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**

D2.8 – Third integration and testing roadmap

Work package	WP2
Task	T2.2
Due date	31/01/2015
Submission date	31/03/2016
Deliverable lead	Brecht Vermeulen (iMinds)
Version	1.0
Authors	Brecht Vermeulen (iMinds)
Reviewers	Alexander Willner (TUB) Peter Van Daele (iMinds)

Abstract	This deliverable of task 2.2 in the WP2 architecture work package defines the integration and testing roadmap for the third cycle of the Fed4FIRE project.
Keywords	Federation, Integration, Testing

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

Executive Summary

This deliverable defines the third testing and integration roadmap of Fed4FIRE, targeting the third cycle of the project.

At the end of cycle 1 all testbeds (Nicta Norbit, iMinds Virtual Wall, iMinds w-iLab.t, UPMC & Inria PlanetLab Europe, FOKUS FUSECO, NTUA Netmode, UTH Nitos, NIA Koren, Ofelia (i2CAT, UBristol, iMinds), except BonFIRE and UC Smart Santander, had a compliant Aggregate Manager (AM) interface. For the BonFIRE testbeds (Inria Grid5000, EPCC) testbeds, the adaptation was not that straight forward and was reached partly in cycle 2 and fully in cycle 3. Smart Santander uses a REST interface as it currently offers a service oriented towards accessing the sensor measurements. For this, we introduced light federation in cycle 2 and in cycle 3 advanced federation through the YourEPM service composition framework.

The new additions for cycle 2 were the following:

- Layer 2 connectivity between multiple sites: target is to interconnect at least 3 sites by phase 1 of cycle 2 and at least 5 sites by the end of cycle 2.
- Infrastructure monitoring: this comes in two flavours. Infrastructure monitoring for federation services (e.g. reputation, SLA monitoring, …) will be adopted earlier by the testbeds (at least one testbed will provide this by phase 1 of cycle 2, the others will implement this by the end of cycle 2), then the infrastructure monitoring for experimenters (which contains more and more detailed monitoring data) which is targeted for cycle 3.
- PDP (policy decision point) for FRCP: this PDP makes it possible to run OMF in a secure way. It will be implemented by 3 testbeds in phase 1 of cycle 2 and more will adopt it by the end of cycle2.
- SLA management module: will be deployed centrally and at least 1 testbed will implement the testbed side of SLAs by the end of cycle 2.
- Reservations in the future: about half of the testbeds will implement this in cycle 2
- Direct (IPv4 or IPv6) or indirect (through gateway) IP access without VPN: possible at all testbeds by the end of cycle 2.
- Ontology based RSpec: first prototypes should be implemented by the end of cycle 2.
- Services: the service directory will be ready by phase 1 of cycle 2, a couple of services by the end of cycle 2.

The new additions and improvements for cycle 3 are the following:

- YourEPM service composition framework
- Speaks-for mechanism to make it possible to deploy services speaking for users without security implications (such as providing private keys)
- More advanced SLA and reputation modules
- New Member and Slice authority APIs to improve the user experience for e.g. collaboration in experiments
- API on top of the continuous testing so the results (e.g. health score) can be used in client tools

In Fed4FIRE there is also a test suite (jFed framework) which is used for testing the different APIs. In this way developments can be tested manually, and afterwards they are tested automatically in nightly and daily tests with the same framework. The test suite now also supports setting up complex scenario testing as e.g. speaks-for testing, automated tutorial testing (through ansible scripting), and health score calculation.

Acronyms and Abbreviations

AM	Aggregate Manager
API	Application Programming Interface
DoW	Description of Work (official document describing the Fed4FIRE project)
FRCP	Federated Resource Control Protocol
GENI	Global Environment for Network Innovations
NEPI	Network Experimentation Programming Interface [5]
OMF	cOntrol and Management Framework [6]
PLE	PlanetLab Europe
RSpec	Resource Specification
SFA	Slice-based Federation Architecture
Testbed	Combination of testbed resources (e.g. servers) and testbed management software

Table of Contents

1 Introduction

This deliverable is the third one in Task 2.2, the integration and testing task of the Fed4FIRE project. The Fed4FIRE project is structured around three development cycles. This deliverable defines the integration and testing roadmap for cycle 3.

2 Refinement of timing

In cycle 1 and 2 we refined the cycle in 2 phases. This was not done in cycle 3 because there was a fully operational federation to be used by experimenters and it was possible to run the cycle 3 testing in parallel.

3 Cycle 2 new functionality

In cycle 2, the following new functionality was added:

- Layer 2 connectivity between multiple sites (also called stitching): target is to interconnect at least 3 sites by phase 1 of cycle 2 and at least 5 sites by the end of cycle 2.
- Infrastructure monitoring: this comes in two flavours. Infrastructure monitoring for federation services (e.g. reputation, SLA monitoring, …) will be adopted earlier by the testbeds (at least one testbed will provide this by phase 1 of cycle 2, the others will implement this by the end of cycle 2), than the infrastructure monitoring for experimenters (which contains more and more detailed monitoring data) which is targeted for cycle 3.
- PDP (policy decision point) for FRCP: this PDP makes it possible to run OMF in a secure way. It will be implemented by 3 testbeds in phase 1 of cycle 2 and more will adopt it by the end of cycle2.
- SLA management module: will be deployed centrally and at least 1 testbed will implement the testbed side of SLAs by the end of cycle 2.
- Reservations in the future: about half of the testbeds will implement this in cycle 2
- Direct (IPv4 or IPv6) or indirect (through gateway) IP access without VPN: possible at all testbeds by the end of cycle 2.
- Ontology based RSpec: first prototypes should be implemented by the end of cycle 2.
- Services: the service directory will be ready by phase 1 of cycle 2, a couple of services by the end of cycle 2.

4 Cycle 3 new functionality

The new additions and improvements for cycle 3 are the following:

- YourEPM service composition framework
- Speaks-for mechanism to make it possible to deploy services speaking for users without security implications (such as providing private keys)
- More advanced SLA and reputation modules
- New Member and Slice authority APIs to improve the user experience for e.g. collaboration in experiments

In Fed4FIRE there is also a test suite (jFed framework) which is used for testing the different APIs. In this way developments can be tested manually, and afterwards they are tested automatically in nightly and daily tests with the same framework. The test suite now also supports setting up complex scenario testing as e.g. speaks-for testing, automated tutorial testing (through ansible scripting), and health score calculation.

