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**

D8.13 Third report on the Federation Standardization Task Force

Work package	WP8 Shared Support Services
Task	Task 8.3 Sustainable standardization
Due date	30/11/2015
Submission date	31/03/2016
Deliverable lead	Timur Friedman (UPMC)
Version	1.0
Authors	Timur Friedman (UPMC)
	Ciro Scognamiglio (UPMC)
	Alexander Willner (TUB)
	Brecht Vermeulen (iMinds)
Reviewers	Piet Demeester (iMinds)

Abstract	This document describes the progress of the Federation Standardization Task Force. The document describes the work that has been done since the roadmap was first established.
Keywords	Standardization

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 document describes the work of the Federation Standardization Task Force since the roadmap was first established (see Deliverable D8.3). In establishing the roadmap, we had identified three general areas in which specifications exist that could potentially benefit from harmonization:

1. resource specifications, or **RSpecs**, which we also refer to as ontologies;
2. the Slice-based Facility Architecture, or **SFA**, which is a set of APIs;
3. the Federated Resource Control Protocol, or **FRCP**.

The RSpecs have moved the most in the direction of classic standard setting, with a specification that is currently being discussed in a World Wide Web Consortium (W3C) Community Group. A key part of the SFA specifications, which is the Aggregate Manager API (AM API) has been harmonized between Europe and the United States in the form of the Common AM API. FRCP has not been subject to harmonization, but the need is not there at present.

The difference with D8.10 is progress on the ontology work for resource descriptions. The other activities were finalised.

Table of Contents

1 Introduction

Fed4FIRE is putting in place a federation of testbeds for experimentation in networked computing environments. To enter into this federation, an individual testbed exposes a set of common APIs for running experiments. Users employ software programs, or "tools", to manage part or all of the experiment life cycle, which includes browsing, reserving and provisioning available resources, configuring those resources, running experiments, collecting data, and releasing the resources. These client-side tools are written to interact with the APIs exposed by the testbeds. Since all testbed owners and all tool developers need to work from a common set of specifications, there is a need for standardization, meaning that the community needs clear documents to serve as common points of reference.

The community in question extends well beyond the partners in the Fed4FIRE project, to others in Europe and across the world. It is a community that is not simply waiting to receive standards from Fed4FIRE. It consists of actors who, in some cases, have themselves authored significant specification documents. Other parties wish to have an active role in shaping these specifications as they move forward. Therefore, it is not sufficient to produce these standards only as project deliverables.

This document provides an update on the efforts within Fed4FIRE to work towards internationally agreed specifications and, eventually, standards, where these are needed.

2   A common ontology for resource descriptions

The area that has moved the most in the direction of official standards setting institutions is the description of testbed resources. There is considerable interest among researchers and developers on both sides of the Atlantic in arriving at a common information and data model for describing the resources involved in networked computer communication. These might, for instance, be end-hosts, routers, and the links between them, any of which might be physical or virtual. Current resource descriptions would list such resources in a way that a knowledgeable human could perhaps understand by reading the words, but that means little to a machine. By describing the resources using a common vocabulary and well-defined models, it becomes possible for a machine to reason about them, asking questions, for instance, such as whether every host that has been requested also has a network connection, and whether the entire set of hosts and connections forms one connected component or not. With ontologies, one can semantically manage the life cycle of a resource: discovery, selection, reservation, provisioning, monitoring, control, termination, authentication, authorization, and trustworthiness.

Interest in the ontology area extends beyond Fed4FIRE to partners in GENI in the United States, notably those working on the ExoGENI testbed (http://www.exogeni.net/); to researchers at the University of Amsterdam who are responsible for fundamental prior work on low-level network ontologies; and to those at ETSI who worked on the Measurement Ontology for IP traffic (MOI). The work bases on what is known as the Semantic Web (http://www.w3.org/standards/semanticweb/), which is a web of information that can be reasoned about by computers. Because of the variety of different actors and uses for such an ontology, there is sense in moving the discussion to a standards setting body, and this is indeed what we have done in Fed4FIRE.

We bootstrapped the process via a discussion forum that we brought into being, called Open Multinet (http://open-multinet.info/). Initial specifications were drafted there, and this formed the basis for bringing the specification to a Community Group (formerly "incubator group") at the World Wide Web Consortium, or W3C (http://www.w3c.org/community/omn).

The ontology current specification draft is available at http://w3c.github.io/omn, the ontology itself is at http://w3id.org/omn, while the ontology source, tools, and code examples are available at https://github.com/w3c/omn.

Discussion within a Community Group remains at the specifications level. Community Groups are considered to be "tuned" for transition to a standards track, though there are no immediate plans for this group to make the transition.

