



Grant Agreement No.: 604590
Instrument: Large scale integrating project (IP)
Call Identifier: FP7-2012-ICT-FI



eXperimental Infrastructures for the Future Internet

D5.5b: XIFI nodes operation, maintenance, assistance and procedures updates

Revision: v1.0

Work package	WP5
Task	T5.3, T5.4
Due date	30/09/2015
Submission date	30/09/2015
Deliverable lead	ORANGE
Authors	Aimilia Bantouna (UPRC), Angelos Rouskas (UPRC), Antonio Fuentes (Red.es/RedIRIS), Aristi Galani (UPRC), Bastien Putegnath (Com4innov), Bernd Bochow (Fraunhofer), Claude Hary (Com4innov), Cristian Cristelotti (TN), Demetrios Kelaïdonis (UPRC), Fernando López (TID), Gábor Vattay (WIGNER), Genci Tallabaci (TN), Georgios Poullos (UPRC), Jacek Kochan (PSNC), Jan Kandrát (CESNET), Joaquin Iranzo (ATOS), Konstantinos Tsagkaris (UPRC), Marios Logothetis (UPRC), Panagiotis Demestichas (UPRC), Panagiotis Vlacheas (UPRC), Philippe Badia (Com4innov), Radek Velc (CESNET), Riwal Kerherve (ILB), Rudolf Vohnout (CESNET), Sándor Laki (WIGNER), Seán Murphy (ZHAW), Theofanis Katsiaounis (Neuropublic), Thomas Günther (Fraunhofer), Thierry Milin (Orange), Uwe Herzog (Eurescom), Vassileios Foteinos (UPRC), Vera-Alexandra Stavroulaki (UPRC)
Reviewers	Rudolf Vohnout (CESNET) , Claude Harry (Com4Innov)

Abstract	This deliverable provides an overview of the different activities related to the running of the XIFI nodes (Consortium's members and Associated partners), an update of the procedures that are used by XIFI and proposed to FI-Core project. It also describes support to FI-Developers.
Keywords	Nodes operation, OLA, maintenance, support, helpdesk

Document Revision History

Version	Date	Description of change	List of contributor(s)
V1.0	10/04/2015	Final and reviewed document version	Thierry Milin (Orange) et al.
D5.5b V1.0	30/09/2015	Addition of some statistics on performed operation from M24 to M30	Thierry Milin (Orange) et al.

Disclaimer

This report contains material which is the copyright of certain XIFI Consortium Parties and may only be reproduced or copied with permission in accordance with the XIFI consortium agreement.

All XIFI Consortium Parties have agreed to publication of this report, the content of which is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License¹.

Neither the XIFI Consortium Parties nor the European Union warrant that the information contained in the report is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using the information.

Copyright notice

© 2013 - 2015 XIFI Consortium Parties

Project co-funded by the European Commission in the 7 th Framework Programme (2007-2013)		
Nature of the Deliverable:		O (Other)
Dissemination Level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to bodies determined by the XIFI project	
CO	Confidential to XIFI project and Commission Services	

¹http://creativecommons.org/licenses/by-nc-nd/3.0/deed.en_US

EXECUTIVE SUMMARY

This deliverable provides an overview of the different activities related to the running of the XIFI nodes and updates the procedures initially defined in Deliverables D5.1 and D5.3.

D5.1 and D5.3 had defined first versions of the operational and technical installation procedures and protocols that a new infrastructure has to implement and to follow in order to join the XIFI federation. Furthermore, they also contained requirements and initial drafts of general procedures and procedures for developer support and maintenance.

The extension of the federation is succeeded in principle. Issues may arise inherently due to the complex distributed operations and are constantly resolved as a part of the regular node maintenance process. The initial set of nodes was formed with nodes from five of the XIFI partners that were part of the project from the start, located in Berlin (Germany), Waterford (Ireland), Brittany (France), Seville/Malaga (Spain) and Trento (Italy). Following the XIFI Open Call, managed by the Federation Office, 11 further nodes provided by the 12 new XIFI partners have been added to the federation. Due to bankruptcy, one new partner was not active during the project and was not federated.

The federation is continuously growing with Associated Partners that request the Federation office to join without additional financing from the Future Internet PPP. At the moment of writing this deliverable, IntelliCloud (Crete), InfoTech (Mexico) joined the federation and offer their resources to FIWARE Lab users whereas University of Messina (Italy) is completing the federation process. Wroclaw University of Technology (Poland) has been just formally accepted by the Office as associated partners to join the federation.

At URL <http://infographic.lab.fi-ware.org/> the current resource status of the XIFI federation is displayed.

On 30th March 2015, the XIFI federation has 8407 federated users, 1744 organizations, 16 federated regions, 2720 CPU cores, 7944 GB RAM and 176 TB of hard disk space with 1485 operational Virtual Machines associated to the federation.

Gained from the experience of several months of operation and maintenance from the XIFI partners, the general procedures that are required to execute the procedures for maintenance and user / developer support are updated in this document from previous deliverable [7]. The update mainly relates to the identification and assignment of operational roles, but also including some refinements. For example, for each of the nodes the specific persons who take care of node help desk, system administrator or network administrator etc. were defined along with their contact details and availability (support hours). In terms of developer or infrastructure support e.g. the members of the Level 1 helpdesk team and their tasks and responsibilities were defined. Level 1 helpdesk will e.g. be the initial contact point for all incoming tickets that are not directly assigned to a node, FIWARE Ops (i.e. the former FI-Ops) tool or GE. The persons assigned to infrastructure support will take care of supporting the operation of the nodes of the federation. Software Component Support is provided by the persons in charge of the respective Software Components of the XIFI federation tool suite (FIWARE Ops).

This document also provides an update of the procedures for operating the federation and will serve as the standard and binding reference for FI-Core and future FI projects. The stakeholders and roles definitions done on a general level for XIFI before have been reviewed and refined in the scope of operational level agreements, node and federation operations and federation maintenance.

Infrastructures provide resources to the federation to enable the federation to commit on service level agreements (SLAs) between the federation and its users. Upon joining the federation, a new node steps into an operational level agreement by accepting the terms and conditions set in place by the federation authority equally for all infrastructure nodes (with distinct parameters for prospective master nodes). These parameters are formulated in terms of minimum requirements regarding network bandwidth, computing resources etc.

A viable federation of IT resources and services maintains a number of SLA's that serve as means for quantitative evaluation of service invocations by the users (developers in case of XIFI). Both SLA and OLA should be seen as evolving frameworks following certain maturity (capability).

Mapping the OLA concept onto XIFI brings however certain complexities, since here we are faced with a federation of independent organizations and not just sections and teams inside a single organisation. It is proposed to implement federation OLA as a set of uniform federation-wide policies (e.g. based on the federation utility preservation and increase) and their mapping to service-specific policies for all services offered by a federation. A federated cloud infrastructure is an attractive option for multiple providers to join their resources and to appear to their customers as a (federated) provider of virtually unlimited capacities.

Finally, regarding operational requirements and procedures, policy for identity management, policy for resource management and resource management implementation to solve uncontrolled consumption of resources are defined. A number of procedures are defined that are intended to describe the Infrastructure Owner operations. This includes e.g. tenant deployment, tenant life cycle, traceability of deployed instances, Openstack release upgrade, etc. These definitions are built on the experience of XIFI partners gained from running FIWARE Lab. This information complements the information in the handbook Deliverables D2.1 and D2.4.

This Deliverable defines also in great detail the maintenance process. This is dedicated to the execution of proactive and reactive maintenance activities to ensure that services provided to developers are continuously available and conform to SLA or QoS performance levels. First, relevant stakeholders are identified with a particular view on their role and obligations in the maintenance process. Specific person are identified that have taken over this role in XIFI, e.g. for the federation maintenance contact or the infrastructure maintenance contact for each node. Next, all components that are subject to the maintenance process are identified along with their location, owner and maintainer. Finally, in order to implement the maintenance process a number of procedures have been defined and specified to some detail. The procedures described focus on infrastructure maintenance, software maintenance and collaboration among nodes.

XIFI utilizes the JIRA helpdesk and in particular it's ticketing system for both the interaction between maintenance stakeholder and developers as well as between maintenance stakeholders in the internal maintenance process. Ticket handling in the interaction flow between maintainers and developers is also defined.

In the last section before the conclusion the process of providing support to FI-developers is defined. The description of support to FI-developers is structured according to escalation levels, i.e. Level-0 /-1 / -2 / -3 support. It is based on and consistent with the basic procedures defined in section 3 "Update on general procedures".

All 4 support levels are introduced first: Level 0 support provides automated or self-service solutions; Level 1 support filters incoming Help Desk requests and provides basic support and may forward to higher escalation levels; Level 2 generally handles break/fix, configuration issues and does troubleshooting; while finally Level 3 support provides specialized troubleshooting, configuration, database administration, and repair – in the scope of FIWARE Lab relating to GEi and FIWARE Ops support.

A few flow diagrams are provided that show the interactions between the support levels. A table lists the assignment of experts to the various roles involved in the FI-developer support process.

In terms of tool support, XIFI decided to use the JIRA platform setup by FIWARE for organising the handling of support requests and their processing. The JIRA platform is hosted in one of the servers of the FIWARE Lab infrastructure, currently in the Spain node.

JIRA collectors has been developed and linked from in the FIWARE Lab (help/contact) homepage and GE catalogue in order to collect the support requests. JIRA is thus used as a joint and unique interface to all users (FI-Developers) of FIWARE Lab, as well as for coordinating the maintenance process within the federation.

An analysis of the requests received from the developers sent to fiware-lab_help list and the JIRA ticket created is detailed. In order to get an idea of the effort that is required for processing the tickets, these requests are examined. The conclusion is that the level of workload for L1-Helpdesk and the nodes for supporting developers is manageable.

This deliverable, as agreed with commission, is an extended version of the D5.5 including additional statistics on the operations performed during the extension phase from M24 to M30. The additional content is included in Section 7, the rest of the deliverable is untouched.

The document finishes with some conclusions.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	3
TABLE OF CONTENTS	6
LIST OF FIGURES	13
LIST OF TABLES	16
ABBREVIATIONS	19
1 INTRODUCTION	21
1.1 Context, Objective and Scope of this Deliverable	21
1.2 Intended Audience and Reading Suggestions	22
2 REPORT ON XIFI NODES OPERATION, MAINTENANCE, AND ASSISTANCE ..	24
2.1 Introduction	24
2.1.1 FIWARE Lab and XIFI Federation Integration	25
2.1.2 Federation Monitoring of Resources and Operations	26
2.2 Waterford Node	27
2.2.1 Description	27
2.2.2 Experience on Support and Maintenance	27
2.2.3 Current Status	28
2.2.4 OpenStack Configuration	29
2.2.5 User-base	30
2.3 Trento Node	31
2.3.1 Description	31
2.3.2 Experience on Support and Maintenance	31
2.3.3 Current Status	32
2.3.4 OpenStack Configuration	32
2.3.5 User-base	33
2.4 Berlin Node	34
2.4.1 Description	34
2.4.2 Experience on Support and Maintenance	35
2.4.3 Current Status	35
2.4.4 OpenStack Configuration	35
2.4.5 User-base	36
2.5 Brittany Node (Lannion)	36
2.5.1 Description	36
2.5.2 Experience on Support and Maintenance	36
2.5.3 OpenStack Configuration	38
2.5.4 User-base	38

2.6	Spain node (Seville/Malaga).....	39
2.6.1	Description.....	39
2.6.2	Experience on Support and Maintenance	41
2.6.3	Current Status	41
2.6.4	OpenStack Configuration	42
2.6.5	User-base	43
2.7	Prague (CESNET) Node.....	44
2.7.1	Description.....	44
2.7.2	Experience on Support and Maintenance	46
2.7.3	Current Status	46
2.7.4	OpenStack Configuration	46
2.7.5	User-base	47
2.8	Gent (IMINDS) Node.....	47
2.8.1	Description.....	47
2.8.2	Experience on Support and Maintenance	48
2.8.3	Current Status	48
2.8.4	OpenStack Configuration	49
2.8.5	User-base	50
2.9	Zurich (ZHAW) Node	50
2.9.1	Description.....	50
2.9.2	Experience on Support and Maintenance	52
2.9.3	Current Status	52
2.9.4	OpenStack Configuration	52
2.9.5	User-base	52
2.10	Poznan (PSNC) Node	53
2.10.1	Description.....	53
2.10.2	Experience on Support and Maintenance	53
2.10.3	Current Status	53
2.10.4	OpenStack Configuration	53
2.10.5	User-base	54
2.11	PiraeusN (Neuropublic) Node	54
2.11.1	Description.....	54
2.11.2	Experience on Support and Maintenance	56
2.11.3	Current Status	57
2.11.4	OpenStack Configuration	58
2.11.5	User base.....	58
2.12	PiraeusU (UPRC) Node.....	58

2.12.1	Description.....	58
2.12.2	Experience on Support and Maintenance	59
2.12.3	Current Status	60
2.12.4	OpenStack Configuration	60
2.12.5	User-base	60
2.13	Volos (UTH) Node	61
2.13.1	Description.....	61
2.13.2	Experience on Support and Maintenance	63
2.13.3	Current Status	63
2.13.4	OpenStack Configuration	63
2.13.5	User-base	64
2.14	Sophia Antipolis (Com4Innov) Node	64
2.14.1	Description.....	64
2.14.2	Experience on Support and Maintenance	65
2.14.3	Current Status	65
2.14.4	OpenStack Configuration	66
2.14.5	User-base	67
2.15	Karlskrona (BTH-Sweden) Node	67
2.15.1	Description.....	67
2.15.2	Experience on Support and Maintenance	68
2.15.3	Current Status	68
2.15.4	OpenStack Configuration	68
2.15.5	User-base	68
2.16	ACREO Swedish ICT Node	69
2.16.1	Description.....	69
2.16.2	Experience on Support and Maintenance	69
2.16.3	Current Status	69
2.16.4	OpenStack Configuration	70
2.16.5	User-base	70
2.17	Budapest (WIGNER) Node	70
2.17.1	Description.....	70
2.17.2	Experience on Support and Maintenance	72
2.17.3	Current Status	72
2.17.4	OpenStack Configuration	73
2.17.5	User-base	73
2.18	Intellicloud – Crete (Associated Partner).....	74
2.18.1	Description.....	74

2.18.2	Experience on Support and Maintenance	74
2.18.3	Current Status	75
2.18.4	OpenStack Configuration	75
2.18.5	User-base	76
2.19	Infotec Mexico (Associated Partner)	76
2.19.1	Description.....	76
2.19.2	Experience on Support and Maintenance	77
2.19.3	Current Status	78
2.19.4	OpenStack Configuration	80
2.20	University of Messina – IT (Associated Partner)	81
2.20.1	Description.....	81
2.20.2	Current Status	81
2.20.3	OpenStack Configuration	82
2.20.4	User-base	82
2.21	Wroclaw University of Technology – Poland (Associated Partner).....	82
2.22	Report on Assistance and Support	82
3	UPDATE ON GENERAL PROCEDURES.....	84
3.1	Management of Nodes	84
3.1.1	Berlin	84
3.1.2	Brittany	85
3.1.3	Spain Node.....	85
3.1.4	Trento.....	86
3.1.5	Waterford.....	86
3.1.6	IMINDS	86
3.1.7	ZHAW	87
3.1.8	PSNC	87
3.1.9	Neuropublic	87
3.1.10	CESNET	87
3.1.11	UPRC	88
3.1.12	Com4Innov	88
3.1.13	ACREO Swedish ICT	88
3.1.14	WIGNER	89
3.1.15	UTH	89
3.1.16	BTH	89
3.1.17	Intellicould (Crete) – Associated Partner.....	89
3.1.18	Infotec (Mexico) – Associated Partner	90
3.1.19	University of Messina (Italy) – Associated Partner	90

3.1.20	Wroclaw University of Technology (Poland) – Associated Partner.....	90
3.2	Developer Support	90
3.3	Infrastructure Support	91
4	UPDATES ON PROCEDURES FOR OPERATING THE FEDERATION.....	93
4.1	Stakeholders and Roles in Establishing and Maintaining Operational Level Agreements.....	93
4.2	Update on Support Process and Procedures for Joining the Federation	93
4.3	Scope and Purpose of Operational Level Agreements	94
4.4	Operational Level Agreements	96
4.5	OLA implementation in XIFI	97
4.5.1	OLA Purpose	97
4.5.2	Root Cause Adaptation (RCA) Mechanisms	98
4.5.3	Operational Definitions for OLA.....	99
4.5.4	Service Policy	99
4.5.5	OLA Policies	99
4.5.6	OLA Implementation Status	102
4.6	Operational Requirements and Procedures	102
4.6.1	Policy for Identity Management in FIWARE Lab.....	103
4.6.2	Policy for Resource Management in FIWARE Lab	107
4.6.3	Resource Management Implementation Status.....	110
4.6.4	Naming of Networks.....	113
4.6.5	DNS Service	116
4.6.6	Tenant Deployment	117
4.6.7	Basic Tenant Deployment Procedure	118
4.6.8	Tenant Life Cycle	124
4.6.9	Traceability of Deployed Instances	125
4.6.10	Local catalogue management.....	128
4.6.11	Managing Images.....	130
4.6.12	Managing Blueprints	134
4.6.13	Use Case Handling	138
4.6.14	Tenant Customization.....	139
4.6.15	Node Administration	141
4.6.16	Openstack Release Upgrade	144
5	MAINTENANCE PROCESS	152
5.1	Relation to the eTOM Framework Objectives.....	152
5.2	Stakeholders.....	152
5.3	Stakeholder Interaction through the Help-desk	158
5.3.1	Interaction of Maintainers and Developers.....	158

5.3.2	Interaction of Maintainers.....	159
5.4	Sub-systems Subject to Maintenance	160
5.4.1	Infrastructure Node	161
5.4.2	Communication Infrastructure	162
5.4.3	Software Components.....	162
5.4.4	Software Sub-systems.....	165
5.4.5	Procedures of the Maintenance Process.....	167
5.4.6	Scheduled Maintenance (Single Infrastructure Node).....	169
5.4.7	Scheduled Maintenance (Multiple Infrastructure Nodes).....	171
5.4.8	Unscheduled Maintenance (Single Infrastructure Node).....	173
5.4.9	Review of the Maintenance Procedures.....	173
5.4.10	Software Repository	176
6	SUPPORT TO FI-DEVELOPERS.....	185
6.1	Introduction to Support Levels	185
6.2	Support Levels Applied in FI-WARE Lab Support.....	186
6.3	Helpdesk Process Flows and Interaction with FI developers	186
6.4	JIRA Ticketing Process	188
6.5	Responsibilities.....	192
6.6	JIRA Administration.....	192
6.7	Reporting (JIRA statistics).....	195
6.8	FAQ and Beginners Guide.....	197
7	REPORT ON ACTIVITIES PERFORMED DURING THE EXTENSION PERIOD (M24-M30).....	198
7.1	Level 1 Helpdesk support	198
7.2	Community Account management	200
7.3	Node performances (Karma points).....	202
7.3.1	Berlin	203
7.3.2	Brittany	204
7.3.3	Budapest	204
7.3.4	Crete.....	205
7.3.5	Gent.....	206
7.3.6	Karlskrona.....	207
7.3.7	Piraeus.....	208
7.3.8	Piraeus.....	209
7.3.9	Poznan.....	210
7.3.10	Prague	211
7.3.11	SophiaAntipolis	212