5 Basic approach and test suite

5.1 Testing and integration steps

The implementation, testing and integration phases are made up out of the following steps:

1. Detailed functional description
2. Development
3. First manual testing per testbed with test suite
4. Automatic nightly testing of testbeds with test suite
5. Cross and integration testing between tools and testbeds

The first two are driven by WP3-7, while 3) and 4) are driven from WP2 with a high interaction with WP3-7. The last step is driven by the tool makers (WP5-WP7) and supported by WP2. Based on the output of deliverables D5.2, D6.2 and D7.2 ('Detailed specifications for second cycle' of WP5/6/7, month 5) and milestones M3.2/M4.2 ('Second design specification for facilities') a matrix will be built of which tool will work and will be tested against which testbed in cycle 2 (e.g. the experiment control framework NEPI [5] will work on testbed A,B,C and D, the graphical front-end for resource control Flack [3] will work on testbed D and E, the portal will be compatible with all testbeds, …).

5.2 Test suite

5.2.1 Rationale

The jFed test suite which was developed in the project aids in steps 3) and 4) of the previous section. This test suite is developed in Java and targets following goals:

- Make it possible to do manual compliance testing of the testbeds. Based on the outputs of milestones M3.2 and M4.2 ('Second design specification for facilities'), the compliance will be defined (e.g. SFA functions compliance, APIs for the identity providers, …).
- When the manual compliance testing is successful, do an automatic nightly testing on each testbed, to assure continuous compliance.
- Provide a reference implementation in Java of the client side of the APIs (GENI Aggregate Manager (AM) APIv3 [1], identity providers, …).
- The test suite focuses on logical tests (all steps in the experiment workflow) and add specific interface tests and negative testing (are things breakable ?) where needed. There is e.g. also an ssh login on the nodes to verify that the node itself is accessible and runs the right image.
- The automatic tests post the test reports on a website and send emails in case of problems.
- The test suite is open source (http://jfed.iminds.be) and is a framework based on plugins so that tests can be easily added.

In cycle 3 the following was added to the test suite:

- Make it possible to test the speaks-for implementations of testbed AM implementations
- Provide manual testing with speaks-for credentials
- Enhance the login tests with scripting after the login to make it possible to verify tutorials
- Calculate a health score per testbed based on the measured parameters
- Test the member authority and slice authority APIs according the Common Federation APIs

5.2.2 Comparison to other test suites

We are aware of one similar test suite, the 'GENI AM API Acceptance Tests' [2] which is built around the omni framework. The differences with the Fed4FIRE test suite are:

- The GENI test suite is written in python just as most AM applications. By developing the Fed4FIRE test suite in java, we can also test cross-language compatibility and as far as we know it is the first Java implementation of the SFA client side, which facilitates the development of Java based interfaces/experimenter tools (such as GUIs).
- The GENI test suite covers at this moment only the AM API, while in Fed4FIRE we will test also the other components (identity providers, testbed directories, …).
- When new functionality will be added in the testbeds of Fed4FIRE (e.g. ontology based RSpecs) the test suite will be extended with this and this is not yet available in the GENI test suite.
- For conducting the manual testing, a graphical user interface in the test suite makes it very convenient to inspect the problems (cf. the details that Flack [3] can show about RSpecs and function calls).
- The GENI test suite does not support scenarios, e.g. including SSH login on a node.

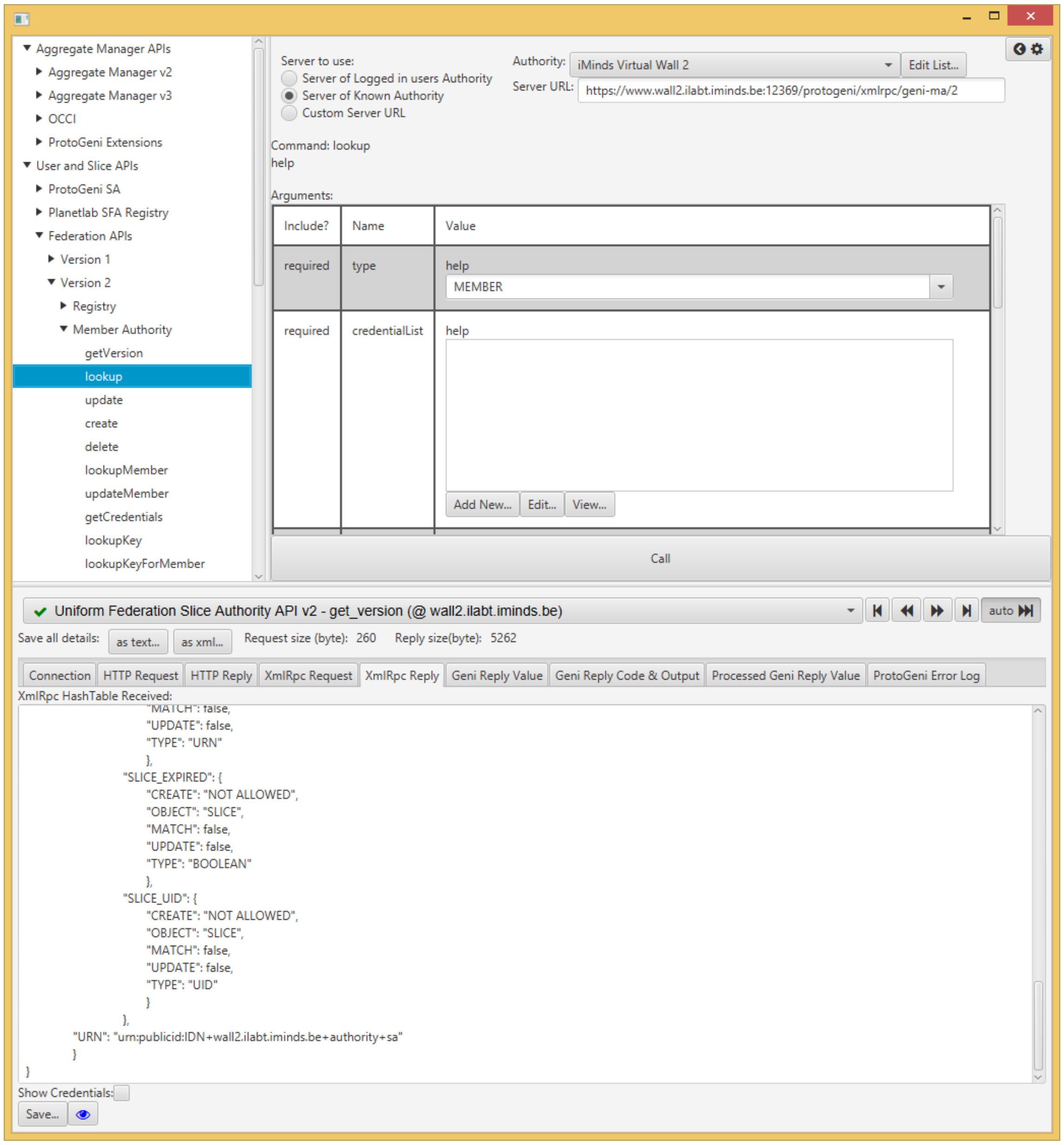
For the functions covered by the GENI AM API Acceptance Tests, the GENI test suite will be used as a cross check.