This effort was, and continues to be, led by Alexander Willner of Fed4FIRE partner TU Berlin, who is the founding chair of the W3C Federated Infrastructures Community Group. It is planned to extend this work beyond GENI and FIRE testbed federations, by to including in particular Intercloud approaches.

The work conducted within Fed4FIRE Task 5.2 and the Ontology Task Force within the Open-Multinet Forum resulted in the preparation of a workshop proposal the Extended Semantic Web Conference (ESWC), one of the most important conferences on semantic web technologies. The Workshop will be held in May 2016 during the conference.

3 The Aggregate Manager API

Another area of common specifications that has moved forward in a very significant way in Fed4FIRE is the development of the Aggregate Manager API (AM API), which exchanges resource descriptions between users and testbeds. Recall that an aggregate manager is the testbed component that responds to requests generated by users for testbed resources, by reserving those resources, in some cases provisioning them, and making them available to the user in question. In addition to keeping track of a testbed's resources, authentication and authorization of the user, or the tool acting on behalf of the user, are a key part of what an aggregate manager does. Fed4FIRE has since the beginning adopted the Slice-based Facility Architecture (SFA) as a basis for defining interactions with aggregate managers, but SFA is a moving target, and different testbeds have implemented different variants. Fed4FIRE took the initiative to engage partners in GENI in the USA in order to draw up a common specification for the aggregate manager portion of the API. This was done on the Open Multinet forum that we created.

Fed4FIRE has been using the GENI AMv3[1] and AMv2[2] specifications as defined by GENI. Defining, prototyping, implementing and developing these APIs was a tremendous effort done by the people at GENI, along with a few partners from Europe and Japan (for example, the first working SFA-based testbed federation involved PlanetLab testbeds in the three countries, with much of the coding to support the API being performed in a project prior to Fed4FIRE by current Fed4FIRE partner Inria, in collaboration with the University of Princeton and the University of Tokyo). However, the documentation of these specifications lacks some information and clarity and also had left some items to be defined. In addition, some functionality also stood to be improved. This was acknowledged by people at GENI and it was agreed that GENI and Fed4FIRE would harmonize on a Common AM API for the two initiatives. Version 1 of that specification is based on GENI AM v3, but improves on the issues found in the existing standard and makes things more clear where appropriate.

The approach was to bring together people from both Fed4FIRE and GENI and to identify the existing issues and figure out improvements that could be made. The community that is interested in this sort of specification is narrower than the community surrounding the ontology specification. For this reason, there are no plans to bring the specification to a standards setting body. The work has gone forward on the Open Multinet forum that we created, together with tools from other sites, such as GitHub.

The collaboration has been organized as shown in this diagram:

---

[1] http://groups.geni.net/geni/wiki/GAPI_AM_API_V3
[2] http://groups.geni.net/geni/wiki/GAPI_AM_API_V2
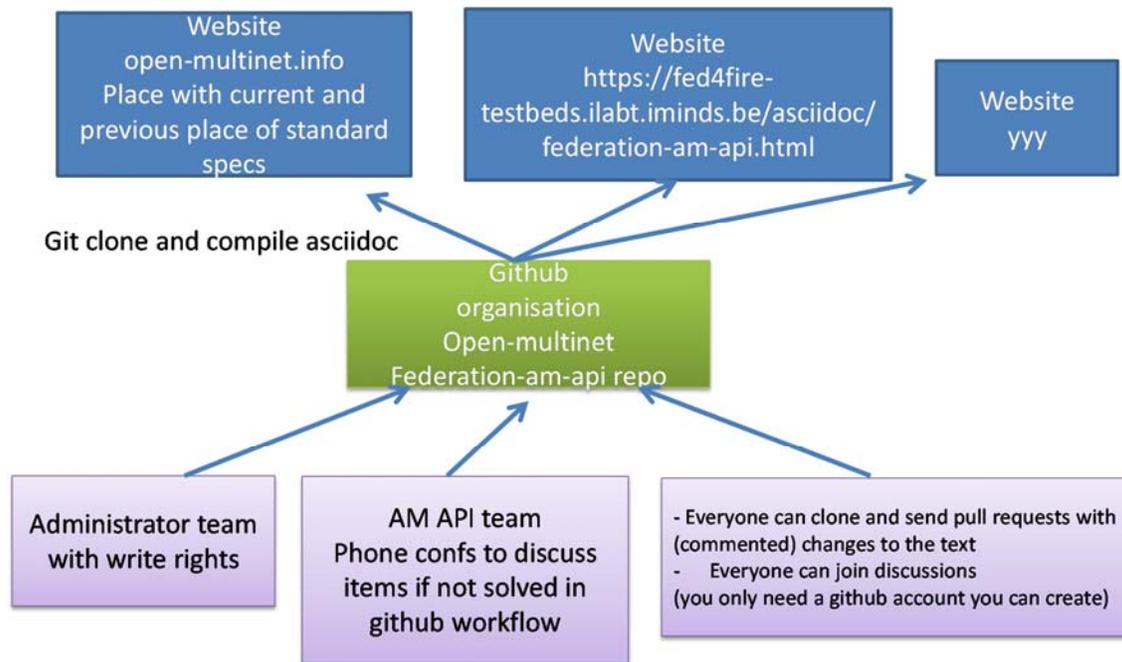
Our aim was to create a specification which could easily be commented upon and improved by all participants, so we did the following, after reaching consensus on this model in the working group:

