user is permitted access, otherwise they are denied.

> Note: the above is the simplest access control policy, and other more advanced ones are possible. For example, a resource may be available for a finite time, and this can be incorporated into the access control policy. Another example is that the owner of a slice can authorise delegates (other users) to use their slice, and the PDP can enforce this. The PDP has been designed so that it is flexible, so it should be able to accommodate other access policies as well.

The concept of "federation authorisation" is founded in interoperability – the ability to reserve resources across multiple testbeds and use the same access token across all of them. A slice can hold resources from multiple testbeds, and the slice owner can therefore use one slice credential to request resources from many testbeds, and subsequently use the same slice credential at each testbed to access the resources they have reserved.

a. What information will be provided by the "federation" to enable the testbeds to make authorisation decisions?

This question relates to the specification of what the "federation" is. This is still currently in discussion, primarily in the sustainability task, T2.3, and the discussion has revolved around what a future (post-project) operator of Fed4FIRE (termed the "federator") can provide to enable different users and testbeds to collaborate. The federator may operate different operational models, depending on their level of involvement – for example the federator may simply advertise testbed resources or they may provide a complete experimenter service where they manage all aspects of testbed interaction. The information the federator provides may vary according to which models they support, but could include:

- User registration and identity provision, in which case the user credentials will be issued by the federator
- Management of testbeds' resource reservations (if the testbeds allow), in which case the reservation slots will be decided by the federator and communicated to the testbeds
- Provision of any standardised vocabularies for resource management and access control (see later)

b. Should there be a federation-wide vocabulary of terms that can be used by one organisation to assert some information (e.g. some form of access right) and can also be used in another organisation when making an authorisation decision? For example, if TB1 says a user is a principal investigator, what meaning does this have at TB2? If TB2 decides it has meaning, it may use this as part of an access control decision - e.g. "let principal investigators have priority access to my resources".

This is not yet decided, but it is asserted here that there should be some mechanism for different parties to understand the intentions of others when declaring permissions or otherwise. At the

current time, the communication mechanism is the slice credential standard and X509 certificates. The slice credential has specific fields to identify the slice ID, the owner ID and any authorised delegated users, and the PDP work has implemented a parser that can understand this format, so any testbed deploying the PDP can process these credentials.

Having said this, there is likely to still be a long way to go before we have an expressive means of declaring the intentions of a party regarding permissions. As a step towards this, we have adopted the notion of assertions (fully described in D7.2), as a mechanism for a party to make statements about entities (specifically rights of users on resources). At the current time, there is no semantic element to these assertions – they are simply conjunctions of tokens that need to be consistent for access to be granted. Whether a fully semantic description of assertions made by parties is required remains to be seen, and this is the work of the ontology tasks in WP5.

c. **How is information needed for an authorisation decision propagated from a provider to a potential recipient? What is the carrier? Should this information be expressed as attributes? If so, what is the schema?**

The communication mechanism is the slice credential described (many times) above. The slice credential has a well-defined schema that is implemented by numerous deployments of SFA.

6. **By its use of the phrase "trusted location of root certificates", the definition of the certificate directory in D2.1 seems to imply that if you trust the location of certificates, then you must trust all certificates in it. This sounds open to compromise – if a bad root certificate gets in the store, it is propagated quickly and the testbeds have no choice but to trust it. A testbed provider should make a decision as to whether they trust other testbeds' identity providers, but having a trusted location is something quite different. Instead of having a trusted location, we can use the certificate directory as a single repository for CA certificates that have passed some kind of entry requirement, but the final decision about which CA certificates they trust is down to the individual organisations and users. This assertion does not change the technical specification of the certificate directory, only its use.**

This is not really a question, but an assertion aimed at giving the testbeds the freedom to make their own trust decisions. It is believed that this assertion is still completely valid, and it is consistent with the assertion in (4) concerned with ensuring testbeds are in full control of their resources. This has no impact on technical implementations, as testbeds can download a bundle of certificates from the certificate directory, then choose which ones to trust, and only install these.

7. **Attributes in certificates should be *static details of users only*. This means certificates should not contain any dynamic information that may also be the part of an authorisation decision. For example, reservations are highly dynamic (they perish with time) and an identity certificate cannot be reissued every time an assertion about the user's rights to a reservation changes.**

a. **How will we communicate any assertions about dynamic authorisation in the federation?**

The assertion above needs to be expanded, in that the SFA uses certificates to identify all entities, not just users, but overall the assertion is deemed to be still valid, because certificates are unsuitable for holding dynamic information – any change of information in a certificate requires the certificate

to be re-minted. There is also no requirement for any kind of dynamic information to be contained within a certificate.

As discussed above, the PDP work has aimed to address the dynamic nature of reservations and permissions through its assertion mechanism where trusted authorities can describe the conditions that need to be true for access to be granted to a particular resource (this is the access policy), and the access control decision is a match of the facts that are present and correct against the policy. Using this pattern, the certificates representing entities need only contain static information about the entity only.