7.3.12	Spain	213
7.3.13	Stockholm	214
7.3.14	Trento	215
7.3.15	Volos	216
7.3.16	Waterford	217
7.3.17	Zurich	218
8	CONCLUSIONS	219
	REFERENCES	220
	APPENDIX A OPERATIONAL LEVEL AGREEMENTS	224
A.1	Operational Level Agreements	224
A.2	OLA Level 3 Rationale	224
A.3	Workflows for Cross-layer Optimisation	227
A.4	Security Benefits	228
A.5	How to Trust by Workflow	230
A.6	Future work: Workflow Manifesto	234
	APPENDIX B FURTHER DETAILS ON OLA	236
B.1	OLA Scheme Description	236
B.2	OLA Computing and Storage Resources Operation and Maintenance	238
B.3	OLA Network Connectivity Operation & Management	239
B.4	OLA Non-conventional Resources	240
B.5	OLA User Support	241
B.6	OLA Federation Services and Software Management	242
B.7	OLA Security & Privacy	243
	APPENDIX C PROCEDURE TO ADD THE REQUIRED IMAGES	245

LIST OF FIGURES

Figure 1: FIWARE Lab User Cloud portal	25
Figure 2: Operational capacity details on 30 th March 2015	26
Figure 3: Example of regional cloud services status	27
Figure 4: Routing table of WIT	28
Figure 5: TN architecture	33
Figure 6: TN external connectivity target	33
Figure 7: SpainNode – network connectivity	40
Figure 8: SpainNode - resource distribution	40
Figure 9: SpainNode – project quota list	43
Figure 10: Graphical Scheme of the Prague node	45
Figure 11: Integrated Facility Central Bohemia	46
Figure 12: Architecture of Neuropublic node	56
Figure 13: Server equipment UTH node	63
Figure 14: Server equipment Sophia Antipolis node	66
Figure 15: Server equipment BTH node	68
Figure 16: Architecture of Budapest node	71
Figure 17: Federation support procedures	94
Figure 18: Generic root cause adaptation scheme	98
Figure 19: Mapping between service and utility KPI's	100
Figure 20: Separation of concerns in KPI mapping	101
Figure 21: OLA via uniform KPI's	101
Figure 22: Tenant user identification	103
Figure 23: Identity management	105
Figure 24: Floating IP database	106
Figure 25: Change of network name on provisioned networks 1	114
Figure 26: Change of network name on provisioned networks 2	115
Figure 27: Split DNS	116
Figure 28: DNS-as-a-Service	116
Figure 29: Account part	119
Figure 30: Create a network	119
Figure 31: Create a router	120
Figure 32: Add an interface	120
Figure 33: Create Security Group	120
Figure 34: Add rules	121
Figure 35: Define the Keypairs	121

Figure 36: Create an instance	121
Figure 37: Launch an instance	122
Figure 38: Instance Log.....	122
Figure 39: Allocate IP	123
Figure 40: Associate IP	123
Figure 41: Ping	123
Figure 42: Connect via SSH.....	124
Figure 43: Create blueprint template.....	134
Figure 44: Adding tier(s) to a blueprint template.....	135
Figure 45: Create a tier.....	135
Figure 46: Adding software to a tier	136
Figure 47: Selecting the menu to change the software attributes	136
Figure 48: Editing the software attributes	137
Figure 49: Launch a blueprint template	137
Figure 50: Blueprint instances.....	138
Figure 51: Manage tenant – select user	142
Figure 52: Modifications on a tenant	143
Figure 53: Modifications on a tenant II.....	143
Figure 54: Sample Maintenance Stakeholder Interaction (Developer initiated issue request on a sub-system issue)	157
Figure 55: Maintenance ticket in Jira.....	160
Figure 56: Management of maintenance procedures (most relevant cases of the management process)	168
Figure 57: Outline of a scheduled maintenance procedure affecting a single infrastructure node.....	170
Figure 58: Outline of scheduled federation-wide maintenance procedure affecting multiple infrastructure nodes	172
Figure 59: Continuous integration workflow - Overview	176
Figure 60: VM image details.....	178
Figure 61: Artifactory Web Interface.....	179
Figure 62: Outline of an unscheduled maintenance procedure affecting a single infrastructure node.....	184
Figure 63: FIWARE Lab Level 1 helpdesk support process.....	188
Figure 64: Pointer to Helpdesk on FIWARE Lab “Help & Info” page ⁶	189
Figure 65: Ticket status tree	191
Figure 66: Priority of Jira tickets.....	191
Figure 67: Issue creation	193
Figure 68: Different email lists in HELP-DESK.....	194
Figure 69: Issues created in the last 30 days	194
Figure 70: Activity stream.....	195

Figure 71: Workflow of the HELP-DESK issues	195
Figure 72: Number of support requests received / Jira tickets created.....	196
Figure 73: Karma history – Berlin Node.....	203
Figure 74: Karma history – Brittany Node.....	204
Figure 75: Karma history – Budapest Node.....	204
Figure 76: Karma history – Crete Node	205
Figure 77: Karma history – Gent Node	206
Figure 78: Karma history – Karlskrona Node	207
Figure 79: Karma history – Piraeus Node.....	208
Figure 80: Karma history – Piraeus Node.....	209
Figure 81: Karma history – Poznan Node	210
Figure 77: Karma history – Prague Node.....	211
Figure 83: Karma history – Sophia Antipolis Node.....	212
Figure 84: Karma history – Spain Node.....	213
Figure 85: Karma history – Stockholm Node	214
Figure 86: Karma history – Trento Node	215
Figure 87: Karma history – Volos Node	216
Figure 88: Karma history – Waterford Node	217
Figure 89: Karma history – Zurich Node	218
Figure 90: Pegasus WF mapping	226
Figure 91: Cross-layer optimisation in D-CAF.....	228
Figure 92: The scope of WAC	229
Figure 93: Workflow as a Unit of Trust in OLA.....	231
Figure 94: Sample workflow.....	231

LIST OF TABLES

Table 1: Status of the sanity check of the Regions on Friday, 13 th March, 2015	25
Table 2: Current WIT node nova flavor-list.....	29
Table 3: Waterford default quotas.....	30
Table 4: Waterford network resource quotas	30
Table 5: WIT node utilization	30
Table 6: TN node utilization	34
Table 7: List of tenants on the Berlin node	36
Table 8: Lannion quota-defaults	38
Table 9: List of tenants on the Lannion node.....	39
Table 10: Resume of usage in the Spain node.....	44
Table 11: List of tenants on the Spain node	44
Table 12: CESNET Usage in August 2014.....	47
Table 13: Gent flavour list	49
Table 14: Gent quota defaults	50
Table 15: iMinds node utilization	50
Table 16: ZHAW system capacity	50
Table 17: root@node-1:~# neutron --os-region Zurich subnet-list	51
Table 18: Default quotas Zurich nodes	51
Table 19: List of tenants on the Zurich node.....	53
Table 20: List of tenants on the PSNC node	54
Table 21: List of tenants on the Neuropublic node	58
Table 22: List of tenants on the UPRC node.....	61
Table 23: List of tenants on the UTH node.....	64
Table 24: Sophia Antipolis infrastructure	65
Table 25: Default quotas Sophia Antipolis node.....	67
Table 26: Tenants Sophia Antipolis node	67
Table 27: Tenants BTH node	69
Table 28: Tenants ACRO node	70
Table 29: Default quotas Budapest node.....	73
Table 30: Tenants Budapest node	73
Table 31: Default quotas Crete node.....	76
Table 32: Tenants Crete node.....	76
Table 33: Server equipment Mexican node.....	79
Table 34: Big-data-specific Server equipment Mexican node	79
Table 35: OpenStack configuration Mexican node.....	80

Table 36: New images configurations Mexican node	80
Table 37: OpenStack equipment Mexican node.....	81
Table 38: Berlin contact details.....	85
Table 39: Brittany contact details.....	85
Table 40: Spain contact details.....	86
Table 41: Trento contact details	86
Table 42: Waterford contact details	86
Table 43: IMINDS contact details	86
Table 44: ZHAW contact details.....	87
Table 45: PSNC contact details.....	87
Table 46: Neuropublic contact details.....	87
Table 47: CESNET contact details.....	88
Table 48: UPRC contact details	88
Table 49: Com4Innov contact details.....	88
Table 50: ACREO contact details	88
Table 51: WIGNER contact details.....	89
Table 52: UTH contact details	89
Table 53: BTH contact details.....	89
Table 54: Intellicloud contact details	90
Table 55: Infotec contact details	90
Table 56: University of Messina contact details	90
Table 57: Wroclaw University of Technology contact details	90
Table 58: Infrastructure support team	92
Table 59: OLA categories	96
Table 60: Resource management strategy –issues identified.....	108
Table 61: Resource management policies description	109
Table 62: Resource management mechanisms and processes.....	110
Table 63: Output of nova list --all-tenants	126
Table 64: Output of nova volume-list --all-tenants	126
Table 65: Output of nova show 9389febd-bcc2-4e2f-83ba-c4c9356dd211 , where 9389febd-bcc2-4e2f-83ba-c4c9356dd211 is an instance ID	127
Table 66: output of nova volume-show 7c615192-4020-4521-9120-827946cec4db , where 7c615192-4020-4521-9120-827946cec4db is an instance ID.....	128
Table 67: Property keys.....	130
Table 68: Set of images currently synchronised	131
Table 69: Openstack default quota.....	140
Table 70: Openstack default flavours.....	140

Table 71: Lannion usage list	144
Table 72: VM list	144
Table 73: Node Openstack release upgrade	151
Table 74: Federation Maintenance Contact.....	153
Table 75: Infrastructure maintenance contact	154
Table 76: Infrastructure Maintenance Escalation Levels	161
Table 77: Communication Infrastructure Maintenance Escalation Levels	162
Table 78: Components under Maintenance.....	165
Table 79: Sub-systems under Maintenance.....	167
Table 80: XIFI software repository hosts and services	178
Table 81: Overall responsibility assignment	192
Table 82: Level 1 Helpdesk team.....	192
Table 83: Overview of status of all tickets assigned to L1-Helpdesk and L2-nodes-helpdesks	196
Table 84: Level 1 Helpdesk schedule	199
Table 85: Number and status of all tickets received during the extension phase	200
Table 86: FIWARE Community Account upgrades as of 30 th September 2015.....	201
Table 87: Workflow (maintenance procedure) creation process.....	232
Table 88: Workflow registration process	233
Table 89: Separation of concerns between the major roles	234

ABBREVIATIONS

ACL	Access Control List
BCP	Best Current Practices
BGP	Border Gateway Protocol
CB	Orion Context Broker
DC	Data Centre
D-CAF	Distributed Context Aware Firewall
DCRM	Data Centre Resource Management
DEM	Datacentre and Enablers Monitoring Adapter
DHCP	Dynamic Host Configuration Protocol
DMS	Domain Name Service
FI-PPP	Future-Internet Private Public Partnership
FTTH	Fibre To The Home
G2	Generation 2
GB	Gigabyte
GE	Generic Enabler
GEi	Generic Enabler Instance
HA	High Availability
HPC	High Performance Computing
IaaS	Infrastructure as a Service
IdM	Identity Manager
IO	Infrastructure Owner
IoT	Internet of Things
IP	Internet Protocol
iSCSI	Internet Small Computer System Interface
ITIL	Information Technology Infrastructure Library
KVM	Kernel-based Virtual Machine
L3-VPN	Layer 3 Virtual Private Network
LUN	logical unit number
LVM	Logical volume management
MBGP	Multiprotocol BGP layer 2 and layer 3 VPNS
MD-VPN	Multi Domain – Virtual Private Network
NAM	Network Active Monitoring
NFS	Network File System
NGSI	Next Generation Service Interfaces
NPM	Network Passive Monitoring
NREN	National Research and Education Network
NRPE	Nagios Remote Plugin Executor
OF	OpenFlow
OLA	Operational Level Agreement
OVS	Open vSwitch
PA	Provider-aggregatable
PaaS	Platform as a Service

PE	Provider Edge router
PI	Provider-Independent
PXE	Pre-Execution Environment
QoE	Quality of Experience
QoS	Quality of Service
RBAC	Role-Based Access Control
RED	Random Early Detection
RT	Request Tracker
SaaS	Software as a Service
SAML	Security Assertion Markup Language
SDC	Software Deployment & Configuration
SDN	Software Defined Networking
SDR	Software Defined Radios
SE	Specific Enabler
SEM	Security Event Management
SFA	Slice-based Federation Architecture
SIEM	Security Information Event Management
SIM	Security Information Management
SME	Small Medium Enterprise
TCP	Transmission Control Protocol
UC	Use Case
VLAN	Virtual Local Area Network
VM	Virtual Machine
VPN	Virtual Private Network
VRF	Virtual Routing Function
WAC	Workflow-based Access Control
WF	Workflow
WP	Work Package
WSAN	Wireless Sensor and Actuator Network
XIMM	XIFI monitoring middleware
Zabbix	Enterprise-class software for monitoring of networks, hardware and applications
ZFS	Zettabyte File System

1 INTRODUCTION

1.1 Context, Objective and Scope of this Deliverable

The XIFI platform is the community cloud for European FI-PPP developers, enabled by the advanced FI infrastructures in Europe. As such XIFI offers a marketplace, enabling large-scale trials. The marketplace provides access to Generic Enablers (GEs) developed by FIWARE and to Specific Enablers (SEs) developed by FI-PPP Phase 2 Use Case projects through a highly available and reliable "federation" of infrastructures.

The XIFI project has the objectives of setting up and operating a Future Internet federation, mitigating the limitations of the existing fragmented infrastructures within Europe, and to cope with large trial deployments. The federation is formed by integrating heterogeneous test infrastructures throughout Europe. To construct the federation, infrastructures are required to follow common and consistent procedures and protocols in their operations inside the federation so that a new infrastructure can join the federation with minimum effort and minimum potential for conflicts within existing operations.

The initial set of nodes was formed with nodes from five of the XIFI partners that were part of the project from the start, located in Berlin (Germany), Waterford (Ireland), Brittany (France), Seville/Malaga (Spain) and Trento (Italy).

Following the XIFI Open Call, managed by the Federation Office, 11 further nodes provided by the 12 new XIFI partners have been added to the federation. Due to bankruptcy, one new partner was not active during the project and was not federated. The federation is continuously growing with Associated Partners that request the Federation office to join without additional financing from the Future Internet PPP. At the moment of writing this deliverable, IntelliCloud (Crete), InfoTech (Mexico) joined the federation and offer their resources to FIWARE Lab users whereas University of Messina (Italy) is completing the federation process.

Wroclaw University of Technology (Poland) has been just formally accepted by the Office as associated partners to join the federation.

A first set of procedures and protocols for XIFI federation has been initially published as Deliverable D5.1 [5] and updated at month 18 in Deliverable D5.3 to define the operational and technical installation procedures and protocols that a new infrastructure has to implement and to follow in order to join the XIFI federation.

Furthermore, this document updates Deliverable D5.3 with mainly:

- the description and the status of each node (from the consortium and Associated partners)
- a short report on assistance and support received from FI-Developers,
- an update on general procedures for new nodes,
- the description of the Operational Level Agreement implementation in XIFI,
- the policy for identity management and the policy for resource management that was necessary to setup to solve excessive and uncontrolled consumption of resources (VM, public IPn, etc.) by FI-Developers
- the detailed definition of the procedures to implement the maintenance process,
- and finally with some details and figures to support FI-Developers.

In September, a section to give some statistics on performed operations during the extension phase from M24 to M30 was added.

The insights gained have been feeding into the preparation of D5.3 and also this deliverable. While it provides an update to the procedures and protocols defined initially in D5.1 and refined in D5.3, it contains significant enhancements e.g. regarding operations and support to FI-Developers.

The experience gained in the past months was also feeding into D5.6 which is being prepared in parallel to D5.5. D5.6 has a clear focus on procedures for new nodes joining the federation and as such D5.5 and D5.6 complement each other.

1.2 Intended Audience and Reading Suggestions

The target audience of this deliverable is:

- The XIFI federation office, in order to evaluate whether a candidate infrastructure meets the minimum technical and operational requirements (in conjunction with D5.1);
- All nodes that have joined or plan to join the XIFI federation, for information regarding the installation procedures and protocols; maintenance, operation and user support procedures
- Experts and technical personnel providing deployment support and end-user support activities. These activities will be fulfilled by the support entity of the XIFI federation;
- Developers and maintenance experts of XIFI tools and FI services who will apply the procedures and use the protocols for the maintenance of the XIFI tools and FI services, hosted by the nodes.
- Designers of SLA and OLA rules and requirements for federated cloud platforms, even beyond XIFI.
- PPP Future Internet members and precisely FI-Core project, FI-Accelerators and Phase 3 projects.

The document is structured as follows:

- **Section 2** provides a short report on operation, maintenance, and assistance for each of the nodes that have joined the federation so far. This section also briefly reports on how the support has been organised so far.
- **Section 3** gives an update of the general procedures defined in D5.1 and completed in D5.3. Mainly, it identifies the responsible persons of all XIFI nodes (node help desk, node manager etc.) and summarises their tasks. The section also lists the roles that were defined and instantiated for developer and infrastructure support, and briefly summarises the tasks and interaction between the roles.
- **Section 4** gives an updates on procedures for operating the federation. It starts with a definition of the stakeholders involved, and while it briefly addresses the Support Process and Procedures for joining the federation (leaving the details for D5.6), the main focus is on Operational Level Agreements.
- **Section 5** defines the maintenance process. It defines in detail the various types of maintenance, the processes for conducting them and specifies the respective process flow.
- **Section 6** describes how support to FI developers is organised (helpdesk).
- **Section 7** gives some statistics on performed operations during the extension phase from M24 to M30.
- **Conclusions** are finally given in section 8.

Protocols and procedures have evolved during the project and are stable now, fed by experience gained from adding further (heterogeneous) nodes and from operating and maintaining a growing federation getting more complex.

The definitions and specifications contained in this document will be transferred to FI-Core to serve as the standard and binding reference.

2 REPORT ON XIFI NODES OPERATION, MAINTENANCE, AND ASSISTANCE

2.1 Introduction

The primary aim of XIFI is to establish a federation of infrastructures that serves as a platform for FI developers and their experiments and projects. The original nodes of this federation were located at Waterford, Berlin, Lannion, Spain (Seville/Malaga) and Trento. These nodes would follow a Master/Slave hierarchy with the Master nodes located, ultimately, in Trento and Seville/Malaga.

The continuous roll out of the MD-VPN links across the federation denotes the steady growth of the cloud across Europe. New nodes selected through the Open Call joined the Federation completed by Associated nodes (ie without any financial support from the PPP) to offer resources to federation users.

Each individual node consisted of heterogeneous hardware architecture running OpenStack cloud Management Platform alongside XIFI subsystems, that include Monitoring, Security, Deployment and Operations, as well as User-orientated and GUI Subsystems.

This section give more details (description, experiences on support and maintenance, current status, openstack configuration and user-base) for nodes that passed the automated tests (Table 1) managed by FIWARE Lab at the time of writing this document. The full description of the tests and the detailed results are available in deliverable D5.6 [8].