5.2.3 Current state

The test suite is currently finished for the AM API, MA and SA API and also more advanced scenarios as speaks-for, stitching, tutorial testing and health score calculation.

5.2.4 jFed probe for manual testing

A screenshot of the jFed probe tool for manual testing is depicted in Figure 1. The left side shows all the APIs that are available and that can be tested.
The bottom shows an example of a call with HTTP, XMLRPC, reply value details.
At the top right, the testbed that will be involved in the test is being selected.



Server to use:
Server of Logged in users Authority
[✓ Server of Known Authority]
Custom Server URL

Authority: iMinds Virtual Wall 2
Server URL: https://www.wall2.ilabt.iminds.be:12369/protogeni/xmlrpc/geni-ma/2

Command: lookup
help

Arguments:

Include?	Name	Value
required	type	help MEMBER
required	credentialList	help

Uniform Federation Slice Authority API v2 - get_version (@ wall2.ilabt.iminds.be)

Save all details: Request size (byte): 260 Reply size(byte): 5262

XmlRpc HashTable Received:

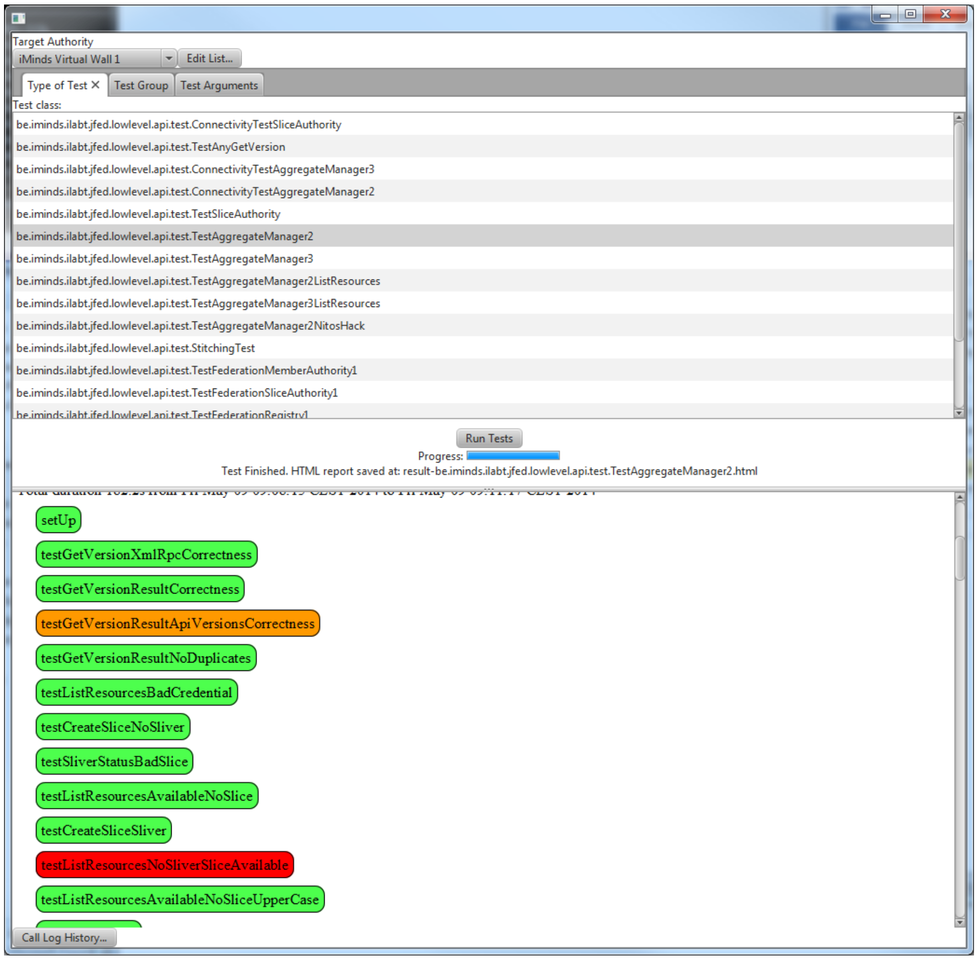
```
                    "MATCH": false,
                    "UPDATE": false,
                    "TYPE": "URN"
                    },
                "SLICE_EXPIRED": {
                    "CREATE": "NOT ALLOWED",
                    "OBJECT": "SLICE",
                    "MATCH": false,
                    "UPDATE": false,
                    "TYPE": "BOOLEAN"
                    },
                "SLICE_UID": {
                    "CREATE": "NOT ALLOWED",
                    "OBJECT": "SLICE",
                    "MATCH": false,
                    "UPDATE": false,
                    "TYPE": "UID"
                    }
                },
        "URN": "urn:publicid:IDN+wall2.ilabt.iminds.be+authority+sa"
        }
}
```

Show Credentials:

Figure 1: Screenshot of jFed probe for manual testing

5.2.5 jFed scenario testing

In the screenshot below both, an overview of the available automated scenarios that can be tested and a single scenario that was run are depicted.



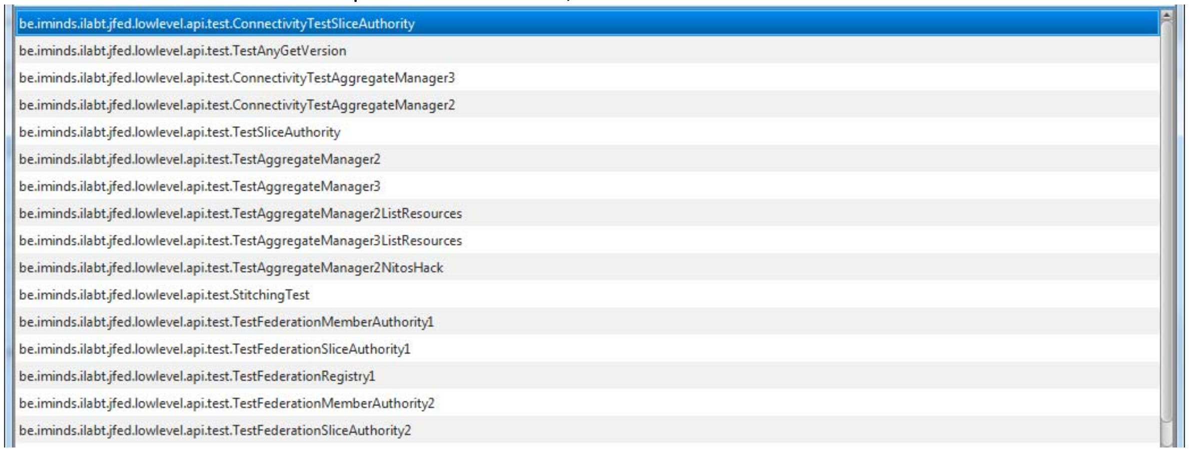
be.iminds.ilabt.jfed.lowlevel.api.test.ConnectivityTestSliceAuthority
be.iminds.ilabt.jfed.lowlevel.api.test.TestAnyGetVersion
be.iminds.ilabt.jfed.lowlevel.api.test.ConnectivityTestAggregateManager3
be.iminds.ilabt.jfed.lowlevel.api.test.ConnectivityTestAggregateManager2
be.iminds.ilabt.jfed.lowlevel.api.test.TestSliceAuthority
be.iminds.ilabt.jfed.lowlevel.api.test.TestAggregateManager2
be.iminds.ilabt.jfed.lowlevel.api.test.TestAggregateManager3
be.iminds.ilabt.jfed.lowlevel.api.test.TestAggregateManager2ListResources
be.iminds.ilabt.jfed.lowlevel.api.test.TestAggregateManager3ListResources
be.iminds.ilabt.jfed.lowlevel.api.test.TestAggregateManager2NitosHack
be.iminds.ilabt.jfed.lowlevel.api.test.StitchingTest
be.iminds.ilabt.jfed.lowlevel.api.test.TestFederationMemberAuthority1
be.iminds.ilabt.jfed.lowlevel.api.test.TestFederationSliceAuthority1
be.iminds.ilabt.jfed.lowlevel.api.test.TestFederationRegistry1