8. **Is it also the task of an identity provider to say what rights the credentials represent at the ID provider's site? For most sites this makes sense, but only for static information about the user that does not change for the lifetime of the user's identity credential. For example if a user is a student, an identity credential asserting this is acceptable, but if the user graduates, they will need a new identity credential that reflects their new status. A vocabulary describing any static assertion attributes meaning will have to be determined.**

If there are rights or roles in an identity credential, it is probable that they are of most interest to the issuing organisation (e.g. a university issuing a student card), but other organisations may choose to trust this information (e.g. rely on the assertion of a university of a person's student status).

As mentioned above, to bind transient or dynamic information in any credentials is not really a good idea, because the credentials become out of date or maintenance becomes a problem.

The point about any static assertion attributes and providing a vocabulary for them is still very pertinent – it is important that different testbeds are able to interpret assertions from external sources, and vocabularies and semantics are a mechanism to provide some means of achieving this. This has not yet been investigated by the project, but it remains a potential research topic.

9. **For the Fall-Back User Registry IdP, the user does not have any rights at the IdP, neither can the Fall-Back User Registry IdP assert rights at any other site (unless the remote sites are happy with delegating access control to a third party, and this is a decision for each testbed). The Fall-Back User Registry IdP might assert common attributes ("academic", etc) and if this is the case we will need to agree on a common shared vocabulary for these assertions.**

This question pertains to the specific case of the fall-back identity provider, and in practice so far this has not been required. Users have been content to sign up to one of the existing registries, e.g. the Virtual Wall. The point about the need for a common vocabulary is clearly relevant, and has been discussed in the previous question.

10. **The SFA defines a hierarchical naming scheme that denotes chains of authority (e.g. "hrn:fed4firefedora14.tb1.pi"). Is there need to define a vocabulary for this in FED4FIRE, specific to FED4FIRE, or will existing conventions suffice?**

So far, it seems that the naming hierarchy conventions of SFA (as given in the example above) are acceptable to most project partners. We should monitor to determine if this continues to be the case, but it is expected that this will be accepted because it is an established standard with support and it is not difficult for parties to adopt.

11. **There is work needed at each of the test-beds for making their individual authentication and registration systems interoperate with the SFA certificates. At present, it is unknown how much effort this will require because each testbed has different implementations and procedures. It is a critical question to determine the level of effort required for each testbed, and this must be coupled with benefits to the testbeds. The testbeds must have a compelling reason to commit effort to interoperation with SFA, and providing this compelling reason is the task of the architecture workpackage.**

The approach to this question has been the design and implementation of the PDP system described in D7.2. This provides a way for users to protect FRCP-accessible resources via a dynamic access policy that is able to be made consistent with the resource allocations made by Aggregate Managers in the SFA domain.

The effort required to deploy the PDP inside testbeds has been shown so far to be in the order of about 2 weeks, including installing the dependencies and learning. This is based on our experiences of deployment in BonFIRE and the Virtual Wall. However, each testbed deployment has some customisation to account for the specific needs and requirements of each testbeds, so this figure may be greater or lesser.

The PDP is not at all mandatory - it is each testbeds' choice whether to install OMF and the PDP, and they will only do so if they determine it is worthwhile. The reason for a testbed to install the PDP is generally if they want to support FRCP-based access to their resources, and the PDP gives them the benefit of enforcement of the decisions of the SFA Aggregate Manager in their testbed.

In most cases so far and those foreseen, a testbed's support of OMF (the reference implementation of FRCP) is independent and in parallel to their existing experiment control systems. Thus far, any installation of OMF and the PDP protecting access via FRCP at testbeds has been complementary to any existing installation at testbeds, and provides an alternative entry point, which does not affect their existing methods of entry and access control.

6 Conclusions

This deliverable has reported on the progress made for the first cycle of Fed4FIRE in WP7 regarding trust, SLA and security. Some specific results were as follows:

- A certificate directory has been deployed at UPMC, where root CA certificates can be downloaded by federation participants.
- During cycle 1, the main activities concerning SLAs have been centred around gathering requirements, identifying the most suitable implementation in the framework of Fed4FIRE (distributed/centralized) and studying the modifications required in order to adapt the SLA tool in order to meet Fed4FIRE's approach. A key outcome of the requirements gathering was that the more well-used (i.e. successful) a testbed was, the greater the resource management challenges they faced because demand outstripped supply. In relation to the architecture of the SLA, different approaches were analysed and the preferred option was "Federated SLA tool" since it involved a thin federation layer, more flexibility to the testbeds to adopt SLA management.
- In order to gather requirements and testbed owners' expectations and needs of the reputation service, the testbeds were surveyed with questions about the metrics that testbeds are planning to monitor and whether the existence of such a service was in the interest of them. The results revealed that most testbed owners wish to have their testbed evaluated and that the reputation service provides an added value with regards to its trustworthiness. As documented in D7.1 and D7.2, the ROCQ algorithm was selected as the basis of the reputation-based trust mechanism, to be implemented and adopted within the Fed4FIRE framework.

The other contribution of this deliverable has made is provide answers to the questions raised at the end of D7.1. We now know answers to most of these, based on the work we have done in the period between delivery of D7.1 and the current time. Many of the questions are addressed by the PDP work implemented up to PM18 and described in D7.2. The main reason for this is that the questions actually represent issues that needed to be resolved in order to make progress in the area of federation authorisation. Thus the work done on the PDP, and the discussion in the project around this subject were the main sources of answers to the questions.