Region	Fail	Error	Skip	Success	Fail
tests.regions.test_poznan.PoznanTestSuite	0	3	0	13	16
tests.regions.test_crete.CreteTestSuite	0	0	0	16	16
tests.regions.test_gent.GentTestSuite	0	11	0	5	16
tests.regions.test_zurich.ZurichTestSuite	0	2	0	14	16
tests.regions.test_sophiaantipolis.SophiaAntipolisTestSuite	0	16	0	0	16
tests.regions.test_piraeusu.PiraeusUTestSuite	0	0	0	16	16
tests.regions.test_berlin.BerlinTestSuite	0	0	0	16	16
tests.regions.test_piraeusn.PiraeusNTestSuite	0	0	0	16	16
tests.regions.test_waterford.WaterfordTestSuite	0	3	0	13	16
tests.regions.test_budapest.BudapestTestSuite	0	2	0	14	16
tests.regions.test_mexico.MexicoTestSuite	5	0	0	11	16
tests.regions.test_karlskrona.KarlskronaTestSuite	0	3	0	13	16
tests.regions.test_volos.VolosTestSuite	0	0	0	16	16
tests.regions.test_lannion.LannionTestSuite	0	0	0	16	16
tests.regions.test_trento.TrentoTestSuite	0	1	0	15	16
tests.regions.test_spain.SpainTestSuite	0	6	0	6	12

Region	Fail	Error	Skip	Success	Fail
tests.regions.test_stockholm.StockholmTestSuite	0	0	0	16	16
tests.regions.test_prague.PragueTestSuite	0	0	0	11	11

Table 1: Status of the sanity check of the Regions on Friday, 13th March, 2015

Finally, in order to improve in a positive way the performance of the IOs to guarantee the stability and the functionalities in their node, it was decided in coordination with FI-Core to measure some basic criteria to establish a Karma Point for each node. Minimal criteria under discussion to include in the mathematical formula are:

- the sanity check,
- the presence in the infographics,
- the resources usage.

2.1.1 FIWARE Lab and XIFI Federation Integration.

In order to get the regional local instantiations of OpenStack federated into a joint FIWARE Lab portal, a number of subsystems need to be installed to aid the process. Here subsystems include Monitoring, Security, Deployment and Operations, as well as User-orientated and GUI Subsystems. By using the subsystems on the node it allows the node to operation, maintain and administrate inside the federation.

The deployment of the local XIFI platform starts with installation of ITBox which is used to deploy the cloud and federation tools. After the installation ITBox still provides support to the operation of OpenStack cluster and can used to enlarge service offering. With these components in place and cloud software configured to join the Federation services, the new federated nodes will be presented to the end federated users via XIFI cloud portal which in effect replaces the local instance of OpenStack Horizon. Here the FIWARE Lab user is granted authorization for access and control privileges to operate their instance on XIFI cloud resources and its networks.

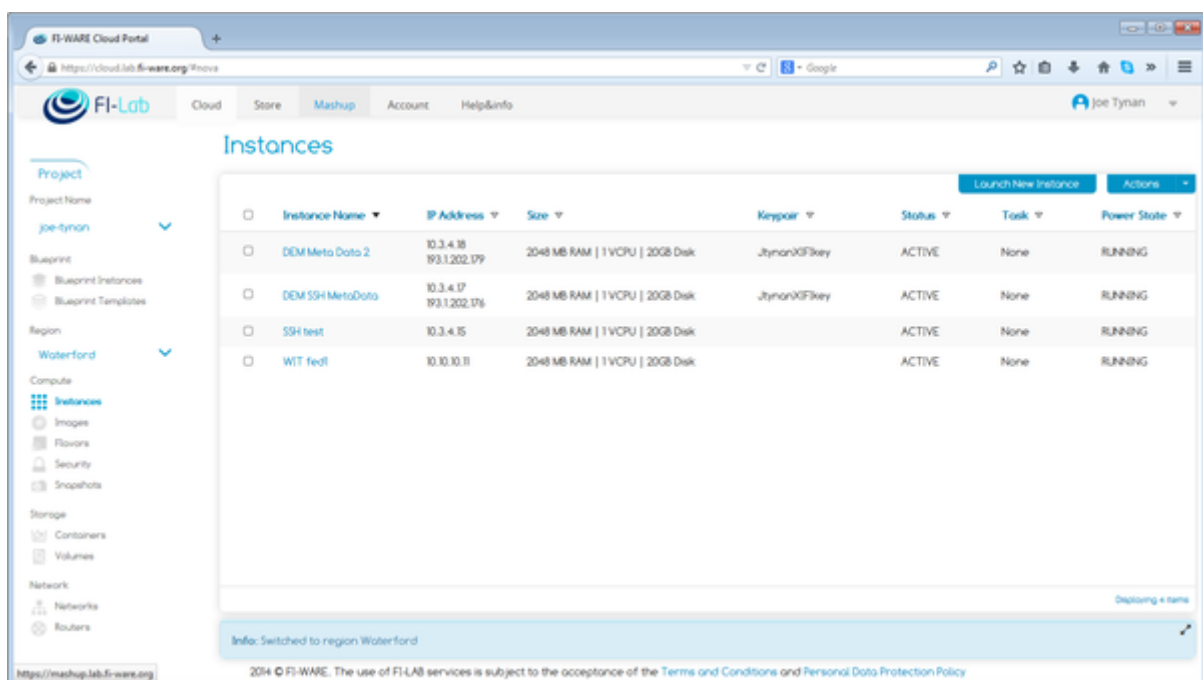


Figure 1: FIWARE Lab User Cloud portal

2.1.2 Federation Monitoring of Resources and Operations

Currently the federation has deployed an infographics and status page that gives an overview of current “capacity” of each geographical region and the available resources hosted there. At URL <http://infographic.lab.fi-ware.org/> the current resource status of the XIFI federation is displayed, as shown in Figure 2 and Figure 3.

On 30th March 2015, the XIFI federation has 8407 federated users, 1744 organizations, 16 federated regions, 2720 CPU cores, 7944 GB RAM and 176 TB of hard disk space with 1485 operational Virtual Machines associated to the federation.

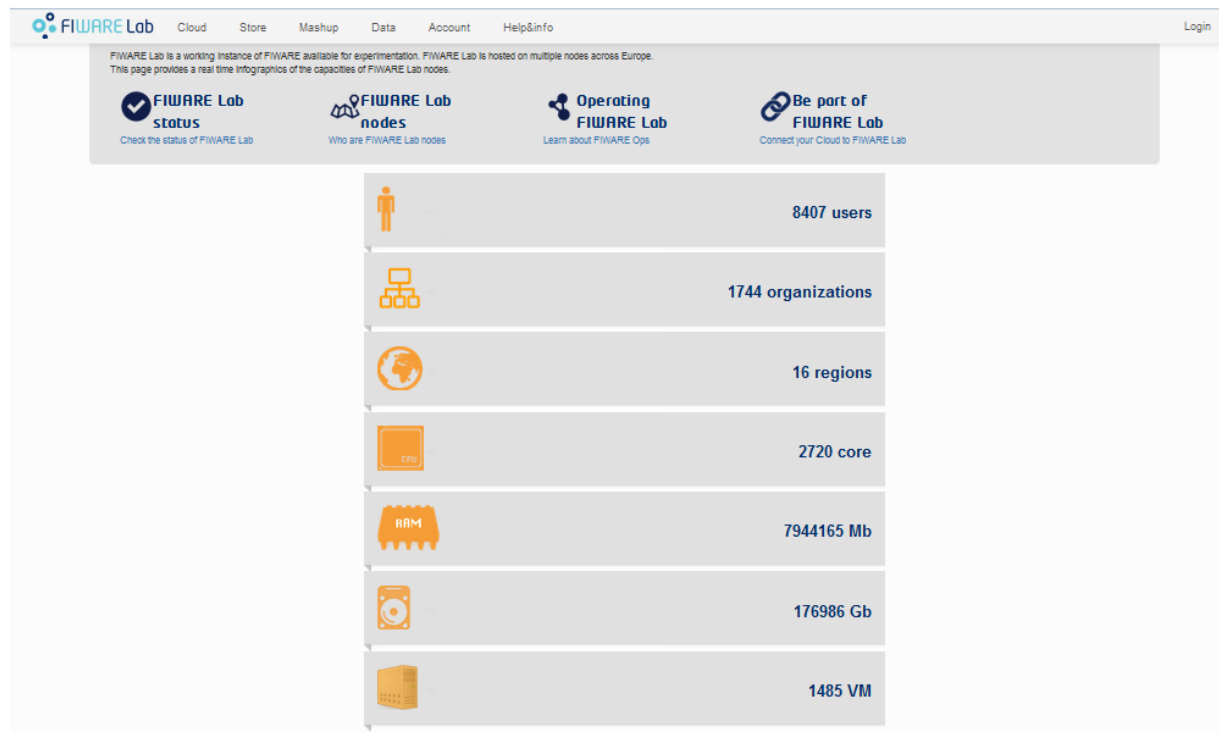


Figure 2: Operational capacity details on 30th March 2015

Node	Overall	Nova	Neutron	Cinder	Glance	Keystone P.	Support
Prague							
SophiaAntipolis							
Karlskrona							
Budapest							
Waterford							
Stockholm							
Lannion							
Berlin							
Gent							
Trento							
Crete							
PiraeusU							
PiraeusN							
Volos							

Figure 3: Example of regional cloud services status

2.2 Waterford Node

2.2.1 Description

The Ireland Node is built using ITBox for the OpenStack deployment.

The node's footprint comes in at 96 cores of compute across 12 servers, 386 GB of compute RAM, with 1.6 TB of live migration space running over NFS.

Our XIFI node networking links comprises of 1 Gbps local link to a 10 Gbps shared up link to our NREN HEAnet core for all XIFI's public IP traffic. As our site relies on HEAnet for both connectivity and Provider Aggregated IP space, the XIFI current allotment is a /25 subnet (193.1.202.128/25). This allotment pool is currently segregated into floating address pool (70 IP's) and the remainder allocated to federation server instances.

The XIFI Federated network link has a 1 Gbps dedicated connection that runs from a PE router hosted in WIT data centre, from the router a MPLS BGP session to HEAnet, then a MDVPN and BGP peering to Géant, which allow us access to other private IP address ranges on all XIFI nodes in the federation. The IP allotment here is a /20 subnet on the 10.0.0.0 network. Again here we have allotted the first Class C in the range for a second OpenStack network with 150 floating IP address. The remainder of the address range is reserved for future provisioning.

2.2.2 Experience on Support and Maintenance

Our Initial deployment of OpenStack was a manual installation, which was based on directly sourcing packages from the Operating System repositories.

But it became clear that this was not sustainable in the long term. We found ourselves in the situation where it was difficult to add on new features as we progressed, due mainly to the proprietary nature of the cloud deployment.

With the increased usage of the platform it became apparent that the allocated resources such as compute nodes, networking IO and disk allocation was incorrectly apportioned for our requirements. The need for a more robust solution was needed when after applying a distribution upgrade to the operating system the upgraded kernel subsequently had compatibility issues with OpenVSwitch.

It was for these reasons we decided to totally re-install our cloud foot print and adhere to the projects own cloud deployment software.

Implementation of a Pre-Production Environment: We found ourselves in the precarious situation where we were making changes to the production environment. This made the node implementation very exposed at maintenance time where we had no location to test roll out procedures, staging and/or roll back plan.

This also allows us to plan configuration changes such as establishing the impact a configuration change on the underlying nova network a change could be made to the network configuration on the pre-production environment without effecting production users.

This allows us to carry our component testing before we deploy. We're going to be exploiting this environment as we prepare for the upcoming OpenStack migration.

2.2.3 Current Status

Currently the WIT XIFI routing table looks like:

XIFIRouter1#sh ip route vrfxifi

```
C      10.0.0.0/20 is directly connected, GigabitEthernet0/2
B      10.0.16.0/20 [20/0] via 188.1.201.9, 1w4d
B      10.0.32.0/20 [20/0] via 193.51.178.40, 1w4d
B      10.0.48.0/20 [20/0] via 193.51.178.40, 1w4d
B      10.0.64.0/20 [20/0] via 62.40.96.18, 1w5d
B      10.0.96.0/20 [20/0] via 62.40.96.22, 1w5d
B      10.0.144.0/20 [20/0] via 130.242.80.54, 6d18h
```

Figure 4: Routing table of WIT

There are a series of cloud images associated with XIFI federation that are required to be hosted locally on the Glance Repository, each hosted image is assigned NID [16] that needs to be configured so that federation can access them via Image ID rather than a direct name lookup. These are currently hosted and provisioned on WIT's glance repository. Federation monitoring tools were also deployed on a standalone server. Monitoring Tools comprise NAM, DEM, OpenStack Data Collector, NPM, NGSI, Context Broker and Security Probes. As with any software deployment, the WIT XIFI node has altered certain software applications in order to fit our requirements. Listed as follows:

- NTP: At installation time the NTP server could be assigned to OpenStack cluster, a manual reconfiguration was required in order to get the cluster on Irish Summer Time (IST).
- Zabbix: As we already have Zabbix deployed on the WIT XIFI node, it has proven itself to be a valuable monitoring resource and therefore we included it into the OpenStack cluster. Deploying the Zabbix probe on each node give us access to a wide array of hardware and software parameters to monitor and respond to accordingly. It also provides us with a meaningful state on how local resources are being utilized at any period of time. Throughout the course of the XIFI project it has been a powerful tool and has assisted in the maintenance of our node.
- File Backup: The WIT XIFI node implemented both online and local backup of important configuration and DB schemas. File backup is provided in two forms, a SVN service that is used for putting configuration files under version control and the second method handles raw file backups using NFS and SSH, similar to the solution provided by rsync.

- OpenStack flavours: We have tailored the ability of end users to deploy high specification virtual machines on the cluster as this could quickly lead to depleted node resources and over allocation on the WIT node.

ID	Name	Memory MB	Disk	Ephemeral	Swap	VCPUs	RXTX Factor	Is_Public	extra_specs
1	m1.tiny	512	0	0		1	1.0	True	{}
2	m1.small	2028	20	0		1	1.0	True	{}
3	m1.medium	4096	40	0		2	1.0	True	{}
4	WITGeneric	1024	5	0		1	1.0	True	{}

Table 2: Current WIT node nova flavor-list

- User Floating IP allotment: As every user is assigned 100 floating IP's on the user account at time of creation, we deemed it necessary to curtail this to 3 as a single user could potentially use all of WIT's external IP address space.
- Customizing Nagios checks: As part of the Federation, NRPE was rolled out across all nodes and specific checks were established as required by the XIFI OpenStack Data Collection component.

2.2.4 OpenStack Configuration

The OpenStack controller, Glance and Quantum networking reside on a single node deployed via ITBox with a full description show in D5.2.

Federation monitoring resides on a standalone server but will be migrated to a virtual instance in the near future. The Node has Nagios, DEM, NAM and NPM deployed on a physical server with the CB deployed on a virtual instance. The Keystone service do not reside on the node but remotely on the keystone proxy. The Horizon front end does not reside on the nodes but is hosted on the XIFI cloud portal. There are quotas applied to compute and networking resources on the Waterford node. Two important limits worth noting is the restriction "floating IP" and "router" instance. As public IPv4 address are a valuable resource.

Property	Value
metadata_items	128
injected_file_content_bytes	10240
ram	51200
floating_ips	2
key_pairs	100
instances	6
security_group_rules	20
injected_files	5
cores	6
fixed_ips	-1

Property	Value
injected_file_path_bytes	255
security_groups	10

Table 3: Waterford default quotas

(quantum) quota-show

Field	Value
floatingip	50
network	10
port	50
router	2
security_group	10
security_group_rule	100
subnet	10

Table 4: Waterford network resource quotas

2.2.5 User-base

The WIT node currently provides resources for Use Case project FINESCE. These resources are deployed for trial, preproduction and development purposes. There is also a growing number of non-UC resources being utilized on the node.

List of the 10 over 57 first tenants Usage from 2015-02-04 to 2015-03-05:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk GB-Hours
11233	70	299.24	0.15	2.92
4259	62	1429823.36	700.03	13950.66
11189	27	1364278.5	666.15	13324.34
2900	23	6307369.55	3533.67	58569.43
3131	16	2417930.54	2706	13440.00
81	14	1697635.23	828.92	16578.47
2317	12	98486.31	192	0.00
3005	12	1760616.32	859.68	34387.04
11466	9	9494323.76	260.06	4702.74
11356	8	1170933.03	571.74	11424.89

Table 5: WIT node utilization

2.3 Trento Node

2.3.1 Description

Trento node is composed by 7 Dell R210 and 6 Dell R715.

Hardware:

- 1x Rack 42U
- 2x Eaton ePDU fully managed
- 1x Alcatel 6224 L2 switch with 24x 100Mb/s port and 4x1Gb/s ports
- 2x HP 3800E L2/L3 switch with 24x 1Gb/s port and 2x10Gb/s sfp+ OpenFlow enabled
- 6x Dell R715 servers with 8x 1Gb/s ethernet NIC and iDRAC management interface
- 7x Dell R210 servers with 4x 1Gb/s ethernet NIC

Controllers, Monitoring, Object Storage are Dell R210 configured with :

- 4 cores
- 16 GB of Ram
- 2 TB of Disk (for 4 servers) and 4 TB (for the 3 Object Storage servers)

The Compute nodes are Dell R715 configured with:

- 16 cores
- 64 GB of Ram
- 1.6 TB of Disk

2.3.2 Experience on Support and Maintenance

Every night our scripts backup all the important configuration files and dump the entire mysql database used by Openstack services.

The service Corosync (crm) is the core of the HA architecture. In a High Availability (HA) environment is mandatory to use crm commands to manage the node and solve the problems. The HA architecture is based on the Galera replication system and to be sure to maintain the controllers in a consistent status it's important to follow best practise during maintenance operation :

- stop the corosync in the right order
- stop the corosync on the controller that has to be mantained
- check the status of the corosync
- check the status of the mysql replication between the nodes
- after the restart of the corosync on the maintained node check the "crm status" to be sure that the all services are working properly

Rabbit messaging system is the component responsible of the communication between the services of the entire Openstack environment. The status of the Rabbit-server must be checked (better with a monitoring tool) in order to grant the fully operativity of the node.

After the deployment of the Openstack environment, FUEL remain a key component of the environment. It collects the log of the servers and shows the logs in a web page. It works as DNS and as NTP server. The HA environment need the date always synchronized on all servers. A good way to grant the NTP synchronization is to add another NTP server.

All the main Openstack services and the mysql process have to be monitored by an automatic

monitoring tool (i.e. Nagios) in order to have a complete view of the node status and to be notified instantly if something goes out of service.

2.3.3 Current Status

Some bugs regarding HA and rabbit-server has been resolved and the Node works properly on Openstack Grizzly version. Trento Node team is ultimating the migration to the Openstack IceHouse version.