Test Finished. HTML report saved at: result-be.iminds.ilabt.jfed.lowlevel.api.test.TestAggregateManager2.html

setUp
testGetVersionXmlRpcCorrectness
testGetVersionResultCorrectness
testGetVersionResultApiVersionsCorrectness
testGetVersionResultNoDuplicates
testListResourcesBadCredential
testCreateSliceNoSliver
testSliverStatusBadSlice
testListResourcesAvailableNoSlice
testCreateSliceSliver
testListResourcesNoSliverSliceAvailable
testListResourcesAvailableNoSliceUpperCase

Figure 2: Screenshot of the jFed scenario testing environment

These scenarios define the testplans that are used, see below:



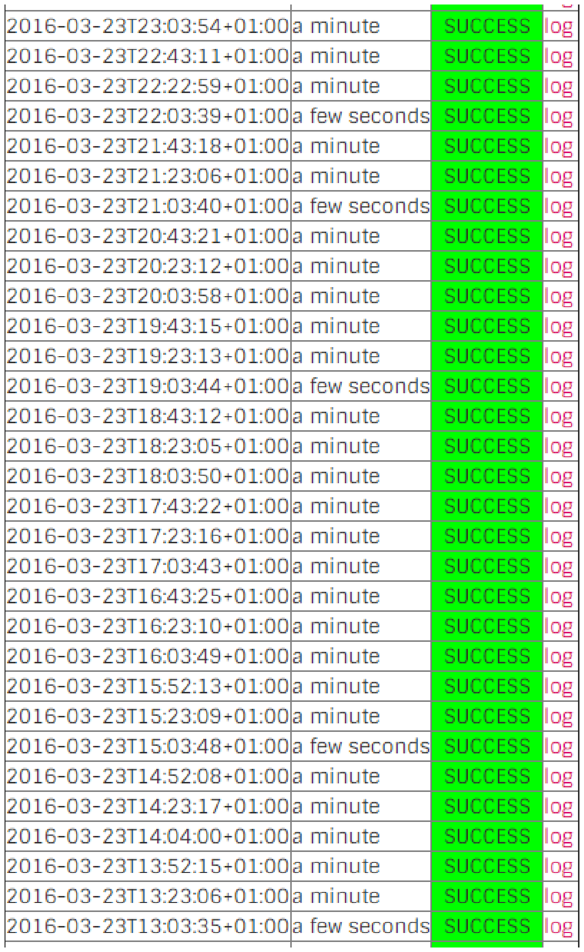
be.iminds.ilabt.jfed.lowlevel.api.test.ConnectivityTestSliceAuthority
be.iminds.ilabt.jfed.lowlevel.api.test.TestAnyGetVersion
be.iminds.ilabt.jfed.lowlevel.api.test.ConnectivityTestAggregateManager3
be.iminds.ilabt.jfed.lowlevel.api.test.ConnectivityTestAggregateManager2
be.iminds.ilabt.jfed.lowlevel.api.test.TestSliceAuthority
be.iminds.ilabt.jfed.lowlevel.api.test.TestAggregateManager2
be.iminds.ilabt.jfed.lowlevel.api.test.TestAggregateManager3
be.iminds.ilabt.jfed.lowlevel.api.test.TestAggregateManager2ListResources
be.iminds.ilabt.jfed.lowlevel.api.test.TestAggregateManager3ListResources
be.iminds.ilabt.jfed.lowlevel.api.test.TestAggregateManager2NitosHack
be.iminds.ilabt.jfed.lowlevel.api.test.StitchingTest
be.iminds.ilabt.jfed.lowlevel.api.test.TestFederationMemberAuthority1
be.iminds.ilabt.jfed.lowlevel.api.test.TestFederationSliceAuthority1
be.iminds.ilabt.jfed.lowlevel.api.test.TestFederationRegistry1
be.iminds.ilabt.jfed.lowlevel.api.test.TestFederationMemberAuthority2
be.iminds.ilabt.jfed.lowlevel.api.test.TestFederationSliceAuthority2

The testplans are detailed in the code, so that there is a single place where all this information resides. Latest version of the testsuite can be found at http://jfed.iminds.be/releases where you can download the code for a specific release.
The testplans are detailed in the source directory:
automated-testing\src\main\java\be\iminds\ilabt\jfed\lowlevel\api\test

Each file there is a scenario test which is built in the standard Java testing framework TestNG (http://testng.org) . In that way the test suite itself is according a standard framework, easily to understand by everyone, and also generates standard HTML output as shown in the screenshots. TestNG also makes it possible to define warnings or hard errors and assertions.