- The standard is written in AsciiDoc, which is a simple markup language that does away with the sort of heavyweight tagging of markup languages such as HTML. As a result, a document written in AsciiDoc can easily be read and edited in its original text version. A number of different compilers have been written for AsciiDoc, so that attractively formatted documents can be prepared both for the web (HTML) and for printed output (PDF). The text-based format of the source documents facilitates version control.

- We use the GitHub website to interact on the specification. The master AsciiDoc document is kept on GitHub (https://github.com/open-multinet/federation-am-api), meaning that:
  o each participant can create his or her own local "clone" of the Git repository, can keep this clone up to date by pulling changes down from the repository, and can upload his or her own suggested changes by issuing what are called "pull requests", which are invitations for others to see what he or she has done;
  o each participant can raise issues having to do with the specification, using GitHub's facilities for this, which allow for discussion around specific questions.

- There is a small set of administrators (currently, one person each from the EU the US) that can update the master document after consensus within the AM API community.

- The master document can easily be cloned to any website, for compiling and publishing there.

We bootstrapped the process with a Google Document, in which we listed the 34 issues, big and small, in the existing GENI specification that we wanted to work on.

https://docs.google.com/document/d/12JyNEAQDPHjm8h3hmyrD2PJkjVulFo1N3OgYI6BzWvw/edit#

We held five conference calls within the working group in order to resolve all of the issues.

In the working group, we decided to divide the specification into five documents:

- General concepts, such as as unique IDs (URNs) and architectural concepts: https://fed4fire-testbeds.ilabt.iminds.be/asciidoc/general.html
- The AM API itself: https://fed4fire-testbeds.ilabt.iminds.be/asciidoc/federation-am-api.html
- The resource specification, or RSpec (note that this is not yet the ontology-based resource specification that is being worked on in the group described above): https://fed4fire-testbeds.ilabt.iminds.be/asciidoc/rspec.html
- SFA credentials: https://fed4fire-testbeds.ilabt.iminds.be/asciidoc/credential-sfa.html (this document is still to be drafted)
- ABAC credentials, to enable attribute-based admission control (ABAC): https://fed4fire-testbeds.ilabt.iminds.be/asciidoc/credential-abac.html (this document is still to be drafted)

The first three documents (general concepts, AM API, RSpec specification) are much improved over what was available before. Many people have been using the API specification, and we have received positive feedback on it. The current state is fairly stable and we plan to improve upon it further as questions or problems arise.

There are still open issues, which we plan to address in a version 2 of the specification: https://github.com/open-multinet/federation-am-api/issues. There are also a number of "pull requests" proposing changes to the documents: https://github.com/open-multinet/federation-am-api/pulls. Ongoing work is on properly harmonizing the approach to credentials. FIRE-GENI face-to-face meetings are an occasion for these specifications, or specific parts of them, to be discussed.

To summarize the work on the AM API, we currently have a working process for defining and improving the Common AM API specification based on:

- An AsciiDoc definition of the API and relevant concepts;
- A discussion process with issues and pull requests handled through GitHub: https://github.com/open-multinet/federation-am-api
- A mailing list for discussion of broader issues or to announce conference calls: https://lists.open-multinet.info/mailman/listinfo/federation-am-api-open-multinet

The most important results of this effort have been:

- to bring the harmonization effort up a level, outside of national projects such as GENI and Fed4FIRE;
- to provide a good working document, open to everyone;
- to provide better standard descriptions.

A current compiled version of the specification can be found at:

https://fed4fire-testbeds.ilabt.iminds.be/asciidoc/federation-am-api.html

4 FRCP

The Federated Resource Control Protocol (FRCP) specification has moved the least towards a community-wide harmonization effort or standardization. However, it is important to underline that this is not a criticism. We do not harmonize or standardize for the sake of being able to claim that we have done so. Rather, such efforts should meet a need, and we simply recognize that in the case of FRCP there is at present very little need. Just a few tools (OMF/OML, NEPI, FITeagle) currently make use of the specification, and the existing specification from Fed4FIRE partner NICTA has proved to be of sufficient quality to satisfy the needs of the authors of those tools. The specification can be found on GitHub at this address: https://github.com/mytestbed/specification/blob/master/FRCP.md.