2.3.4 OpenStack Configuration

Trento node hosts the Grizzly Openstack version deployed via the usage of the Mirantis Fuel 3.2.1 suite. The Trento node consists of :

- 3 Controller nodes in HA mode
- 6 compute nodes
- 1 Monitoring node running on Ubuntu 12.04
- 1 Fuel 3.2.1
- 3 Object Storage
 - Deploying mode: Multi-node HA
 - Networking model: Neutron with GRE
 - Compute Hypervisor: KVM

Network distribution on the NICs:

- Compute nodes
 - G1 - untagged traffic for Fuel
 - G2, G3 - active-passive bonding, tagged traffic for OpenStack and XIFI vlans
 - G4 - untagged traffic for local management
 - iDRAC - untagged traffic for local management
- Other nodes
 - G1 - untagged traffic for local management
 - G2, G4 - active-passive bonding, tagged traffic for OpenStack and XIFI vlans
 - G3 - untagged traffic for Fuel

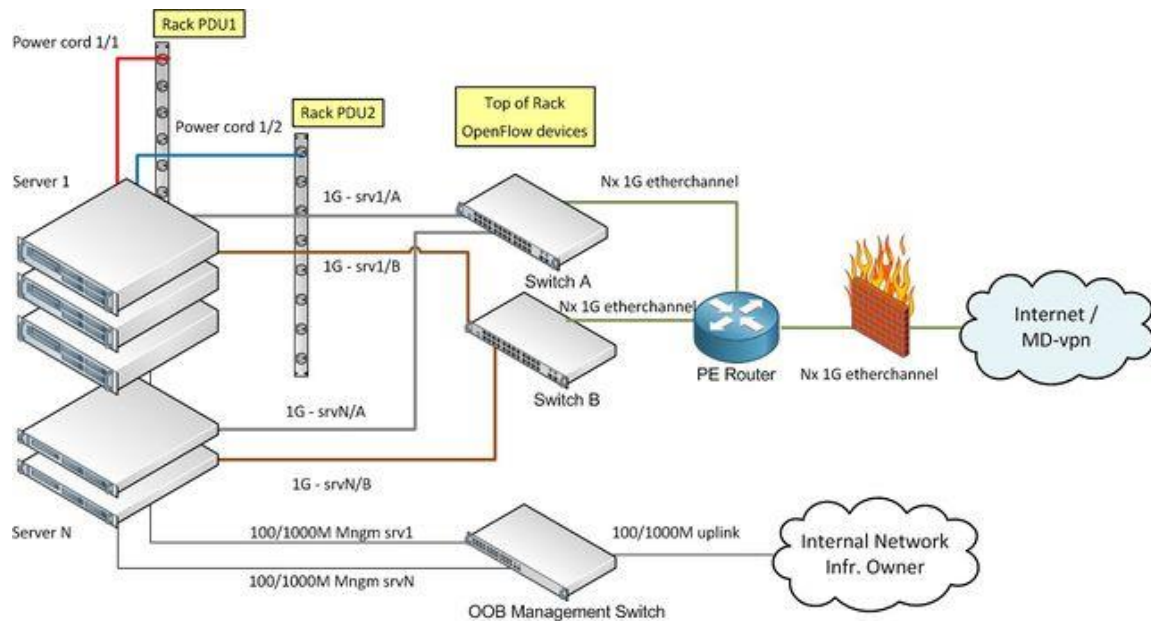


Figure 5: TN architecture

The external connectivity target architecture is divided in two different connectivities for different purposes:

- 1 Gbps GEANT/NREN connectivity for the backbone of the federation, that could be implemented also in IPv6.
- 100 Mbps Internet connectivity for end-users

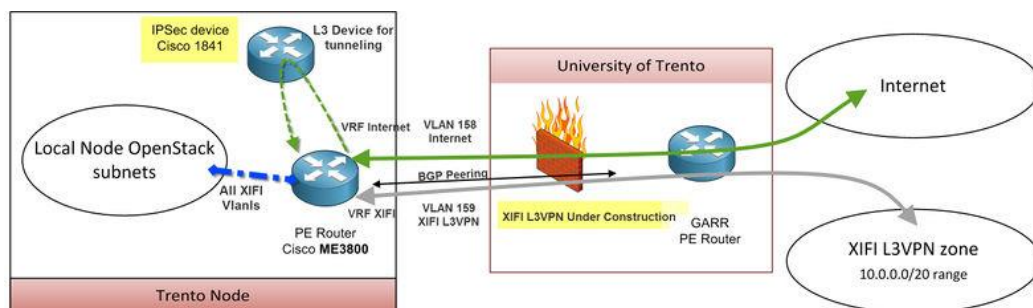


Figure 6: TN external connectivity target

The federation network has been configured in every physical machine. Currently the network of the federation is achieved through the VPN connecting directly to Lannion.

2.3.5 User-base

The Trento Node is providing resources for multiple projects coming from different area and is also providing the master node of the monitoring system.

Some partners of FIWARE are also using this Trento node:

- Engineering
- Atos
- Telecom Italia
- Create-Net
- UPM

- Thales
- Synelixix
- IT-innovation

The Trento Node is providing resources also on a Test node for the WP2 in order to validate the FIWARE GEs.

Use Case projects:

- UC4 - Marketplace services
- UC5 - Quality of Experience in NaaS
- UC7 - Monitoring Qos in the Node
- UC10 - Security monitoring
- UC11 - GE monitoring
- UC13 - Augmented Virtual Tourist

List of the 10 over 62 first tenants Usage from 2015-02-02 to 2015-03-03:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk Hours	GB-
11233	19	89.32	0.09	0.87	
2782	12	13185140.05	9744.11	167042.83	
4259	9	53.90	0.06	0.47	
3954	5	10577231.64	5164.66	165763.65	
81	5	729187.56	712.10	7120.97	
11266	4	423091.20	453.32	5718.37	
3373	4	2835227.02	2768.78	27687.76	
11661	3	20910.36	20.42	204.20	
442	3	3563520.00	3480.00	48720.00	
11595	2	931.27	1.77	43.33	

Table 6: TN node utilization

2.4 Berlin Node

2.4.1 Description

The Berlin node consists of data centre functions located at Fraunhofer FOKUS premises and of a wireless testbed located at DT premises in Berlin. The two sites are connected through fibre (1 Gbit/s) and utilize a dedicated VPN tunnel for L3 connectivity. Both sites connect to the MD-VPN on distinct addresses each in the 10.0.16.0/20 range, while the Fraunhofer site additionally provides public IPv4 addresses for external access.

The current OpenStack setup in the data centre at FOKUS was deployed through the ITBox v1.2.4.0. At the moment OpenStack is operational with the Grizzly release as distributed setup with no high availability. The OpenStack services running on physical Dell PowerEdge servers. Whereas the compute nodes are Dell PowerEdge M620 blades. The monitoring subsystem is running on a dedicated physical node. For shared storage a NetApp Metro-Cluster is used that provides 20TB, fully redundant storage through NFS.

2.4.2 Experience on Support and Maintenance

Most of the user support was related to instances that get stuck in the ERROR state. After analyzing the situation it turned out that most of the user of the FIWARE Lab doesn't know how to correctly launch an instance. Dependent on the network configuration of the FIWARE Lab Region it is required to create a virtual tenant router and tenant network prior to launching a virtual instance. New instances in ERROR state appear every day, therefore a cronjob is removing instances in ERROR state every night.

The current configuration of the networking service requires at least two public IPs for each tenant. At the moment the available floating IPs are consumed very fast. This happens because the IP pool is too small and the IPs are required for accessing the virtual instances remotely e.g. SSH. During the operation of the node it was analyzed that floating IPs have been allocated to tenants but not assigned to any instances. This requires manual deletion of floating IPs that are not assigned to instances, in order to make them available for other users.

The security policies of Fraunhofer define that all external incoming connections have to be denied as long as they are not explicitly allowed. This policy requires a lot of interaction between different stakeholders. The user is not able to access specific ports of its service and is contacting the helpdesk. The node operations team needs to contact the NOC of the institute which administers the central firewall, and request to open specific IP/port and protocol. The dynamic nature of the floating IP pool makes this task even more complex. To increase the user experience it was decided to open SSH for the whole public floating IP pool.

One of the most critical parts is that the infrastructure owners are not able to contact their users. Legal issues don't allow the Spanish node, which hosts the central IDM (Keystone), to share the contact information with the other infrastructure owners. Trial users pass by, do some test and leave without freeing up the consumed resources. The infrastructure owner cannot know if the allocated resources are still in use or not. To address this problem several approaches are discussed that will be implemented in the near future. One solution would be to put a metadata parameter which contains the lifetime of an instance. This will allow the node operator to delete all instances on its infrastructure where the lifetime is expired.

2.4.3 Current Status

The Berlin node currently passes all the sanity check which runs every night to validate the basic FIWARE Lab features for every node. Also the status monitoring (<http://status.lab.fi-ware.org/>) of the FIWARE Lab services indicates that the Berlin node is fully operational.

Currently the preparation of the Upgrade to a new OpenStack release is ongoing. This includes the test and evaluation of features and settings of the new OpenStack release as well as the migration planning. The new OpenStack release is installed on separate hardware to not disturb the operational setup. There we can evaluate available solutions to solve issues that we have experienced in the Grizzly release.

In order to provide more public floating IPs an additional IPv4 addresses are requested. It is also planned to provide IPv6 access to the instances with the new release. This will relieve the most limited resource we have at the moment.

2.4.4 OpenStack Configuration

The Berlin node operates in a distributed OpenStack setup with no high availability. Most of the OpenStack services are running on the controller node. Quantum network provides two L3 agents for the external networks, where the first L3 agent is attached to L3-MDVPN and the second L3 agent provides access to the public IP network. Furthermore a per tenant-router with private networks configuration allows isolation of user traffic among different tenants.

2.4.5 User-base

The Berlin node is hosting images from the FI-PPP use case projects (e.g. FI-Content; FI-STAR) Additionally we've been in contact with several SMEs that are using the platform. Recently the number of instances has been increased which is due to the Phase 3 and Accelerator activities (e.g. Speedup Europe)

List of the 10 over 57 first tenants Usage from 2015-01-28 to 2015-02-26:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk GB-Hours
4661	47	450820.00	880.51	0.00
11233	35	131.41	0.06	1.28
3005	24	229960.56	112.29	2245.71
81	12	1578819.75	770.91	15418.16
3015	10	14509695.63	8178.29	134406.37
9200	10	171657.53	84.24	1673.53
2988	5	15.36	0.01	0.15
3233	5	694083.74	1338.68	113.03
10769	5	4161682.87	2032.07	40641.43
3273	4	4311.04	2.10	42.10

Table 7: List of tenants on the Berlin node

2.5 Brittany Node (Lannion)

2.5.1 Description

The node of Lannion is composed of 8 Dell C6220 with a total 96 Cores, 256GB of RAM and 32 TB of Disks.

The node was deployed in HA using the ITBox v1.2.4. It is composed of 6 computes and 2 controllers (HA deployment). The Object Storage role is hosted by the 2 controllers.

1 Pica8 switch with openflow capabilities is used to connect the servers based on a 1 Gb links. ImaginLab relies on Renater for internet and MDVPN connectivity and as well for Public IP Allocation.

Currently, there are 2 external networks deployed on our configuration: A public External network with a /26 subnet of Public IPs (64 Public IPs) and a MDVPN external network with a /24.

An extension of the node is ongoing and the goal is to double the actual capacity. We are going to add to the current configuration 8 more Dell C6220 with 96 Cores, 512 GB of RAM and 32 TB of Disks. The final node when deployed will have 3 controllers and 13 computes, the object storage role will still be hosted on the controllers. We will add a new Public external network, it will be a /24 and this will extend the Public IP available to the node to 320.

2.5.2 Experience on Support and Maintenance

Our first OpenStack installation was manual and mainly to get hands on experience of OpenStack. At the time of this installation, no automatic installation tool was available using Ubuntu as OS. We

installed one server as controller and Network node, one server as cinder and one server as compute node. DCRM has been installed and we saw a lot of instability related to it and generating a CPU load of 100%.

A re-installation was made using the first release of the ITBox including DCRM. This was made in order to have an OpenStack node stable and ready for a review that attends end of 2013 beginning of 2014. The configuration was as following: 1 server with ITBox, 1 server as controller and network node and 1 server as compute.

A new re-installation was finally made in order to have HA as the previous version of the ITBox did not have this option. This was done in order to fit the requirement listed in the D1.4. The final configuration is described in the D5.2: two servers acting as controller, network and swift node in HA and six servers as compute nodes.

Lately, a need that were coming from the WP2, we deployed a pre-production environment in HA, composed of 2 controllers and 1 compute. In this environment, we can perform configuration changes "without risk".

The main problem we had with our node was due to the High availability. ITBox v1.4.1 was based on a Fuel version where Ubuntu and high availability were newly supported. Then some scripts had some bugs and only been corrected in further versions:

- Some processes taken care by "crm", were started at boot via "upstart" (like quantum-metadata-agent)
- Some processes taken care by "crm", were not stopped properly (like metadata-proxy were not stopped when quantum-agent-metadata was been stopped)
- Some processes were rebooting without reason (like quantum-agent-ovs)

Beyond what proposed XIFI to monitor the federation, we choose Nagios to monitor our node. The check of the node is done as following:

- By host: ping, fanX, tempX,etc.
- By services:
 - Check mysql request
 - Check Mysql DB Sync
 - Check_Openstack_NTP
 - Check Corosync_status
 - Check Load average
 - Check Nova services
 - Check status Rabbit conductor queue
 - Check status Rabbit q-plugin queue
 - Check Swift services
 - Check HA Services

We performed after installation of the node an image of each controller. This image will be used in case of a crash of the server.

As well, every night, our backup server backup all important files in all servers of the node (computes and controllers). It stores them for a couple of days in a loop strategy.

Current Status:

Since the correction of multiple bugs found in the scripts for HA, the Lannion node is quite stable.

The support to user is handled through the Jira Helpdesk. This activity includes all topics regarding administration of VMs, management of Quotas and Attribution of Floating IPs, etc.

2.5.3 OpenStack Configuration

Currently, the Lannion node is a High Availability node running on grizzly over Ubuntu 12.04 LTS and KVM as hypervisor. The services (glance, nova, etc) are deployed in respect of what is described in the D5.2.

Node federation:

We created a tenant named "Imaginlab". We configured this tenant to be connected to the federation external network and deployed the VMs used for the federation supervision to it.

In this tenant, there is 4 VMs used for the federation supervision: One VM for CB and NGSI, one VM for DEM, NAM and NPM. The other two VMs are used to host the BigData.

This tenant is also hosted the Access Control and the Security Probe GE.

A list of modifications on OpenStack configuration file has been made in order to use the security proxy of Santander instead of the keystone provided by default by OpenStack. By changing the local keystone to the keystone proxy, the OpenStack dashboard (horizon) is no more usable. The cloud portal is taking over the functionalities.

Quota:

Due to our national law enforcement, we decided to allow Public IP only on demand. We also decided to restrain the global size of disk to 50 GB by default for a tenant. This default configuration can be modified for a specific tenant in case needed. The default quota has been configured as follows:

Property	Value
metadata_items	128
injected_file_content_bytes	10240
ram	6000
floating_ips	1
key_pairs	100
instances	3
security_group_rules	30
injected_files	5
cores	6
fixed_ips	10
injected_file_path_bytes	255
security_groups	20

Table 8: Lannion quota-defaults

2.5.4 User-base

The Lannion Node is providing resource for multiple projects coming from different area:

- FI-Content, like:
 - FIC2Lab: The Lannion node as member of the FIC2Lab task force is hosting the FIC2Lab.

- It assembles the main technical results of FIcontent. The FIC2Lab Playground, geared towards the needs of developers, allows quick and easy testing and tweaking of the software modules either independently, combined with other FIcontent enablers or with FIWARE Generic Enablers.
 - Smart City Guide
 - Connected TV
- Projects related to Accelerator Projects (FI-C3):
 - 4 planet
 - Doxanet
 - ndmac-systems
- Use Case project:
 - Use Case 12: Fire to the FI-PPP
 - Use Case 10: Security Monitoring

List of the 10 over 38 first tenants Usage from 2015-02-02 to 2015-03-03:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk Hours	GB-
3273	126	1203703.94	587.75	11754.92	
11233	45	134.26	0.07	1.31	
2988	42	770992.59	374.61	7378.10	
8576	32	25896867.53	12146.88	232891.69	
2983	8	12730374.47	6720.00	120960.06	
3437	5	11010053.60	5376.00	107520.05	
3005	3	1290334.39	630.05	12600.92	
3997	3	6193155.15	4032.00	0.00	
4511	3	3096577.57	2016.00	26880.01	
11595	3	33232.17	16.23	324.53	

Table 9: List of tenants on the Lannion node

2.6 Spain node (Seville/Malaga)

2.6.1 Description

The Spain node is a distributed infrastructure where the resources (computational and storage) are located in many Spanish Cities. The network connectivity of the resources are provided by RedIRIS through RedIRIS-NOVA (RedIRIS dark fiber network). The figure below shows a schema of the resources.

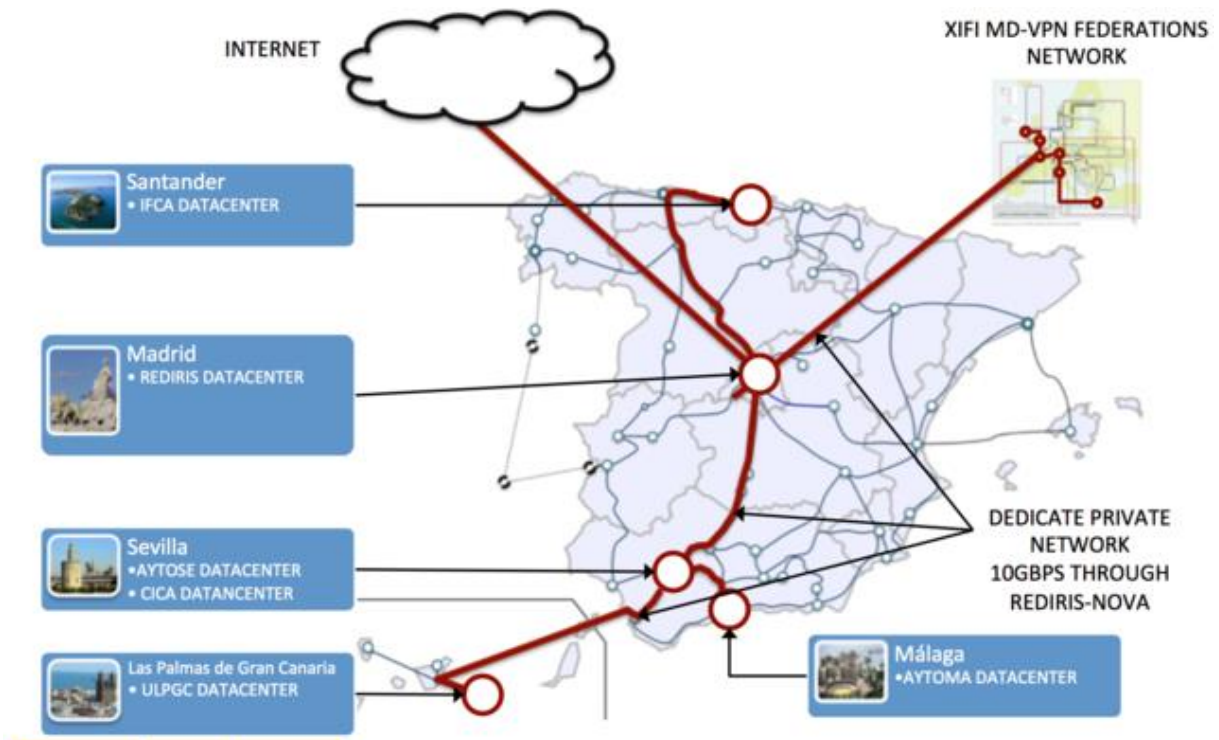


Figure 7: SpainNode – network connectivity

Figure 8 shows a summary of the resources deployed in the Spain node. As you can see, there are a set of datacenters which have a well defined role and each one has a set of computational and storage resources. The table summarizes the total capacity per datacenter and includes the quantity of Memory, Cores, GPUs Cores, TeraBytes of Storage (shared and distributed).