5.2.6 Nightly testing

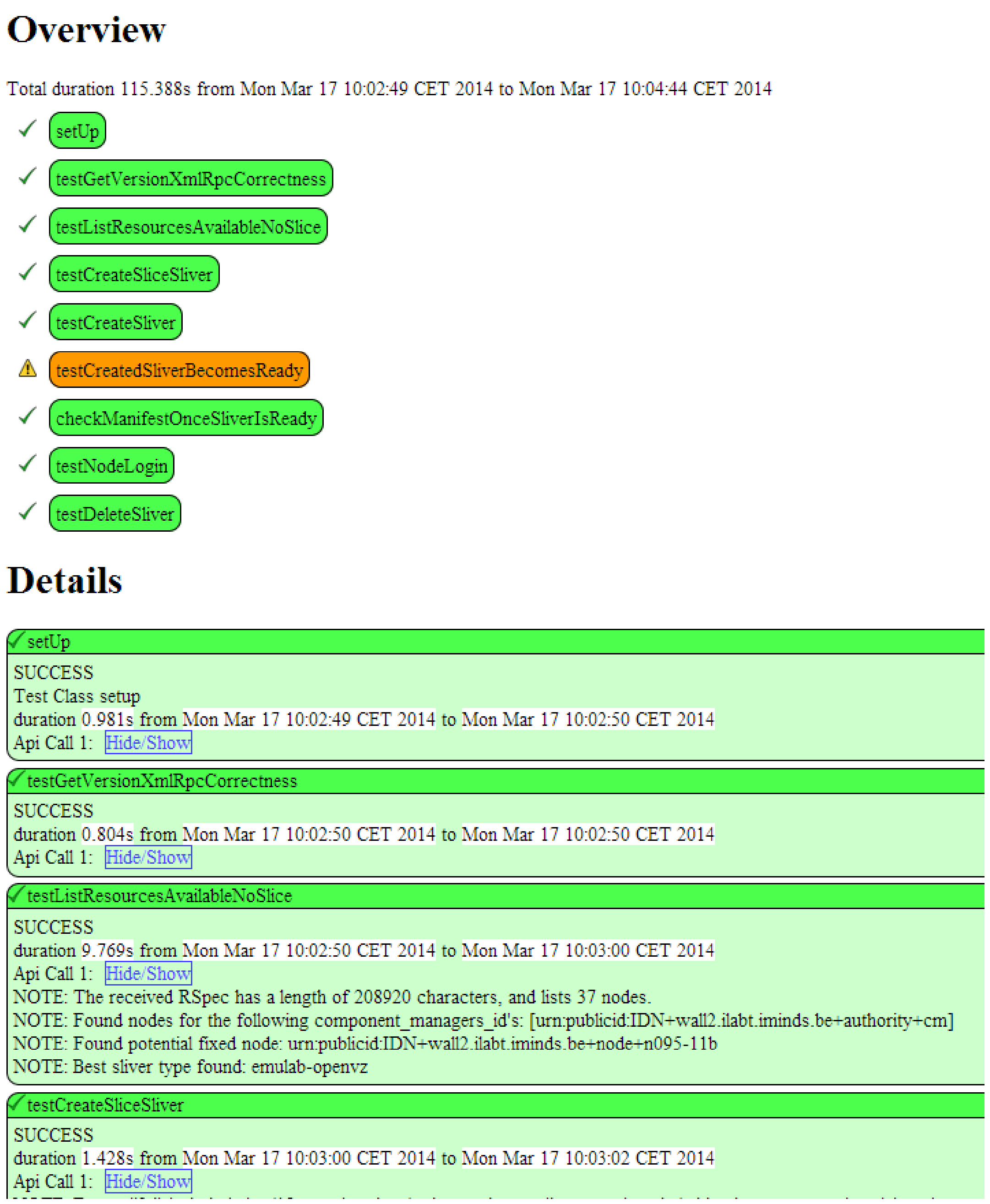
The website http://flsmonitor.fed4fire.eu/wizard shows all nightly testing that is being done. Below an overview of a number of automatic tests on a single testbed is shown.



2016-03-23T23:03:54+01:00	a minute	SUCCESS	log
2016-03-23T22:43:11+01:00	a minute	SUCCESS	log
2016-03-23T22:22:59+01:00	a minute	SUCCESS	log
2016-03-23T22:03:39+01:00	a few seconds	SUCCESS	log
2016-03-23T21:43:18+01:00	a minute	SUCCESS	log
2016-03-23T21:23:06+01:00	a minute	SUCCESS	log
2016-03-23T21:03:40+01:00	a few seconds	SUCCESS	log
2016-03-23T20:43:21+01:00	a minute	SUCCESS	log
2016-03-23T20:23:12+01:00	a minute	SUCCESS	log
2016-03-23T20:03:58+01:00	a minute	SUCCESS	log
2016-03-23T19:43:15+01:00	a minute	SUCCESS	log
2016-03-23T19:23:13+01:00	a minute	SUCCESS	log
2016-03-23T19:03:44+01:00	a few seconds	SUCCESS	log
2016-03-23T18:43:12+01:00	a minute	SUCCESS	log
2016-03-23T18:23:05+01:00	a minute	SUCCESS	log
2016-03-23T18:03:50+01:00	a minute	SUCCESS	log
2016-03-23T17:43:22+01:00	a minute	SUCCESS	log
2016-03-23T17:23:16+01:00	a minute	SUCCESS	log
2016-03-23T17:03:43+01:00	a minute	SUCCESS	log
2016-03-23T16:43:25+01:00	a minute	SUCCESS	log
2016-03-23T16:23:10+01:00	a minute	SUCCESS	log
2016-03-23T16:03:49+01:00	a minute	SUCCESS	log
2016-03-23T15:52:13+01:00	a minute	SUCCESS	log
2016-03-23T15:23:09+01:00	a minute	SUCCESS	log
2016-03-23T15:03:48+01:00	a few seconds	SUCCESS	log
2016-03-23T14:52:08+01:00	a minute	SUCCESS	log
2016-03-23T14:23:17+01:00	a minute	SUCCESS	log
2016-03-23T14:04:00+01:00	a minute	SUCCESS	log
2016-03-23T13:52:15+01:00	a minute	SUCCESS	log
2016-03-23T13:23:06+01:00	a minute	SUCCESS	log
2016-03-23T13:03:35+01:00	a few seconds	SUCCESS	log

Figure 3: Extract of nightly testing results that are exposed on http://flsmonitor.fed4fire.eu/wizard

For each run, a detailed overview of all calls can be analysed:



Overview

Total duration 115.388s from Mon Mar 17 10:02:49 CET 2014 to Mon Mar 17 10:04:44 CET 2014

- ✓ setUp
- ✓ testGetVersionXmlRpcCorrectness
- ✓ testListResourcesAvailableNoSlice
- ✓ testCreateSliceSliver
- ✓ testCreateSliver
- ⚠ testCreatedSliverBecomesReady
- ✓ checkManifestOnceSliverIsReady
- ✓ testNodeLogin
- ✓ testDeleteSliver

Details

✓ setUp

SUCCESS
Test Class setup
duration 0.981s from Mon Mar 17 10:02:49 CET 2014 to Mon Mar 17 10:02:50 CET 2014
Api Call 1: Hide/Show

✓ testGetVersionXmlRpcCorrectness

SUCCESS
duration 0.804s from Mon Mar 17 10:02:50 CET 2014 to Mon Mar 17 10:02:50 CET 2014
Api Call 1: Hide/Show