Node/ DATECENT ER	CORES			STORAGE (TB)			RAM MEMORY (TB)	SECURITY		ROLE
	Computati onals resources (cores)	Computational resources (cores) with high speed storage	GPUs Cores	Distributed	High performance storage	Shared	Distributed on the nodes	Firewall	Type of Security	
UNICAN/ IFCA	224			4,2		10	0,448	No	Perimet ral	TESTBED
REDIRIS/ CICA	104			38			1,664	Yes	Perimet ral	Fiware lab and GEs
AYTOSE/ AYTOSE ^{*(1)}	288	128	896	28,4	48	100	3,328	Yes	Local	Fiware lab
AYTOMA/ AYTOMA	288	128	896	28,4	48	100	3,328	Yes	Local	Fiware lab and GEs
ULPGC/ ULPGC ^{*(2)}	768			96		100	6,144	No	Perimet ral	Fiware lab
REDIRIS/ TELMAD	Dedicated to Network Routing to Internet and XIFI MD-VPN Federation. VPN concentrator to provide the MD-VPN access to those nodes that are not able to get a MD-VPN uplink (Ej. Mexican node, TID, i2Cat, ...).									

Figure 8: SpainNode - resource distribution

2.6.2 Experience on Support and Maintenance

The maintenance of the Spanish node has become harder and harder. This is not only due to the problems related to the old version of OpenStack, but the node is running with some other problems that came up.

The node runs out of resources

The Spanish node run out of computational resources, after 18 months of life, the Spanish node got out of resources. At this moment, some more virtual hosts could be deployed in the Spanish node. However, the instances that have been deployed kept on running properly. We managed to add 4 extra hosts in order to be able to support the Campus Party held in Brazil on February 2015. However, these 4 extra hosts will disappear in a few days as well as the instances deployed on them. We are waiting to the movement of the location of the datacenter in Sevilla to resolve this issue.

Moving from one datacenter to another

The Spanish node got to move the resources from one datacenter (in Seville) to another datacenter (In Málaga) in order to keep the Spanish node running. This movement of information is needed because the hosts, which run the Spanish node and some other important services for the whole federation, will change its location and they will be unavailable for several days. This process will conclude in February.

Migration to OpenStack - Juno

It is working on an OpenStack Juno installation, which will replace the current OpenStack installation in the Spanish node. The replacement will happen gradually. Only the Instances of the Community users will be migrated to the new OpenStack version, taking into account the new account policies that will be implemented in the Federation. This migration process will have to meet at least these requisites:

- Respect the information in the instances. Although it will be impossible to keep the IPs since the IP plan must change.
- Respect the information stored in the nova-volume, keeping the external disks attached to the instance. This means that nova-volume information must be migrated to new Cinder block storage.
- Respect the current accounts, which means that users will be able to handle their new instances without changing anything in their accounts.

Other issues

We are constantly facing problems due to the lack of care in security terms that some FIWARE Lab users show us. We keep on having constant problems with easily breakable configurations and weak passwords, which makes some instances to be compromised, and thus putting the whole FIWARE Lab node under security risk. We have to mention that we put an extra effort, together with the FI-Core project, into the images synchronization between all the nodes in the federation in order to keep the whole FIWARE Lab consistent. The security has also been a concern and we've been working on setting up the server to use secure protocols and we'll keep on working on this.

2.6.3 Current Status

The Spanish node is working with an old version of OpenStack but efforts are being done in order to have a brand-new installation as described before.

We are handling more than 2000 instances, but it is near to reach its limits. However we are working to increase that capacity. New datacenter will be added to the Spanish node, which will make it able to more than double its capacity. The Las Palmas datacenter will give us in the short time the capacity of tripling of resources to offer to the FIWARE Lab users.

2.6.4 OpenStack Configuration

Currently, the Spain node is running on Essex version over Ubuntu 12.04 LTS and KVM hypervisor. You can see it by executing the following commands.

We can obtain information about which version of OpenStack we have installed by executing the following commands in the controller:

```
$ dpkg -l | grep nova-common
```

```
ii nova-common 2012.1.3+stable-20130423-e52e6912-0ubuntu1.4 OpenStack Compute -
common files
```

And following to the OpenStack published releases we can see that it corresponds to the Essex 2012.1.3. Regarding the operating system we follow the indications and we are using the Ubuntu 12.04 LTS, we can check it by executing:

```
$ lsb_release -a
```

```
No LSB modules are available.
```

```
Distributor ID: Ubuntu
```

```
Description: Ubuntu 12.04.5 LTS
```

```
Release: 12.04
```

```
Codename: precise
```

Besides, the hypervisor that we are using is KVM version 1.0 how we can see in the execution of the following commands:

```
$ kvm --version
```

```
QEMU emulator version 1.0 (qemu-kvm-1.0), Copyright (c) 2003-2008 Fabrice Bellard
```

Last but not least, the libvirt that we are using and associate to the OS is the 0.9.8 how we can see in the execution of the following command:

```
$ virsh --version=long
```

```
Virsh command line tool of libvirt 0.9.8
```

See web site at <http://libvirt.org/>

Compiled with support for:

- Hypervisors: Xen QEmu/KVM UML OpenVZ LXC Test
- Networking: Remote Daemon Network Bridging Nwfilter VirtualPort
- Storage: Dir Disk Filesystem SCSI Multipath iSCSI LVM
- Miscellaneous: AppArmor Secrets Debug Readline

Monitoring: The node is monitored by Nagios 3.4.1, NGSI Adapter 1.1.1, NGSI Event Broker 1.3.1, NAM and OpenstackDataCollector. Besides standard monitoring configuration, Nagios is checking a set of services in OpenStack in order to know if they are working properly.

Federation Core Public Services: currently, we have configured a hosts that offer services to the rest of the federation. This host maintain the following services: Wiki of FIWARE, Jira, Web site of FIWARE, GE Catalog, Cloud Portal and Help&Info site together with the Store and Mashups portal.

Additionally, a specific virtual machine is reserved in this host in order to offer a router/firewall to the OpenStack services.

Federation Core Private Services: we keep two specific hosts in order to offer private services to the federation. The services that we have on each host are the following:

- Pegasus - PaaS Manager, Sagitta - SDC Manager, IdM, Keystone Proxy (HA), Orion Context Broker Global and CKAN Portal.
- Second instance of Keystone Proxy (HA), Nagios and Federation Monitoring.

Network: Actually, the Spain node is using the nova-network and not neutron, which means that all the virtual machines are deployed in a flat network. The installation of a new version of OpenStack will change this issue and introduce the definition of different network in order to access and work with.

Compute: Currently, the Spain node is configured with 14 physical hosts together with one host to keep swift, another in which we have the nova controller and Access Control GEi. Regarding the resources available, we had to reduce the quota limit to be used to the following values.

nova-manage --os-region Spain project quota list

Quota	Limit
metadata_items	128
volumes	2
gigabytes	50
ram	25000
security_group_rules	30
instances	3
fixed_ips	10
security_groups	20
injected_file_content_bytes	10240
floating_ips	1
injected_files	5
cores	6

Figure 9: SpainNode – project quota list

2.6.5 User-base

The Spain Node is providing resource for multiple projects coming from different area:

- FIspace – 1 instance (20 Gb Hd, 2 cores, 2 Gb RAM)
- FIContent – 3 Instances (60 Gb Hd, 5 cores, 6 Gb RAM)
- FINESCE – 8 Instances (160 Gb HD, 6 Cores, 14 Gb RAM)

Some partners of FIWARE are also using this Spanish node

- Engineering – 11 instances (220 Gb HD, 16 Gb RAM, 5 Cores)
- Telefónica – 56 instances (1120 Gb HD, 84 Gb RAM, 28 Cores)
- Atos – 11 instances (220 Gb HD, 16 Gb RAM, 5 Cores)
- Thales Group – 7 Instances (140Gb Hd, 10 Gb RAM, 6 Cores)
- Orange – 9 Instances (180 Gb Hd, 5 Cores, 14Gb RAM)

Use Case projects:

- UC5: Quality of Experience in NaaS
- UC6: SDN traffic engineering
- Both of them are integrated into the new MG2: MG2: How do I get the best out of what's available?

Our infrastructure has also been chosen as a developing place for many Startups or companies as well as university students and many curious people. The increase of capacity will give the opportunity to the different accelerators program to work with the available resources that we provide in Spain node.

Finally, the typical the resume of node utilisation from 2015-01-17 to 2015-02-15 is the following:

Resume of usage in the Spain node				
Σ Tenant ID	Σ Instances	Σ RAM MB-Hours	Σ CPU-Hours	Σ Disk GB-Hours
1699	2679	2338018512	1457336,66	31309265,61

Table 10: Resume of usage in the Spain node

List of the 10 over 1859 first tenants Usage from 2015-02-04 to 2015-03-05:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk GB-Hours
11233	49	492,09	0,24	7,21
11366	40	1356396,18	789,97	19896,46
81	34	7063259,71	3864,03	99313,94
4259	31	807412,85	394,28	11826,94
11364	30	1616952,72	808,32	23497,87
11278	26	972433,15	475,77	14235,17
6250	25	768131,38	718,88	7820,5
140	23	6641893,29	3243,11	97293,36
11331	23	1847954,09	921,26	26880,29
11325	19	276547,27	190,28	3498,55

Table 11: List of tenants on the Spain node

For more details about the resources consumption please refer to the administrators of the Spain Node

2.7 Prague (CESNET) Node

2.7.1 Description

Prague XIFI node performed and controlled by CESNET has 8 servers with 24 cores per each server with the power of 128GB of RAM per server. The available data storage is 4x 250GB SAS 2.5,, 10k

RPM HDD (WD2500BHTZ-04JCPV0) per server, plus 1x SSD Intel DC3500 480GB (SSDSC2BB480G4) per each server. We in CESNET do intend to enable any XIFI user to get the state-of-the-art accelerator's possibilities within FI PPP project:

- The node stability is considered to be essential
- Unique Mac OS X platform for the special Mac software development is ready and installations are being prepared
- CESNET's Experimental Network Facility (testbed) which consists of 5000km of leased fibres is ready to be used by any partner who fulfils security and administrative requirements (testbed on demand). It is a unique Integrated Facility where researchers are enabled to examine the real network environment as the testbed is being shared with the CESNET2 production network.

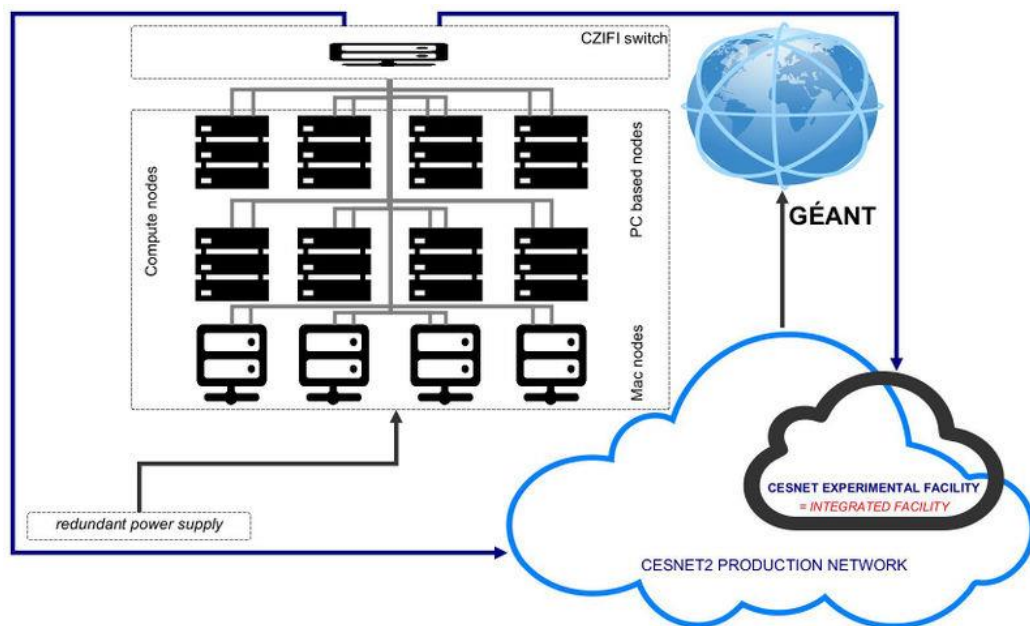


Figure 10: Graphical Scheme of the Prague node

Figure 10 shows the original hardware configuration of Prague node and the upcoming enlargement with the unique Mac OS X farm built in 1Q2015 which enables Mac software development for those who are interested.

CESNET Integrated Facility

March 2015

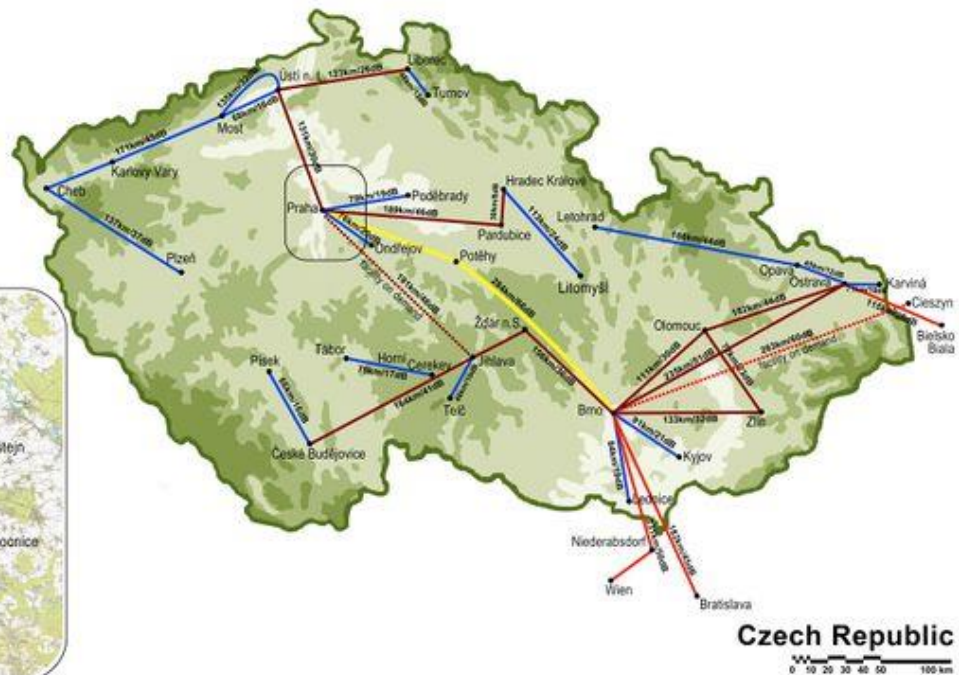


Figure 11: Integrated Facility Central Bohemia

Figure 11 shows the Integrated Facility: unique CESNET's testbed being shared with production network (5000 km) and which is ready to be used for other research parties.

2.7.2 Experience on Support and Maintenance

We wanted to avoid ITbox installation when installing Openstack in Prague Node because of security reasons. We decided for Grizzly version of Openstack because of XIFI recommendations. When installing, we took advantages of systems' mass management Puppet solution which enables to repeat exact configurations automatically anytime.

Today, deployment of Juno version is being done at the moment, part of the servers are running on that at testing mode. The new software deployment was made difficult by the fact that VMs' owners running in our node were not identified clearly. As we are running low with public IPv4 addresses, we provide users with them only on specific demand. On the other hand Prague Node is fully compatible with IPv6 addresses which we provide automatically.

2.7.3 Current Status

Prague node has been fully operational and stable since it became first operational node within the group of new members of XIFI federation in 2014. Mac OS X as a platform for other software development is being constructed as an enlargement of the node now in 1Q2015.

2.7.4 OpenStack Configuration

The node is monitored by Collectd, CESNET Nagios by own scripts and federated monitoring. The configuration is fully under Puppet management with complete configuration in Git repository. In case re-installation or extension of infrastructure is needed, PXE related setup together with Puppet is very

useful. The very advantage of our solution is the fact that the node computing extension (e.g. 100times) would take a few minutes. The computing nodes were installed using a minimal Kickstart recipe. The actual bootstrapping was done via the iPXE. When optimizing Openstack, Jan Kunderát from CESNET XIFI team contributed with patches, bugreports and actively participated in code review process:

http://stackalytics.com/?user_id=jkt%40kde.org&project_type=all&release=all&metric=all&company

2.7.5 User-base

List of the 10 over 79 first tenants Usage from 2015-02-01 to 2015-03-01:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk GB-Hours
11233	24	1684.42	0.82	16.45
4412	13	71304159.88	104449.45	87041.21
services	6	3903914.92	1257.46	17809.17
4408	5	50795620,56	8674,55	79410,98
7573	3	5200773,98	2539,44	50788,81
7247	3	4060528,74	1982,68	39653,6
11356	3	103725,65	71	877,28
11206	3	20042,37	39,15	0
7630	2	5505024	2688	53760
7735	2	4060987,32	1982,9	39658,08

Table 12: CESNET Usage in August 2014

2.8 Gent (IMINDS) Node

2.8.1 Description

XIFI node in Gent is deployed using the technical requirements and protocols described in D5.1. Our initial deployment was manually and after correct the configuration we used Chef scripts to integrate the node deployment within our own infrastructure and avoid conflicts and overhead on administration tasks. A deployment journal with the most important items can be found on D5.4.

The node is composed by 8 physical servers providing a total of 128 CPU cores, 384Gb Ram and 8Tb Storage. The node is connected to the MD-VPN, currently used for monitoring purposes and users, and 2 public IPv4 pools (one for Federation control services, other to provide floating public IP's to users).

The OS used for the deployment is Ubuntu 12.04LTS and OpenStack Grizzly from Ubuntu Cloud Archive repository. The kernel was upgraded due to an incompatibility bug with OpenVswitch kernel modules and the default kernel of the OS version, that was causing a kernel panic each time the OVS module was loaded.

2.8.2 Experience on Support and Maintenance

Management of floating public IPv4 pool, is done manually. Since there is a lack of control from the OpenStack software and it is a time consuming task, necessary since public IPv4 addresses are a very limited resource. Also, is noticed that users allocate public IP on a project a never release them, preventing other user to get a public IP address to run tests. To minimize this problem a local policy is applied. Allocate public address are disassociated after a week if not used.

Active monitoring for different components is needed to keep running the node and minimize downtime. For example, messaging software Rabbitmq used by OpenStack components to communicate, crashes often and is need to watchdog to take action on crash and restart the software. Also OpendataCollector, in charge of getting use statistics crashes if loses the communication, and needs take action to restart the service. All this tasks are monitored by scripts to take active action in case of a raised issue.

Reported abuse/spam users appeared on the Federation, and a cleaning action of VM's is need to have a proper usage of the node resources.

Basic tasks of maintenance of the software components, like upgrade new versions, correct configurations and create routines for check security and cleaning monitoring logs to have a proper run of the server nodes.

In order to allocate special Use Cases, that need more resources than the permitted by the defaults quotas, need a manual intervention to check if there is enough room to allocate it and modify specially the quotas for the user/project that is requesting more resources.

All the user and infrastructure support is managed by a project folder set up at the general Jira instance. When a support ticket is open, an email is received at the support-xifi@intec.ugent.be list and is handled as a best-effort from our team.

Two kind of tickets have been opened: Request for a coordinate upgrade on an infrastructure component based on the Federation needs and proposal to all nodes and support check from a helpdesk admin for endusers.

2.8.3 Current Status

The node is running since the first day of the Federation with minor changes and upgrades, as part of the maintenance tasks.

Federation runs every night a batch of tests to validate the basic FIWARE Lab features for every node. Also the status monitoring (<http://status.lab.fi-ware.org/>) of the FIWARE Lab services indicates that the iMinds/Gent node is fully operational.

Everyday the node is supervised and services status are continuously checked by active monitoring scripts. To solve the problem of exhausted public IP's, allocated floating IP's not used within a week, are de-allocated to free them to other users.

Also, we have dedicated 3 physical servers with Openstack Juno version, for testing purposes with the objective to upgrade the whole node to that running version when all the Federation tests finish and all compatibility issues were solved.

The main problem is that all users ask for a Public Floating IP and there is not enough to everybody. The problem is expected to be solved with the next Openstack version Juno that have IPv6 support.

Currently, our main focus is explore the options to run node with IPv6 support, as the lack of IPv4 addresses, after discussion with the rest of the nodes on which should be the approach and strategy to get the support also after checking the compatibility with the Cloud Portal.

As a first step taking to the objectives is upgrade to next OpenStack version Juno, that is the first with real IPv6 support.

2.8.4 OpenStack Configuration

Openstack Grizzly version is deployed with OS Ubuntu 12.04 LTS from Ubuntu Cloud Archive repository.

iMinds Gent node is composed by:

- 8 physical servers.
- 128 CPU cores.
- 384Gb Ram.
- 8Tb Storage.

Infrastructure network is connected to the MD-VPN, currently used for monitoring purposes and user VMs, and 2 public IPv4 pools (one for Federation control services, other to provide floating public IP's to users).

iMinds/Gent node is deployed over a distributed architecture with no high availability. Since we use our own Chef scripts to deploy the Openstack services, we are able to add new physical compute nodes on-demand as needed.

Since the resources are limited is needed to configure default flavors for VM's and quotas for users to prevent undesired uses or abuse and permit a proper equal service for all the users. But exceptions for special use cases can be allocate under request and always as a best-effort.

Overview of the default flavor configured at the node:

nova --os-region Gent flavor-list

ID	Name	Memory_MB	Disk	Ephemeral	Swap	VCPUs	RXTX_Factor	Is_Public	extra_specs
1	m1.tiny	512	0	0		1	1.0	True	{}
2	m1.small	2048	20	0		1	1.0	True	{}

Table 13: Gent flavour list

Overview of the default quotas as optimal use of the limited resources offered by the node:

nova --os-region Gent quota-defaults

Property	Value
metadata_items	128
injected_file_content_bytes	10240
ram	3072
floating_ips	2
key_pairs	100
instances	3
security_group_rules	20
injected_files	5
cores	6
fixed_ips	-1

Property	Value
injected_file_path_bytes	255
security_groups	10

Table 14: Gent quota defaults

2.8.5 User-base

iMinds is actively engaging with the local ecosystem via various FIWARE accelerators with which it is closely involved (CREATIFI, FI-C3, FINISH, FABulous).

List of the 10 over 21 first tenants Usage from 2015-02-01 to 2015-03-01:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk GB-Hours
3273	4	1022112.88	1996.31	0.00
4259	3	10.81	0.01	0.11
43	1	344064.01	672.00	0.00
3483	1	579349.08	282.89	5657.71
6470	1	344064.01	672.00	0.00
8642	1	1376256.05	672.00	13440.00
8708	1	1376256.05	672.00	13440.00
8916	1	344064.01	672.00	0.00
10538	1	98.99	0.05	0.97
10454	1	344064.01	672.00	0.00

Table 15: iMinds node utilization

2.9 Zurich (ZHAW) Node

2.9.1 Description

The ZHAW Node was deployed using Fuel 5.1.1 and manually configured - the configuration was performed mostly according to the instructions provided in D5.2 although there were some minor modifications to these procedures for the Openstack Icehouse release which were circulated within the project by Neuropublic. The cluster currently consists of 1 deployment node, 1 controller node, 1 monitoring node, 5 compute nodes and 1 storage node. A table showing the capacity of the system is below.

Resource	Capacity
Cores	208
RAM	1.63 TB
Disk storage	44.2 TB

Table 16: ZHAW system capacity

The Zurich node is currently connected to the other nodes in the federation via a high-speed 1Gb/s connection. This is connected to the other nodes via an EoMPLS connection to the GEANT POP in Geneva. The GEANT POP acts as the gateway for the Zurich node and advertises its availability within the internal federation network. This affords access to the MD-VPN via the 10.x address space.

id	name	cidr	allocation_pools
0e7ef2da-60dc-4d5c-85fe-0b55e6386b66	federation-ext-sub-1	10.0.176.0/24	{"start": "10.0.176.2", "end": "10.0.176.254"}
4da388ca-0c17-4f56-81ab-2755c0cab316	federation-int-sub-1	192.168.101.0/24	{"start": "192.168.101.2", "end": "192.168.101.254"}
53a7c488-a59d-4b59-8eb3-758b0aee98b0	public-ext-sub-1	160.85.2.0/24	{"start": "160.85.2.11", "end": "160.85.2.95"}
63ff4560-1c37-4b99-89c6-e778c8ab2e60	node-int-sub-1	192.168.100.0/24	{"start": "192.168.100.2", "end": "192.168.100.254"}

Table 17: `root@node-1:~# neutron --os-region Zurich subnet-list`

The default quotas on the Zurich nodes are listed below.

Property	Value
metadata_items	128
injected_file_content_bytes	10240
ram	8192
floating_ips	1
key_pairs	10
instances	3
security_group_rules	20
injected_files	5
cores	6
fixed_ips	10
injected_file_path_bytes	255
security_groups	10

Table 18: Default quotas Zurich nodes

2.9.2 Experience on Support and Maintenance

The Zurich node has had some issues with service provision resulting in a somewhat later entry into a clean support and maintenance modus operandi. The Zurich node spent some time working on tests to ensure that the infrastructure is performing correctly and developed tests similar to those in the central ‘sanity check’ primarily for internal use. Once the sanity check tests were developed, we obtained a copy of these which are run locally each night with the results emailed to the support team. There can be inconsistencies between these and we are currently looking into how to minimize these and how the can be tuned.

The Zurich node did have one major outage in March 2015 due to the data centre overheating which in turn was caused by failure of the air conditioning systems. The servers comprising the node all switched off automatically, taking the node down. Luckily this was exactly during the time of the upgrade of the cloud portal to support a newer Keyrock so the impact on the users was minimal. Once the heating problem was addressed (within a couple of hours) the nodes were rebooted and the system came back to the operational state.

The maintenance team has been tracking issues filed on the JIRA most of which relate to floating IP issues at present. Some other issues have been identified pertaining to the Zurich node, but mostly these are more general issues or can be resolved by adjusting the configuration of the tenant’s account.

2.9.3 Current Status

The system passes the full set of tests in the test suite developed by Telefonica (05/03/2015). (There is also a local variant of the same test suite which is run locally nightly and circulates the results to the team internally via email).

2.9.4 OpenStack Configuration

The keystone was configured to point at the Spanish keystone as per the XiFi architecture. The following components were installed on the monitoring node:

- Nagios
- NGSI Adapter
- Context Broker
- Openstack Data Collector

Some of the default service configurations were also modified as the default settings did not reflect the typical use of the system - eg the neutron networking service starts many processes on the networking node (which is also the controller in our configuration). The default configuration settings were that only a small amount of threads were expected for this function; we had to increase the number of allowed threads within nagios from 1 to 49 for the neutron processes.

2.9.5 User-base

List of the 10 over 80 first tenants Usage from 2015-02-18 to 2015-03-19:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk GB-Hours
5854	5	853481.51	833.48	8334.78
139	3	742227.03	724.83	7248.31
10778	10	1618720.94	790.39	15807.82

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk GB-Hours
7868	3	2288177.09	1517.33	22380.67
11898	7	2425844.00	1184.49	23689.88
11490	6	1609880.42	1268.28	15721.49
5301	5	771217.89	753.14	7531.42
11916	3	894518.01	873.55	8735.53
11919	23	1166632.95	907.22	11294.75
7018	2	1566179.50	764.74	15294.72

Table 19: List of tenants on the Zurich node

2.10 Poznan (PSNC) Node

2.10.1 Description

PSNC node has 20 8-cores servers with 8-12 GB RAM with connecton to SAN utilizing 5TB of external storage. The node was deployed using Fuel 5.1.1. It is composed of Fuel node, 1 controller, 1 storage and 17 compute nodes. There are 2 external networks: public with a /27 subnet and a MDVPN XIFI federated network /24 subnet.

2.10.2 Experience on Support and Maintenance

First deployment of Openstack in XIFI project was installed using ITBox with Grizzly version. We eccountered many issues and bugs in Grizzly version, so we decided to deploy Icehouse from scratch. The Icehouse version seems to be very stable comparing to previous version. Because there is no ITBox version based on Icehouse, We used native Mirantis Fuel 5.1.1 and then we installed monitoring components according to XIFI deliverables instructions.

2.10.3 Current Status

All components of PSNC node are fully operational and stable.

2.10.4 OpenStack Configuration

Network: Controller/network node has configured two br-ex interfaces - one for Internet connection and second for XIFI federated network. Compute nodes has only one br-ex connected to XIFI federated network. Because of lack public ips and due to security concerns We decided to assign public ips on demand. **Storage:** Storage node is connected to SAN by 2-channel FC card and has assigned 2 TB and 3 TB LUNs. The storage hosts:

- Glance image folder mounted on controller
- Nova instances folders mounted on every compute node to the common shared folder
- Cinder volumes
- Backup data

Monitoring: The node is monitored by Nagios 3.4.1, NGSI Adapter 1.1.1 and OpenstackDataCollector. Besides standard monitoring configuration, Nagios is checking floating ip and ssh service on one assigned monitored instance, to check if public networking is working well.

2.10.5 User-base

List of the 10 over 39 first tenants Usage from 2015-02-04 to 2015-03-05

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk Hours	GB-
11135	2	331.09	0.27	6,59	
11269	1	281527.06	137.46	4123.93	
11233	80	246.90	0.12	3.62	
7161	3	1932770.96	1447.74	36712.08	
8916	3	3096578.05	2016.00	53760.04	
8485	3	4128770.73	2016.00	60480.04	
9929	1	281635.71	137.52	4125.52	
9884	2	2752513.82	1344.00	40320.03	
11716	2	740758.50	361.70	10850.95	
3938	1	344064.23	672.00	13440.01	

Table 20: List of tenants on the PSNC node

2.11 PiraeusN (Neuropublic) Node

2.11.1 Description

Neuropublic is a private company located in the city of Piraeus in Attica, Greece. The node hosted in the Neuropublic Data Center is running Openstack Icehouse installed on Ubuntu 12.04 operating system with the use of the Fuel tool created by the Openstack community and released by Mirantis. There are currently nine (9) servers running in the node infrastructure which employs a High Availability architecture. Storage wise Neuropublic node relied on Ceph in order to deploy a highly available distributed storage structure. Ceph is hosted on three servers and there are three copies of the VM data kept. In the networking part Neuropublic node is connected to the internet via fiber optic leased lines running at a speed of 100Mbps. Also there is a federation connection via a L2VPN to the local NREN (GRNET). The federation network connection runs at 300Mbps. All servers are connected to two 1Gb ethernet ports with one port carrying untagged traffic to the Fuel vlan and another port carrying vlan tagged traffic for all other networks. In the Neuropublic node a user can instantiate a VM based on all FIWARE GE images converted to a format that works with Ceph storage backend. Finally Neuropublic hosts its instance of monitoring tools available with the federation (namely ngssi_adapter, ngssi_event_broker, context broker) along with a nagios backend to monitor the infrastructure state.

Neuropublic XIFI node consists of the following equipment:

Server Equipment

- 1x Dell R320 Server hosting Mirantis Fuel v5.1.1.

The hardware specifications of the Fuel server are:

8 GB of RAM

1x Quad Core Intel Xeon E5-2403v2

2x 300GB SAS HDD in Raid-1 (mirroring) array

Dual 1Gbit network adapter

- 3x Dell R520 Server hosting Controllers and Ceph Storage nodes.
The hardware specifications of the Controller/Ceph servers are:
16 GB of RAM
2x Quad Core Intel Xeon E5-2403v2
2x 600GB SAS HDD in Raid-1 (mirroring) array
3x 2TB SAS HDD in Raid-5 array for Ceph
Dual 1Gbit network adapter
- 4x Dell R520 Server hosting Compute nodes.
The hardware specifications of the Compute servers are:
64 GB of RAM
2x Twelve Core Intel Xeon E5-2695v2 (48 cores per server including Hyper Threading)
2x 600GB SAS HDD in Raid-1 (mirroring) array
Quad 1Gbit network adapter
- 1x Custom Made PC/Server running Proxmox Virtual Environment hosting the Monitoring Components.
The hardware specifications of the proxmox server are:
1x Intel Core 2 Quad Q9400 CPU
6 GB of RAM
1x 250 GB HDD for the VM's
1x 320 GB HDD for the image transfer VM
2x 1Gbit network adapters

Network Equipment

1x HP 3800-24G-2SFP+ OpenFlow Capable Switch.

1x Cisco 3845 Router.

1x Kerio Control Firewall Appliance.

The topology of the node corresponds to the following diagram:

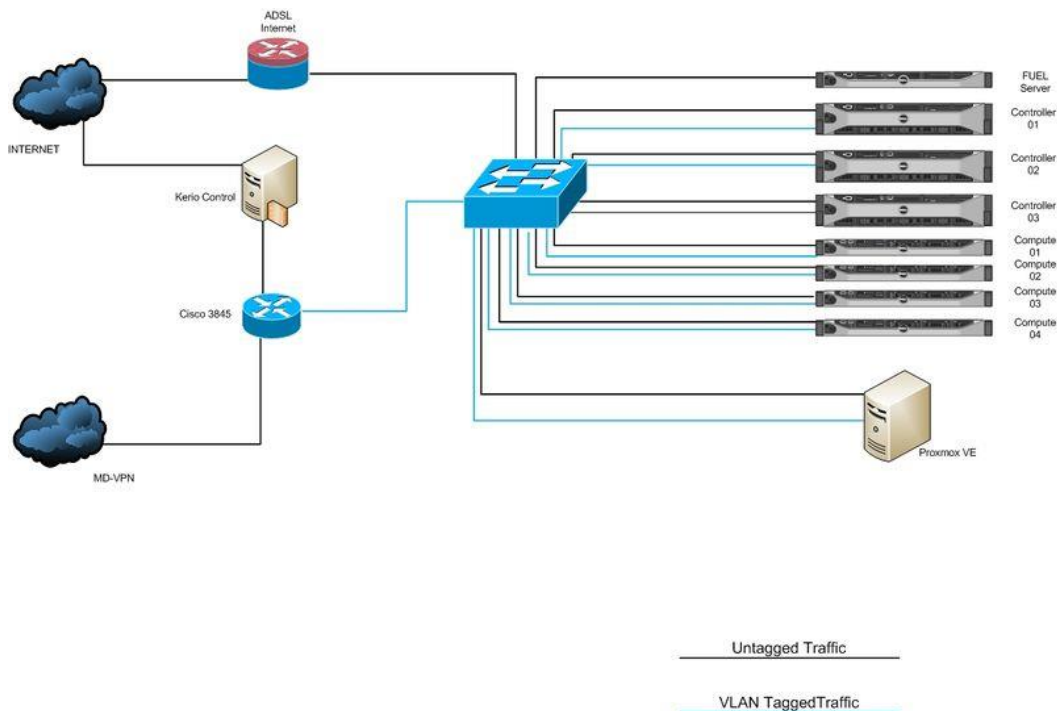


Figure 12: Architecture of Neuropublic node

2.11.2 Experience on Support and Maintenance

The node installation of Neuropublic consists of two phases. Each of these phases posed its own problems and solutions which brought Neuropublic deeper knowledge and experience on Openstack.

Phase 1 - Grizzly Deployment

In the first phase the Neuropublic node consisted of one ITBox server, one Controller server, one Cinder server, one Monitoring server and four Compute servers.

High Availability Setup

In the first deployment of Openstack Grizzly using ITBox 1.3.4 we attempted to deploy a node with High Availability. Various issues arose with the setup of corosync and pacemaker especially concerning the second external network. The synchronization of the controllers never seemed to work properly and the node could not be deemed functional. After several different configurations and due to the pressure of time we opted to deploy a non-High Availability node with one controller server.

Monitoring Node Setup

The monitoring node was deployed by ITBox automated installation tool. There were some issues though which resulted in a broken installation with some components not being installed. 1. Nagios. Nagios was not installed on the monitoring node. Moreover in all other nodes the NRPE plugin was not installed correctly. After testing a newer beta version of ITBox with the same result we proceeded and installed Nagios and NRPE on all nodes manually from sources. 2. NGSI Adapter The `ngsi_adapter` although present on the monitoring node was not installed properly. We downloaded source files from the xifi svn followed the guidelines in the readme file and deployed `ngsi_adapter` successfully. 3. NGSI Event Broker The `ngsi_event_broker` nagios plugin was not present in the respective directory (`/opt/fiware/ngsi_event_broker/lib/`) so we had to compile it from the sources in the xifi svn according to the readme file.

Phase 2 - Icehouse Deployment

In the second phase with the Icehouse deployment the Neuropublic node consists of one Fuel server, three controller/Ceph servers, four compute servers and a proxmox virtual server with the monitoring components. The second phase of deployment was the most painful and learning experience as Neuropublic committed to migrate to Icehouse before the other nodes. This experience also produced a series of configurations guides circulated to the XIFI node community via WP5 mailing list.

Second external network setup

Icehouse version of openstack gives the choice of using one L3 agent to run two separate external networks. As the two L3 agent setup with a lot more configuration required caused some issues in the Grizzly setup we opted for the one L3 agent approach. We tested various configuration scenarios until we finally got to one that worked. The working configuration was then circulated to all the other nodes via the WP5 mailing list.

Ceph storage backend for all openstack needs (cinder, glance, nova, swift)

We chose Ceph in order to satisfy our storage needs as it offers a cheap and reliable shared storage plan and it also enables vm migration. We had an issue when the images added to our glance took a long time to deploy (6-10 minutes for an Ubuntu 14.04 cloud image). After lots of investigation and mail exchange in the WP5 mailing list we found out that Ceph copy-on-write feature does not work if you use qcow2 format for images. Instead you have to use RAW format. We tested raw images and deployment time decreased dramatically (5-10 seconds for an Ubuntu 14.04 cloud image).

FIWARE image deployment

FIWARE images are in qcow2 format. After a discussion with Telefonica we decided that since they do not have space in their node to have both qcow2 and raw images Neuropublic will acquire the images, convert and upload them in our glance. We successfully arranged with UPRC the transfer of the images, we converted the ones needed to RAW and uploaded them in our glance.