✓ testListResourcesAvailableNoSlice

SUCCESS
duration 9.769s from Mon Mar 17 10:02:50 CET 2014 to Mon Mar 17 10:03:00 CET 2014
Api Call 1: Hide/Show
NOTE: The received RSpec has a length of 208920 characters, and lists 37 nodes.
NOTE: Found nodes for the following component_managers_id's: [urn:publicid:IDN+wall2.ilabt.iminds.be+authority+cm]
NOTE: Found potential fixed node: urn:publicid:IDN+wall2.ilabt.iminds.be+node+n095-11b
NOTE: Best sliver type found: emulab-openvz

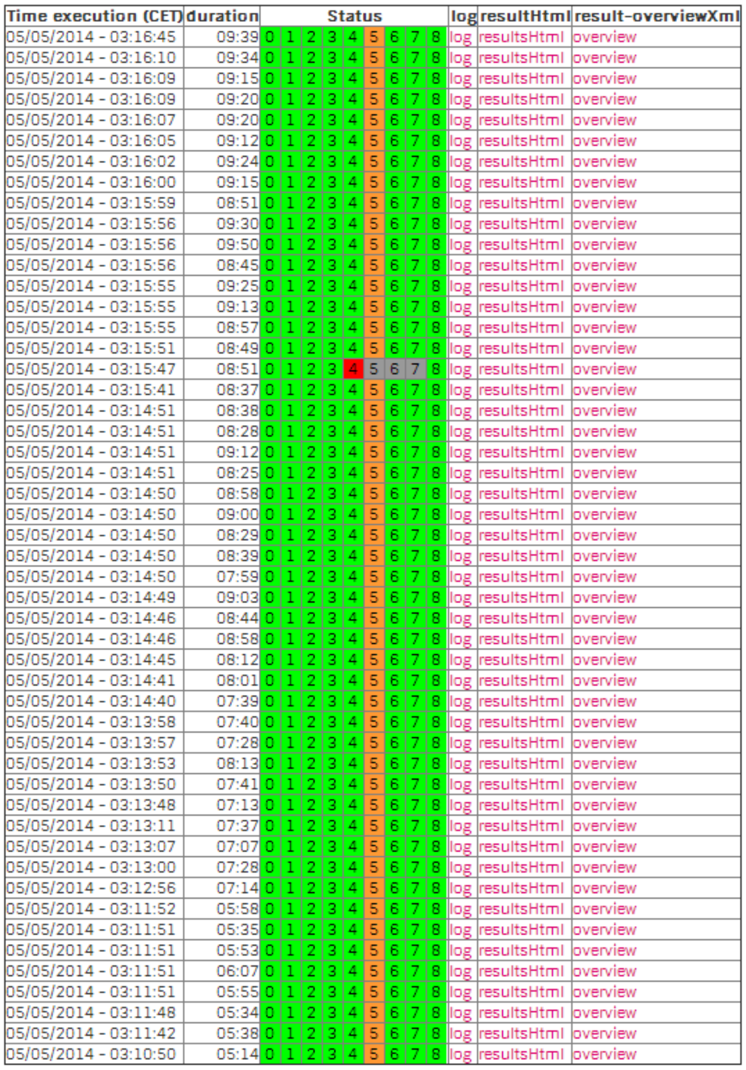
✓ testCreateSliceSliver

SUCCESS
duration 1.428s from Mon Mar 17 10:03:00 CET 2014 to Mon Mar 17 10:03:02 CET 2014
Api Call 1: Hide/Show

Figure 4: Details of all API call results belonging to the nightly testing

5.2.7 Stress testing with jFed

We run also stress tests (a large number of experimenters starting at the same moment an experiment), to see how the testbeds can cope with that (typically for education or tutorials people are in the same room and start at the same moment).
Below the output of a stress test is shown. Each row is the equivalent of an experimenter starting an experiment. The numbers 0-8 mean the different steps in setting up an experiment (e.g. create slice, create sliver, ssh login, delete experiment, …) so that we can see what went wrong. Also important is to see how it takes, so you can anticipate for a tutorial or education class when all people should be ready.

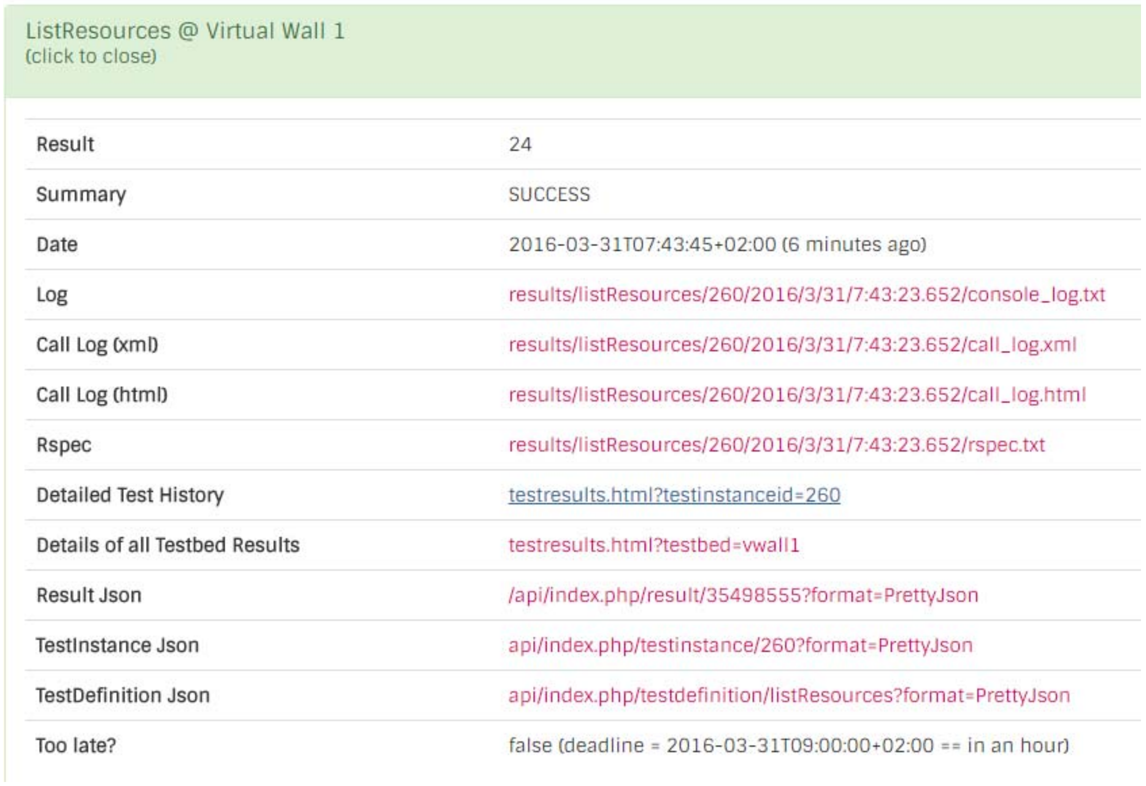


Time execution (CET)	duration	Status									log	resultHtml	result-overviewXml
05/05/2014 - 03:16:45	09:39	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:16:10	09:34	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:16:09	09:15	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:16:09	09:20	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:16:07	09:20	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:16:05	09:12	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:16:02	09:24	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:16:00	09:15	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:15:59	08:51	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:15:56	09:30	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:15:56	09:50	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:15:56	08:45	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:15:55	09:25	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:15:55	09:13	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:15:55	08:57	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:15:51	08:49	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:15:47	08:51	🟩 0	🟩 1	🟩 2	🟩 3	🟥 4	▨ 5	▨ 6	▨ 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:15:41	08:37	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:51	08:38	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:51	08:28	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:51	09:12	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:51	08:25	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:50	08:58	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:50	09:00	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:50	08:29	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:50	08:39	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:50	07:59	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:49	09:03	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:46	08:44	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:46	08:58	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:45	08:12	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:41	08:01	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:14:40	07:39	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:13:58	07:40	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:13:57	07:28	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:13:53	08:13	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:13:50	07:41	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:13:48	07:13	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:13:11	07:37	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:13:07	07:07	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:13:00	07:28	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:12:56	07:14	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:11:52	05:58	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:11:51	05:35	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:11:51	05:53	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:11:51	06:07	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:11:51	05:55	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:11:48	05:34	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:11:42	05:38	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview
05/05/2014 - 03:10:50	05:14	🟩 0	🟩 1	🟩 2	🟩 3	🟩 4	🟧 5	🟩 6	🟩 7	🟩 8	log	resultsHtml	overview

Figure 5: Output of a testbed stress test

6 API on test results of continuous testing

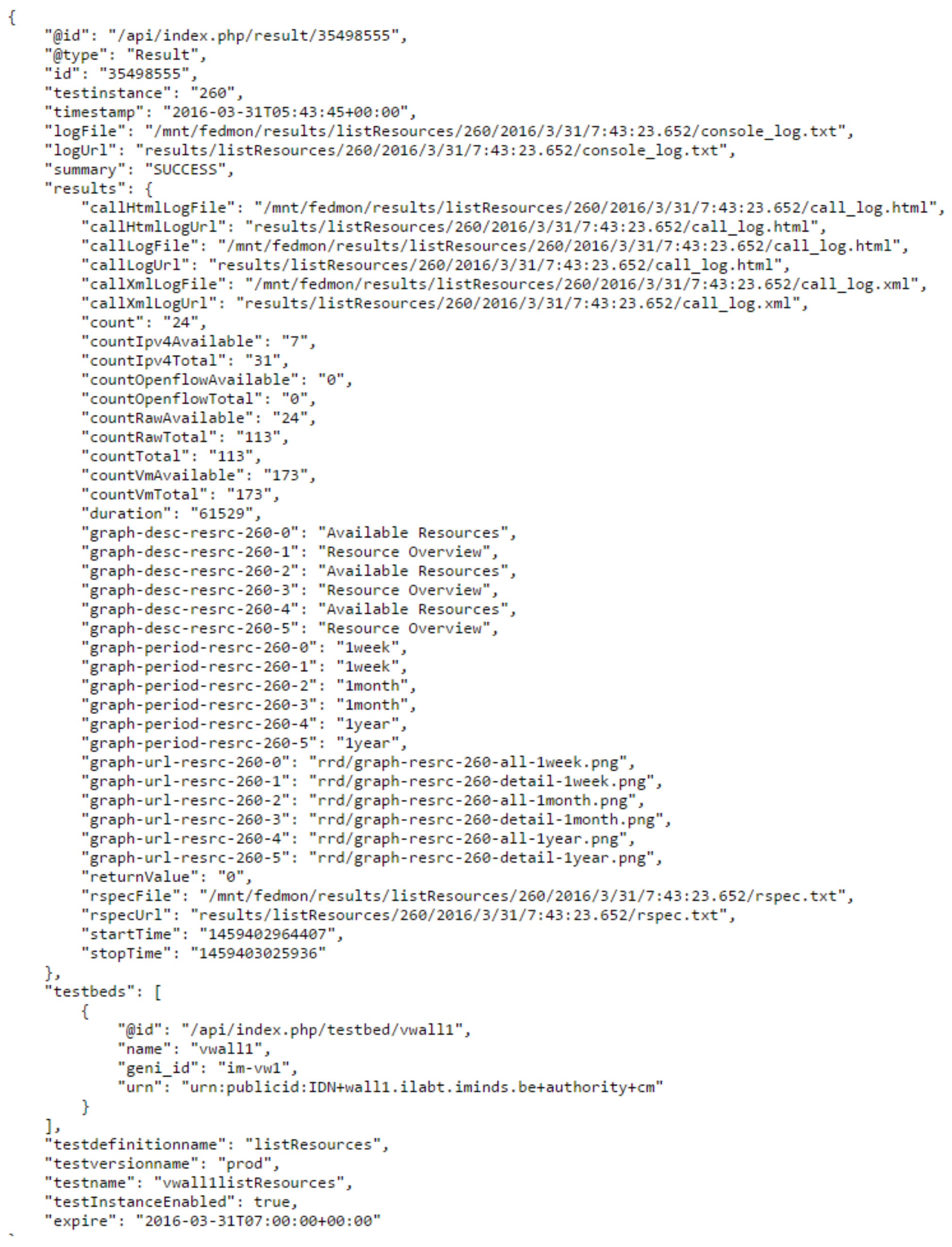
All the test results are centrally saved and can be retrieved through a REST API to use e.g. in client tools. The screenshot below shows for a single test what is available:



ListResources @ Virtual Wall 1
(click to close)

Result	24
Summary	SUCCESS
Date	2016-03-31T07:43:45+02:00 (6 minutes ago)
Log	results/listResources/260/2016/3/31/7:43:23.652/console_log.txt
Call Log (xml)	results/listResources/260/2016/3/31/7:43:23.652/call_log.xml
Call Log (html)	results/listResources/260/2016/3/31/7:43:23.652/call_log.html
Rspec	results/listResources/260/2016/3/31/7:43:23.652/rspec.txt
Detailed Test History	testresults.html?testinstanceid=260
Details of all Testbed Results	testresults.html?testbed=vwall1
Result Json	/api/index.php/result/35498555?format=PrettyJson
TestInstance Json	api/index.php/testinstance/260?format=PrettyJson
TestDefinition Json	api/index.php/testdefinition/listResources?format=PrettyJson
Too late?	false (deadline = 2016-03-31T09:00:00+02:00 == in an hour)