Monitoring Issues

We had various issues with the monitoring servers. Due to XIFI compatibility with an older context broker version (0.13) and not with the more recent (0.13 and up) we could not deploy all our monitoring components to one single server. We opted for a small server on which we installed proxmox virtual environment and deployed an Ubuntu node hosting NGSI adapter and NGSI event broker and a CentOS node to host the context broker. On the CentOS machine the libraries the context broker needs are auto installed and that offers a better compatibility instead of porting (forcing it essentially) to install on an Ubuntu machine.

Giving back to the community

Through the experience acquired from Openstack deployment for XIFI Neuropublic was able to spot and file a couple of bugs regarding Fuel. These are the following

<https://bugs.launchpad.net/fuel/5.1.x/+bug/1407307>

<https://bugs.launchpad.net/fuel/+bug/1408935>

User Support

Neuropublic was able to answer and resolve user related issues and bugs filed in Jira bug tracking tool.

2.11.3 Current Status

Neuropublic node is currently running OpenStack Icehouse version with no apparent issues. The FIWARE GE images are present and properly converted to RAW format so they can be used with the Ceph storage backed used in the node.

2.11.4 OpenStack Configuration

Neuropublic node runs Openstack Icehouse version deployed on Ubuntu 12.04.1 with the use of Mirantis Fuel deployment tool version 5.1.1. Apart from the core components of the node Neuropublic uses a proxmox VE server which hosts the monitoring virtual machines. There is one virtual machine running Ubuntu 12.04.5 on which ngisi_adapter (version 1.1.1) and ngisi_event_broker (version 1.3.1) run, and one virtual machine running CentOS 6.5 on which the Orion context broker (version 0.15) is located.

Neuropublic deployed a High Availability architecture that consists of

- 1x Fuel deployment server used to deploy the whole environment.
- 3x Controller Nodes which also host the neutron services and the Ceph storage services.
- 4x Compute nodes
- 1x Proxmox VE hosting the monitoring VM's

2.11.5 User base

The usage of the node during the past month is as follows:

List of the 10 over 29 first tenants Usage from 2015-02-04 to 2015-03-05 in the PiraeusN node:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk GB-Hours
11615	6	119753.22	232.53	13.67
5216	5	333.65	0.65	0.00
11515	4	154608.11	301.97	0.00
8930	3	4461.51	3.33	53.82
11653	3	1255725.98	613.15	18394.42
3809	2	688128.04	1344.00	0.00
11290	2	994182.31	682.52	12592.43
3373	2	2752512.18	1344.00	40320.00
3005	1	1370608.80	669.24	20077.28
8916	1	1376256.09	672.00	20160.00

Table 21: List of tenants on the Neuropublic node

2.12 PiraeusU (UPRC) Node

2.12.1 Description

The PiraeusU node is located in Piraeus, Attica in Greece. The node hosts the ninth release of OpenStack, aka OpenStack Icehouse, which has been installed over CENTOS 6.5. It is comprised by 6 different nodes that are allocated as following:

- 1 x Controller node (8 Cores, 16 GB RAM, 1 GBps Connectivity, 0,6TB storage).
- 3 x Compute and Cinder node (32 Cores, 64 GB RAM, 1 GBps Connectivity, 0,5TB storage).
- 1 x Compute and Cinder (16 Cores, 32 GB RAM, 1 GBps Connectivity, 0,5TB storage).
- 1 x Monitoring node (8 Cores, 16 GB RAM, 1 GBps Connectivity, 0,2TB storage).

The servers are connected with each other with 1GBps over a Cisco switch (Cisco Catalyst 3560). Moreover, the node support the connectivity over the Internet and the XIFI MDVPN, whereas it allows the allocation of Public and MDVPN IPs to the hosted VMs through the FIWARE cloud portal web-based UI.

Specifically, the node offers two (2) external networks;

a) a public External network offered and supported by the University of Piraeus network infrastructure, (providing 64 Public IPs) and b) a MDVPN external network (federated IPv4 range 10.0.208.0/20).

Furthermore, the node allocated 33 different FIWARE GE images that can be successfully used for the instantiation of VMs on the node, by each tenant. Further to the above, the node has already started to be integrated with the IoT infrastructure that is installed in the UPRC building in Piraeus, whereas there is a first integration between the IoT infrastructure custom software modules with the Orion Context broker (v0.13) that allows the interaction with the IoT infrastructure (retrieval of measurements, pushing commands to actuators), by using the Orion GE facilities. The next steps includes the integration with NGSi9/NGSi10 interface towards NGSi enabled Orion CB.

2.12.2 Experience on Support and Maintenance

The Mirantis FUEL 5.1.1 release has been used for the installation and the setup of the OpenStack Icehouse on the PiraeusU node. The configuration of the FUEL parameters for the installation and the setup of the node, included the following core options that refer to the hypervisor, the storage and the networking components. In particular:

Hypervisor: KVM.

Storage nodes: Cinder LVM over iSCSI for volumes.

Networking: Neutron agent with OVS VLAN splinters hard trunks workaround enabled.

After the successful setup of the node, and before its federation, the node was tested by exploiting the OpenStack CLI API, as well as the capabilities provided by OpenStack Horizon dashboard Web-based UI. After some specific reconfigurations in the FIWARE OS services endpoints the node was provided for external use over the FIWARE Cloud portal, while it is monitored through the XIFI monitoring infrastructure.

For the federation of PiraeusU node, after its successful setup, there were some reconfiguration options in the existing setup, so as to achieve the integration with the FIWARE infrastructure. In particular:

Update of the corresponding point(s) that refer to the authentication service endpoints was performed, so as to be compatible with the keystone authentication process, which is hosted by XIFI infrastructure in the public keystone in the region of Spain.

Update of the neutron, nova and cinder configuration files was performed in the controller node so as to agree with the credentials that correspond to the PiraeusU node software agents.

Further to that, manual setup and configuration of the Node Monitoring System (NMS) in our infrastructure was executed. For monitoring, our NMS includes the Nagios 3.4.1 over Ubuntu 12.04.1, while a set of additional components for the XIFI monitoring were included and configured as to achieve a successful federation in the monitoring part, as well. In particular:

The latest version of the ODC on the controller node was installed, and the corresponding odc.conf file was updated with the appropriate configuration options, in order to provide access to the PiraeusU node information that, among others, are related with the available VMs, active user, available networks, active floating IPs, and so on.

The FI-WARE NGSi Adapter (v1.1.1) and the FI-WARE Context Broker GE - Orion (v0.13) were installed.

For the realization of the above steps, existing guides and deliverable documents that have been written in the context of the XIFI project were exploited. Namely :

- a) For the OpenStack Data Collector module: <http://wiki.fi-xifi.eu/Public:OpenstackDataCollector>,
- b) For the Orion Context Broker (v0.13): <http://catalogue.fi-ware.org/enablers/publishsubscribe-context-broker-orion-context-broker>,
- c) For the NGSI Adapter (v1.1.1): https://github.com/telefonicaid/fiware-monitoring/tree/v3.5.2/ngsi_adapter,
- d) For the Infrastructures monitoring and interoperability adaptation components toolkit and API (Revision v.1.1), the deliverable D3.2.

Finally, after the finalization of the above setup, the requested information for the public access on PiraeusU Orion GE instance (Public Context Broker) was filled at http://wiki.fi-xifi.eu/Xifi:Wp5:Context_Broker_Public_IP_Address.

2.12.3 Current Status

Currently, the PiraeusU node works properly on OpenStack Icehouse, and it is accessible through the FIWARE cloud portal at <https://cloud.lab.fiware.org/>, as well as, it is publicly monitored by the XIFI monitoring infrastructure in the Infographics (<http://infographic.lab.fi-ware.org/>) and node status (<http://status.lab.fi-ware.org/>) web-pages. In addition, the node offers 33 FIWARE GE images that can be used by tenants so as to instantiate their own VMs on the infrastructure.

2.12.4 OpenStack Configuration

The PiraeusU node hosts OpenStack Icehouse release on CentOS 6.5 with KVM hypervisor, cinder storage and neutron network module. The configuration of the OpenStack services (nova, neutron, cinder, glance, and so on) have been deployed in respect of what is described in the deliverable document D5.2 XIFI Core Backbone, whereas for additional configurations the deliverables D.2.1 and D.2.4 that constitute the XIFI handbooks were used.

Particular configurations have been performed with respect to the XIFI project guideline about the allowed usage and resource consumption by each user connected to the node. Specifically some of the most important configurations refer to the network quotas, the VCPUs, the RAM and of course the number of the allowed instances per tenant. For the quotas configuration the OpenStack configuration has been defined as; a) floating IPs: 1 per tenant, b) fixed IPs: 10 per tenant, c) VCPUs: 6 per tenant, d) RAM: 6GB per tenant and e) 3 instances per tenant. The table underneath, presents an integrated view of PiraeusU quotas configuration.

Further to the above, the OpenStack configuration includes some additional options that had to be performed for the successful federation of the node in the public XIFI infrastructure. The next tables present in detail the set of configurations, by making a reference to the OpenStack node, as well as to the configuration file that should be updated so as to achieve the final result that correspond to the successful federation of the node. It should be highlighted that the following configuration was performed by following the corresponding guidelines that were circulated by XIFI project partner Neuropublic.

2.12.5 User-base

The PiraeusU node provides resources to various user tenants that are associated with the FIWARE platform. In total, for now, there are 25 unique users on the node, while the table below presents in details the resource usage per tenant.

List of the 10 over 25 first tenants Usage from 2015-02-03 to 2015-03-04:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk GB-Hours
11233	65	348.73	0.17	5.11
4259	8	1384469.72	676.01	20280.32
3846	8	4128177.22	2015.71	60471.35
11653	6	1166396.51	569.53	17085.89
10565	6	2345151.14	1145.35	34350.27
10340	5	3700385.25	1806.83	54204.86
10114	4	688505.24	336.18	10085.53
11018	4	1796645.68	877.27	26318.05
3005	4	2679505.49	1308.35	39250.57
10421	4	1376137.77	671.94	20158.27

Table 22: List of tenants on the UPRC node

2.13 Volos (UTH) Node

2.13.1 Description

NITOS cloud infrastructure is located in Volos, Magnesia in Greece and is hosted in the premises of the University of Thessaly. The infrastructure is based on HP G8 and G9 blade servers and an HP DL380p GEN8 one. Each blade server has two powerful 16-core Intel Xeon processors, 96 GBs of RAM and two 300GB 6G SAS 10K HDDs, formatted in level one (1) RAID. In addition, each unit consists of two 10GB and four 1GB Ethernet interfaces used for the deployment of the hardware's virtualization, as well as for the connection with the shared storage unit, which has 9 Tb of storage capacity. As far as the rack mounted server is concerned, it comprises of two 12-core Intel Xeon processors, 64 GBs of RAM and five 450GB 6G SAS 10K HDDs, formatted in level five (5) RAID. Regarding the networking hardware, two 10GB and four 1GB Ethernet interfaces are used as well.

As far as the networking is concerned the HP 5412r modular switch is used. The switch hosts 4 switch modules, 2 of which are 1Gb and the residual ones are 10Gb. The 10Gb links are used for the storage and management in order to provide a seamless and robust environment interconnection. The entire system is supported by UTH's Network agency, i.e. NOC, which in turn provides a range of 64 public ips in service of Openstack's public external network. In advance, a second external network is configured and provided connecting NITOS node and its tenants' to the XIFI's md-vpn network.

Last but not least, NITOS node hosts 33 different FIWARE GE images that can be used by the tenants in order to initiate instances. Furthermore, we are in the process of integrating the cloud part of NITOS infrastructure with the wireless and broadband one, i.e. LTE and WiMAX basestations, wireless Sensors, ICARUS nodes, etc. Towards this goal we are elaborating the Orion Context Broker, the NGSI adapter and the corresponding GE images.

<u>Components</u>	<u>Description</u>	<u>Comments</u>
Servers	7 x HP blade servers 1 x HP DL380p server 1 x Storage Server	
Total Capacity	<ul style="list-style-type: none"> · CPU: 256 Cores · RAM: 784 GB · HDD capacity: 13 TB <ul style="list-style-type: none"> o Shared storage(CEPH): 9TB 	
Per server capacity	Blade Servers <ul style="list-style-type: none"> • CPU : 2 processors (32 cores per server) • RAM : 96 GB • HDD (local): 2 x 300 6G SAS 10k disks • Network: <ul style="list-style-type: none"> o 2 x 10Gbit Ethernet NIC o 4 x 1Gbit Ethernet NIC DL380p server <ul style="list-style-type: none"> • CPU : 2 processors (24 cores) • RAM : 64 GB • HDD (local): 5 x 450 6G SAS 10k disks • Network: <ul style="list-style-type: none"> o 2 x 10Gbit Ethernet NIC o 4 x 1Gbit Ethernet NIC Storage Server: <ul style="list-style-type: none"> • CPU: 1 processor (8 cores) • RAM: 16 GB • HDD: 9TB • Network: 2x 10Gbit Ethernet NIC 	

Switch	HP 5412r	OpenFlow 1.3
Firewall	Software solution	

Figure 13: Server equipment UTH node

2.13.2 Experience on Support and Maintenance

NITOS's successful attempt to deploy Openstack Grizzly environment through ITBOX suite was a demanding, yet challenging, procedure. Our servers were not compatible with the version of the suite and we had to find work arounds in order to conclude the deployment of the node. The incompatibility is well documented (<https://bugs.launchpad.net/fuel/+bug/1312311>) and resolved in newer versions of the suite. In advance, we faced some problems that had to do with the Grizzly version of Openstack. The problems and the solutions, proposed and applied, are documented here: http://wiki.fi-xifi.eu/Xifi:Wp5:t5.5#Grizzly_issues

The aforementioned procedures elevated both our experience and expertise in the field of Openstack. As a result, the Icehouse HA migration was a smoother process to conclude. For the latter, we used the Mirantis Fuel5.1.1 suite, for the openstack deployment and XIFI's documentation for Monitoring and Federation. Nevertheless, the process was not a straight forward one. There were sections that were not applicable in the case of HA Openstack. For example, the monitoring installation for each controller dictated the download and set up of an nrpe agent. We discovered that, after the finalization of the procedure the system was not functional anymore and this was the result of an install/ uninstall misbehavior on the system. In order to resolve this issue we had to re-install the Openstack environment and replace some of the given installation procedures with alternative ones.

2.13.3 Current Status

NITOS node is currently running Icehouse HA Openstack version. The node comprises of 3 controller node, 7 compute nodes, 1 CEPH node and a monitoring node. The federation with the FiWARE Lab Platform and the rest of the XIFI cloud nodes has been completed and the node is accessible via the federated Openstack platform (<https://account.lab.fiware.org/>). In advance, we integrated the federation of the monitoring components, i.e. the infographics (<http://infographic.lab.fiware.org/status>) and the status tool (<http://status.lab.fi-ware.org/>). Last but not least, the node hosts the 33 official FiWARE images, which can be used by the tenants in order to launch virtual machines.

2.13.4 OpenStack Configuration

NITOS node hosts the Icehouse Openstack version deployed via the usage of the Mirantis Fuel 5.1.1 suite. This solution was preferred since the official ITBOX XIFI tool was not yet available. The overall implementation performed is aligned and in respect to the deliverables 2.1 and 2.4, that constitute the official XIFI handbooks, as well as to the deliverable 5.2, which provides the official description of the Core Backbone Connectivity built process.

The NITOS node consists of:

- 3 Controller nodes in HA mode.
The controller nodes also include the neutron services. This is configured with one external network, with a prefix of 26, and a public internal one, as well as an md-vpn external network and a public internal md-vpn one
- 7 compute nodes.
- 1 Monitoring node running on Ubuntu trusty OS.
 - 1 NGSI-Adapter: 1.1.1
 - 2 NGSI-EventBroker: 1.3.1
 - 3 Context-Broker:0.17
- 1 shared storage node using CEPH framework.

The CEPH environment dictates the usage of raw images instead of qcow/2 ones in order to work properly. As a result we are in the process of downloading converting and re-uploading the official XIFI images whenever this is requested. In advance the CEPH handles all the different needs of the environment for storage, i.e. ephemeral, object, etc. The storage backbone network is a 10G one attached on an HP modular openflow switch.

2.13.5 User-base

The different users/ tenants registered to the FiWARE Lab and requesting resources are nine. The 7 tenants and their usage statistics, collected from the Openstack's nova service, are presented below.

Usage from 2015-02-04 to 2015-03-05:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk Hours	GB-
3920	1	166611.84	81.35	1627.07	
4017	1	165342.08	80.73	1614.67	
6a76dc53ebee 424397c5b249 d6743ce7	4	3410.49	1.67	33.31	
11233	15	44.37	0.02	0.43	
11660	1	211184.13	103.12	2062.34	
5805	9	140512.74	68.61	1372.19	
7968	1	8083.37	3.95	78.94	

Table 23: List of tenants on the UTH node

2.14 Sophia Antipolis (Com4Innov) Node

2.14.1 Description

SophiaAntipolis node is ran by Com4Innov and is located in Sophia-Antipolis in the South-Eastern part of France. Business wise, Sophia Antipolis is the largest advanced technology park in France.

Sophia Antipolis node used ItBox 2.3.4.1 to deploy Openstack and current XIFI's tools. The deployment uses and respects the technical requirements described in D5.1. The node is composed by a virtual server 2 cpus / 2 gb of Ram for the Fuel/itbox usage.

- 3 Dell poweredge R820 providing 96 cores and 192 gb of Ram memory with 4,8 TB of storage (3 x 1,6 TB per host in RAID 5) to delivering the Openstack/XIFI current services.
- It uses a 1Gb internal network link vlan-based in our Datacenter. The Internet connection is provided by the datacenter's Internet service provider with 1Gb link.

The node is connected to the MDVPN by a dedicated link based on Vlan provided by RENATER, the NREN which also provides Public IP subnet 193.48.247.192/26 that have been allocated.

We have allocated 10.0.224.0/20 subnet provided by the federation.

Quantity	Description
1	Dell PowerEdge R820 32 CPU Intel XEON E5-4620 at 2,2 GHz 64 Gb de Ram, 1,6 To RAID 5 ESXi 5.5 > Virtual Machine ItBox 1.3.4.1 2 cores, 2Gb RAM, 120Gb HDD
1	Server Controller: Dell PowerEdge R820 32 CPU Intel XEON E5-4620 at 2,2 GHz 64 Gb de Ram, 1,6 To RAID 5, 4 x1Gb Nic
1	Cinder & Compute Server: Dell PowerEdge R820 32 CPU Intel XEON E5-4620 at 2,2 GHz 64 Gb de Ram, 1,6 To RAID 5, 4 x1Gb Nic Volume Cinder: 1 To Volume Compute: 600 Gb
1	Cinder & Compute Server: Dell PowerEdge R820 32 CPU Intel XEON E5-4620 at 2,2 GHz 64 Gb de Ram, 1,6 To RAID 5, 4 x1Gb Nic Volume Compute: 1,6 To
1	Switch Pica8 Pronto 3290 48 Ports Openflow

Table 24: Sophia Antipolis infrastructure

2.14.2 Experience on Support and Maintenance

First deployment of Openstack in XIFI project was intalled using different itbox versions with Grizzly release and xifi's tools embedded.

We encountered many issues and bugs in Grizzly version especially in the first versions. We actually use 1.3.4.1 itbox release and we still correct some major bugs and issues due from monitoring and fiware tools. We are planning to jump to Icehouse release to migrate directly to Juno release. Before deploying this version from scratch we decided to measure all the impact and the time we need for.