And a specific reply in JSON format:



```
{
    "@id": "/api/index.php/result/35498555",
    "@type": "Result",
    "id": "35498555",
    "testinstance": "260",
    "timestamp": "2016-03-31T05:43:45+00:00",
    "logFile": "/mnt/fedmon/results/listResources/260/2016/3/31/7:43:23.652/console_log.txt",
    "logUrl": "results/listResources/260/2016/3/31/7:43:23.652/console_log.txt",
    "summary": "SUCCESS",
    "results": {
        "callHtmlLogFile": "/mnt/fedmon/results/listResources/260/2016/3/31/7:43:23.652/call_log.html",
        "callHtmlLogUrl": "results/listResources/260/2016/3/31/7:43:23.652/call_log.html",
        "callLogFile": "/mnt/fedmon/results/listResources/260/2016/3/31/7:43:23.652/call_log.html",
        "callLogUrl": "results/listResources/260/2016/3/31/7:43:23.652/call_log.html",
        "callXmlLogFile": "/mnt/fedmon/results/listResources/260/2016/3/31/7:43:23.652/call_log.xml",
        "callXmlLogUrl": "results/listResources/260/2016/3/31/7:43:23.652/call_log.xml",
        "count": "24",
        "countIpv4Available": "7",
        "countIpv4Total": "31",
        "countOpenflowAvailable": "0",
        "countOpenflowTotal": "0",
        "countRawAvailable": "24",
        "countRawTotal": "113",
        "countTotal": "113",
        "countVmAvailable": "173",
        "countVmTotal": "173",
        "duration": "61529",
        "graph-desc-resrc-260-0": "Available Resources",
        "graph-desc-resrc-260-1": "Resource Overview",
        "graph-desc-resrc-260-2": "Available Resources",
        "graph-desc-resrc-260-3": "Resource Overview",
        "graph-desc-resrc-260-4": "Available Resources",
        "graph-desc-resrc-260-5": "Resource Overview",
        "graph-period-resrc-260-0": "1week",
        "graph-period-resrc-260-1": "1week",
        "graph-period-resrc-260-2": "1month",
        "graph-period-resrc-260-3": "1month",
        "graph-period-resrc-260-4": "1year",
        "graph-period-resrc-260-5": "1year",
        "graph-url-resrc-260-0": "rrd/graph-resrc-260-all-1week.png",
        "graph-url-resrc-260-1": "rrd/graph-resrc-260-detail-1week.png",
        "graph-url-resrc-260-2": "rrd/graph-resrc-260-all-1month.png",
        "graph-url-resrc-260-3": "rrd/graph-resrc-260-detail-1month.png",
        "graph-url-resrc-260-4": "rrd/graph-resrc-260-all-1year.png",
        "graph-url-resrc-260-5": "rrd/graph-resrc-260-detail-1year.png",
        "returnValue": "0",
        "rspecFile": "/mnt/fedmon/results/listResources/260/2016/3/31/7:43:23.652/rspec.txt",
        "rspecUrl": "results/listResources/260/2016/3/31/7:43:23.652/rspec.txt",
        "startTime": "1459402964407",
        "stopTime": "1459403025936"
    },
    "testbeds": [
        {
            "@id": "/api/index.php/testbed/vwall1",
            "name": "vwall1",
            "geni_id": "im-vw1",
            "urn": "urn:publicid:IDN+wall1.ilabt.iminds.be+authority+cm"
        }
    ],
    "testdefinitionname": "listResources",
    "testversionname": "prod",
    "testname": "vwall1listResources",
    "testInstanceEnabled": true,
    "expire": "2016-03-31T07:00:00+00:00"
```

7 Conclusion

This deliverable defines the third testing and integration roadmap of Fed4FIRE, targeting the third cycle of the project.

At the end of cycle 1 all testbeds (Nicta Norbit, iMinds Virtual Wall, iMinds w-iLab.t, UPMC & Inria PlanetLab Europe, FOKUS FUSECO, NTUA Netmode, UTH Nitos, NIA Koren, Ofelia (i2CAT, UBristol, iMinds), except BonFIRE and UC Smart Santander, had a compliant Aggregate Manager (AM) interface. For the BonFIRE testbeds (Inria Grid5000, EPCC) testbeds, the adaptation is not that straight forward and was partly achieved in cycle 2 and finished in cycle 3. Smart Santander uses a REST interface as it currently offers a service oriented access to the sensor measurements. For this, we introduced light federation in cycle 2 and in cycle 3 advanced federation through the YourEPM service composition framework.

The new additions for cycle 2 are the following:

- Layer 2 connectivity between multiple sites: target is to interconnect at least 3 sites by phase 1 of cycle 2 and at least 5 sites by the end of cycle 2
- Infrastructure monitoring: this comes in two flavours. Infrastructure monitoring for federation services (e.g. reputation, SLA monitoring, …) will be adopted earlier by the testbeds (at least one testbed will provide this by phase 1 of cycle 2, the others will implement this by the end of cycle 2), than the infrastructure monitoring for experimenters (which contains more and more detailed monitoring data) which is targeted for cycle 3.
- PDP (policy decision point) for FRCP: this PDP makes it possible to run OMF in a secure way. It will be implemented by 3 testbeds in phase 1 of cycle 2 and more will adopt it by the end of cycle2
- SLA management module: will be deployed centrally and at least 1 testbed will implement the testbed side of SLAs by the end of cycle 2
- Reservations in the future: about half of the testbeds will implement this in cycle 2
- Direct (IPv4 or IPv6) or indirect (through gateway) IP access without VPN: possible at all testbeds by the end of cycle 2
- Ontology based RSpec: first prototypes should be implemented by the end of cycle 2
- Services: the service directory will be ready by phase 1 of cycle 2, a couple of services by the end of cycle 2.

The new additions and improvements for cycle 3 are the following:

- YourEPM service composition framework
- Speaks-for mechanism to make it possible to deploy services speaking for users without security implications (such as providing private keys)
- More advanced SLA and reputation modules
- New Member and Slice authority APIs to improve the user experience for e.g. collaboration in experiments
- API on top of the continuous testing so the results (e.g. health score) can be used in client tools

In Fed4FIRE there is also a test suite (jFed framework) which is used for testing the different APIs. In this way developments can be tested manually, and afterwards they are tested automatically in nightly and daily tests with the same framework. The test suite now also supports setting up complex scenario testing as e.g. speaks-for testing, automated tutorial testing (through ansible scripting), and health score calculation.

References

[1] GENI AM API v3, *http://groups.geni.net/geni/wiki/GAPI_AM_API_V3*
[2] GENI AM API Acceptance Tests, http://trac.gpolab.bbn.com/gcf/wiki/AmApiAcceptanceTests
[3] Flack, GENI experimenter tool, http://www.protogeni.net/wiki/Flack
[4] Jenkins, continuous integration server, http://jenkins-ci.org/
[5] NEPI, the Network Experimentation Programming Interface, http://nepi.inria.fr
[6] OMF, cOntrol and Management Framework), http://omf.mytestbed.net/