ITbox does not exist on this version, we will probably use a Mirantis Fuel 6.0. We will start to test all Fiware tools to be sure about the compatibility /stability on this version before deploying to be compliant with XIFI's federation requirements.

Frequently, we do basic tasks of maintenance of the software components, like correcting configurations or upgrades and we check status and monitoring to have delivering service nodes.

2.14.3 Current Status

Currently, the Sophia Antipolis node is implemented on OpenStack Grizzly, and is accessible through the FIWARE cloud portal at <https://cloud.lab.fiware.org/>, as well as, it is publicly monitored by the XiFi monitoring infrastructure in the Infographics (<http://infographic.lab.fi-ware.org/>) and node status (<http://status.lab.fi-ware.org/>) web-pages.

Following the changes of our credentials node in the last few weeks, sanity checks tests were in error for a short period. After the update of our configuration files only 5 tests concerning volume mounting are still in error. So currently, we still have a configuration problem on our storage node to create and attach the volumes to an instance. This problem is under resolution.

As soon as this problem will be overridden the node will be capable of offering FIWARE GE images

that can be used by tenants in order to instantiate their own VMs on the infrastructure.

2.14.4 OpenStack Configuration

The Sophia Antipolis node consists of:

- 1 Controller/Monitoring node
- 1 Cinder/Compute node
- 1 Compute node
- 1 virtual fuel Itbox 1.3.4.1

Our node is deployed by Itbox 1.3.4.1. We currently installed following FIWARE's software:

- Context Broker 0.13
- NGSI Adapter 1.1.1
- Event Broker 1.3.1
- Nagios 3.5.1

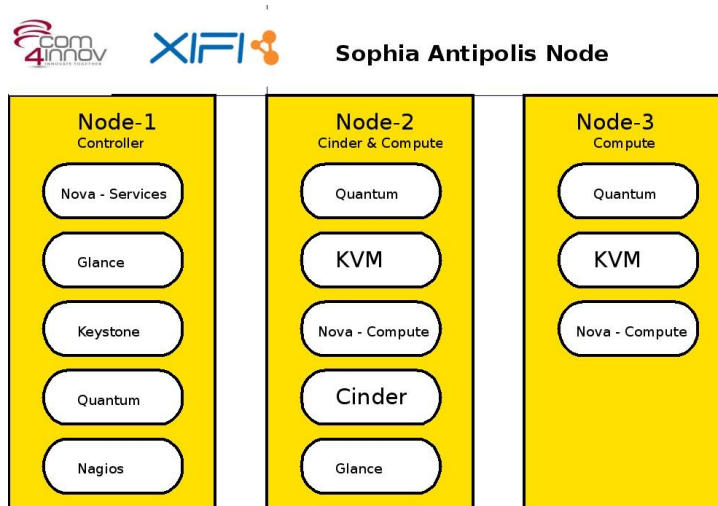


Figure 14: Server equipment Sophia Antipolis node

The default quotas for the SophiaAntipolis node for all users are depicted in the following table:

Property	Value
metadata_items	256
injected_file_content_bytes	10240
ram	4096
floating_ips	3
key_pairs	100
instances	3
security_group_rules	10

Property	Value
injected_files	5
cores	100
fixed_ips	-1
injected_file_path_bytes	255
security_groups	10

Table 25: Default quotas Sophia Antipolis node

2.14.5 User-base

List of the 10 over 32 first tenants

Usage from 2015-02-09 to 2015-03-10

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk GB-Hours
11233	43	165.55	0.08	1.62
10538	8	108184.46	52.82	1056.49
08822	4	1463827.95	714.76	14295.19
0677	4	583641.90	285.62	5695.36
05112	4	601648.02	385.87	5261.51
04463	4	984296.70	1142.0	5202.84
02107	4	11010054.42	5376.00	107520.06
05111	3	2064385.20	2016.00	13440.01
09883	3	8257540.82	4032.00	80640.05
09886	3	8257540.82	4032.00	80640.05

Table 26: Tenants Sophia Antipolis node

All users and infrastructure nodes are supported via Jira trouble ticket system. When a support ticket is opened in our cue, an email is received at the support.xifi@com4innov mailing list and is handled as a best-effort from our team. We inform and communicate to users the node status, interruption or maintenance process in such a way that it does not interfere with their project and usage. We report to the federation any abusive user's activities that can adversely affect the operation of the service or the security. In these cases, we make cleaning actions on abusive resources.

2.15 Karlskrona (BTH-Sweden) Node

2.15.1 Description

The Karlskrona Node was built using ITBox for the OpenStack (Grizzly) deployment. The nodes footprint comes in 128 cores (96 operational, 32 for IceHouse upgrade) of compute across 4 servers, each

having 128Gbyte RAM and 1Tbyte disc space. There is also an NFS server, that provides shared storage.

The Karlskrona Node is connected via 1 Gbps links to the Blekinge Institute of Technology (**BTH**) core network, which in turn connects to out NREN the Swedish University Network (SUNET). From BTH we have a C subnet (194.47.157.0/24), from this we draw our FloatingIP addresses.

2.15.2 Experience on Support and Maintenance

We deployed from ITBox, and after some tinkering with the hardware configuration it has operated properly. As the IceHouse upgrade, we probably will do it manually directly from FUEL. After federating, we have not had any major issues. We intend to setup a secondary environment for testing purposes, where we will initially install IceHouse. Once functional we will migrate the other Compute nodes here, and release the nodes used for the initial IceHouse deployment. This way we will have one controller and compute node in standby (testing).

2.15.3 Current Status

System looks stable, we perform clean up of floating IPs periodically (every 2 weeks). We've also cleaned out bad/error VMs occasionally, with the new freemium/premium users we need to do this once a week.

2.15.4 OpenStack Configuration

We are running Grizzly, neutron-vlan, without the high availability option, schematic topology shown here.

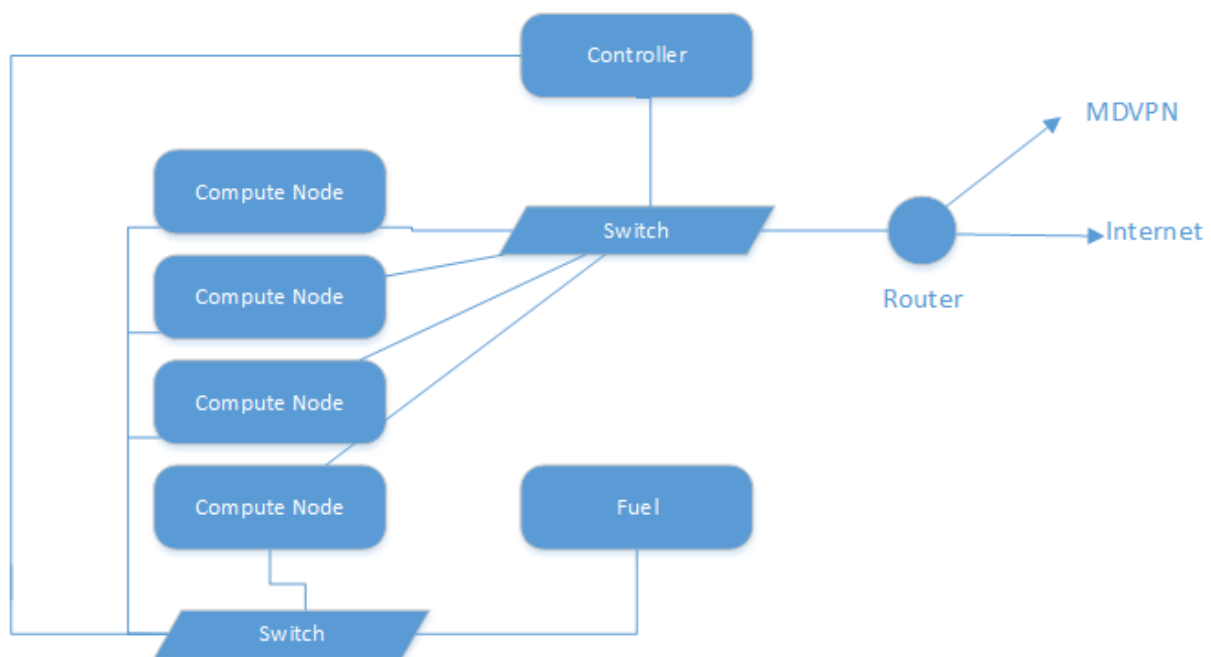


Figure 15: Server equipment BTH node

2.15.5 User-base

The Karlskrona node will host the FI-PP use case project FI-STAR as well as a set of projects coming from the accelerator projects.

List of the 10 over 28 first tenants Usage from 2015-02-04 to 2015-03-05:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk Hours	GB-
11233	70	259.98	0.13	3.81	
7865	16	737840.39	360.28	10807.50	
7241	6	1382964.81	675.28	3376.38	
4259	6	39520.63	19.30	578.92	
10538	5	303046.09	147.97	4439.15	
10021	3	4154347.99	2028.49	10142.45	
7171	3	29535.94	57.69	0.00	
9985	2	353943.74	691.30	0.00	
9742	2	1245760.68	608.28	3523.08	
6778	2	1415615.10	691.22	14016.55	

Table 27: Tenants BTH node

2.16 ACREO Swedish ICT Node

2.16.1 Description

The Acreo node used ITBox to deploy OpenStack.

The node is physically distributed between a site in Stockholm and a site in Hudiksvall. It comprises three compute nodes of 32 cores, 128GB of RAM and 2TB of disk each. Two virtualisation servers with a total of 8 cores, 48GB of memory and 4TB of disk (2x2TB in RAID1 per host) are used to host the various openStack/XIFI services as virtual machines: one controller guest, one cinder guest and one monitoring guest.

Internally (on each site and between the two sites), we use 1Gb, vlan-based network links. We are connected to the Internet via a 100Mb link to our ISP. We have allocated a /24 subnet of public IPv4 addresses (194.28.122.0/24) to XIFI.

We are connected to the XIFI federated network through Sunet using a 1Gb link. We received a /20 subnet on the 10.0.0.0 network range from the XIFI project.

2.16.2 Experience on Support and Maintenance

We deployed using ITBox and encountered a number of bugs, including one due to an incompatibility with the AMD-based servers that we use as compute nodes: the bug resulted in an incomplete OpenStack deployment that we used as a base. The remainder of the deployment was performed by hand, with the support from our support IO node in Waterford.

2.16.3 Current Status

We are using the standard OpenStack deployment configuration provided by XIFI. As a newcomer to the project, we chose not to add any custom services, to avoid interfering with the specific services developed by XIFI and Fi-ware (such as monitoring adapters).

As pointed out by others, we are very reluctant to making changes to our federated environment: third parties have deployed virtual machines on our node and we are very wary of causing disruption to any active tenant. Hence, we keep changes in configuration and services to a minimum.

At the moment, we are using only one of our three compute nodes (32 cores) in the federated environment. This allows us to use the reminder of the hardware for staging and prepare for future upgrades.

2.16.4 OpenStack Configuration

The OpenStack Cinder and monitor each reside on one guest virtual machine on a virtualisation host. The controller, Glance and Quantum services reside on the same guest virtual machine.

2.16.5 User-base

The Acreo node has an increasing number of guest tenants, who register through the Fi-Ware portal with little prior knowledge of Acreo. Several Acreo partners have shown interest in using the XIFI resources. The European project MobiS, which Acreo is part of, is in the process of moving its servers to XIFI. Another partner wants to deploy a big Hadoop cluster and is investigating if the XIFI platform is stable enough for that purpose.

Usage from 2015-02-03 to 2015-03-04:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk GB-Hours
000135	2	1720320.11	1344.00	13440.00
003807	1	237345.36	463.57	0.00
004979	1	11010048.71	5376.00	107520.01
005584	4	2918384.87	1789.51	26069.75
005768	7	4523788.47	2885.02	39670.06
007161	2	8257536.53	4032.00	80640.01
007827	4	2614040.10	1276.39	25527.74
008392	1	5505024.36	2688.00	53760.00
008715	6	1041707.57	624.30	9401.91
008822	3	1162456.60	567.61	11352.12
009785	1	2752512.18	1344.00	26880.00
009985	1	344064.02	672.00	0.00
000179	1	1376256.09	672.00	13440.00
010384	3	61156.46	29.86	597.23
010684	1	334054.06	652.45	0.00
011233	65	263.96	0.13	2.58
011283	1	51865.94	25.33	506.50
011432	2	1114111.54	544.00	10880.00
011660	6	1317581.94	643.35	12867.01

Table 28: Tenants ACRO node

2.17 Budapest (WIGNER) Node

2.17.1 Description

The Budapest node is operated by Wigner Research Centre for Physics (Wigner RCP) located in Budapest, Hungary. The provided infrastructure is based on a SGI ALTIX ICE 8200XE cluster and an external SAN. Originally, one individual rack unit (IRU) of the cluster was dedicated to XIFI, but since February 2015 it has been extended with a second IRU to serve the increasing number of users. Each IRU consists of 16 blade diskless servers with powerful 8-core Intel Xeon processors, 16 GBs RAM and one 1 Gbps Ethernet and two 20Gbps InfiniBand interfaces. Currently, two external storage

servers of capacities 10TB and 50TB are used as SANs (RAID 5) to provide the blades with file systems as well as to store users' volumes and objects. Both are mounted via the ultrafast InfiniBand interfaces. The Ethernet interface of the blade servers are used for management traffic only, and all the data traffics are forwarded through InfiniBand links. The gateway node of the cluster connects to the outside network through two 1 Gbps Ethernet interfaces and to the blade servers via a 20 Gbps InfiniBand interface.

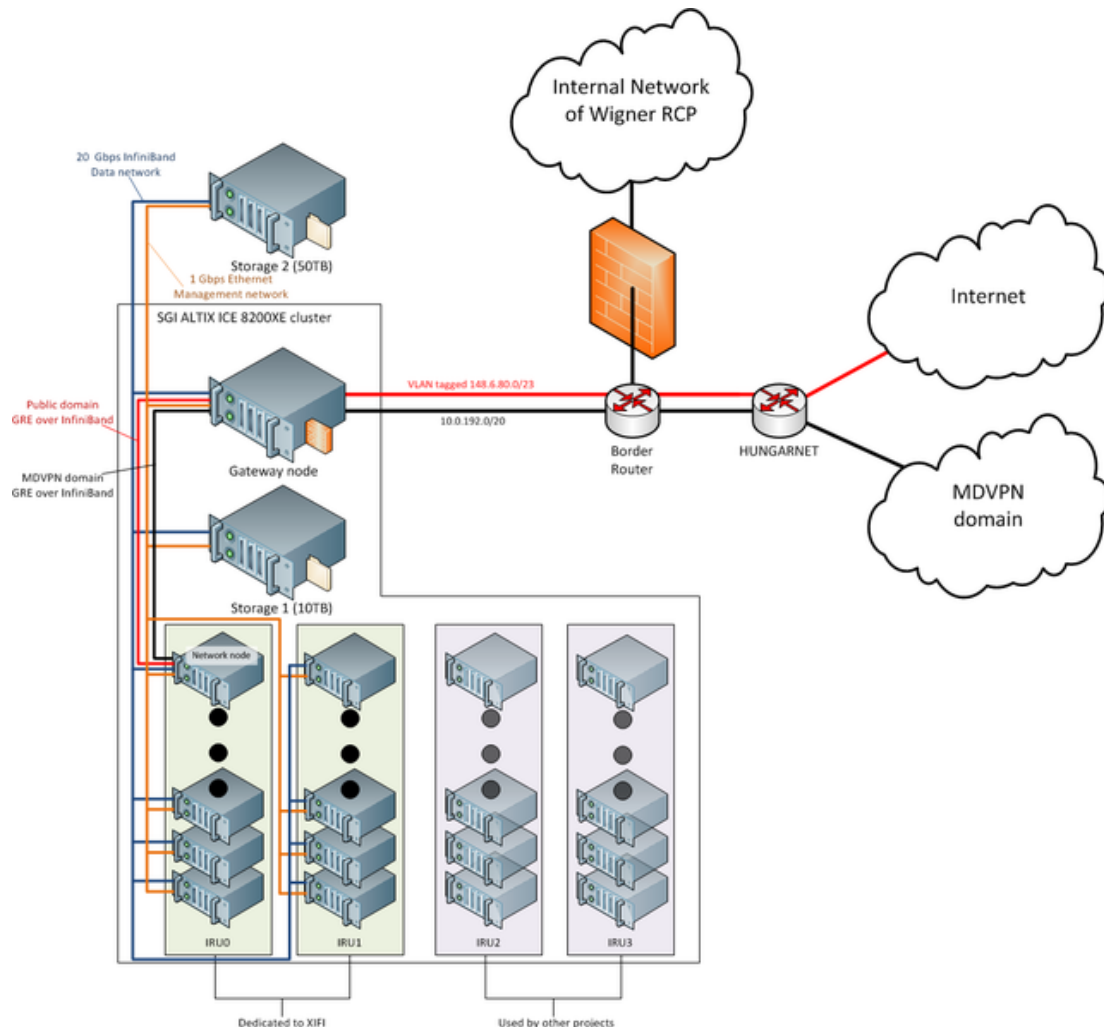


Figure 16: Architecture of Budapest node

Currently, the gateway node is connected to a 10 Gbps switch that is directly linked to the border gateway router of the research centre via a 10Gbps optical cable. The border gateway is connected to HUNGARNET (the Hungarian NREN) via a 10 Gbps link and a 1 Gbps backup link. Currently, there are two external networks deployed on our configuration: A public External network with a /23 subnet of public IPs (148.6.80.0/23 with almost 256 IPs dedicated to XIFI) and a MDVPN external network with a /20 subnet.

Note MDVPN is provided by HUNGARNET. Figure 16 depicts the high-level architecture of the Budapest node. Note that the public /16 IP range of the research centre is rarely used, and thus the above XIFI range could be extended in the future.

- Servers:
 - 1 2x IRUs of SGI ALTIX ICE 8200XE (2x 16 blade servers)
 - 2 2x storage servers

- Total capacity:
 - 3 CPU: 256 cores
 - 4 RAM: 512 GB
 - 5 HDD: 60TB on shared storage servers

Per server capacity:

- Blade servers:
 - 6 CPU: 2 processors (8 cores)
 - 7 RAM 16GB, HDD: diskless
 - 8 Network: 1x 1Gbps Ethernet NIC, 2x 20Gpbs InfiniBand NIC
- Storage server 1 (part of SGI ALTIX ICE):
 - 9 CPU: 2 processors (8 cores)
 - 10 RAM 16 GB
 - 11 HDD: 10 TB
 - 12 Network, 1x 1Gbps Ethernet NIC, 2x 20Gpbs InfiniBand NIC
- Storage server 2 (in separate rack):
 - 13 CPU: 2 processors (16 cores)
 - 14 RAM 64 GB
 - 15 HDD: 50 TB
 - 16 Network, 1x 1Gbps Ethernet NIC, 1x 20Gpbs InfiniBand NIC

2.17.2 Experience on Support and Maintenance

Since the blade servers of SGI ALTIX ICE 8200XE are diskless, the ITBOX deployment proved to be unfeasible. However, during the manual installation of OpenStack Grizzly we managed to use some of ITBOX's puppet scripts. This procedure was not straight forward, but a significant support was given by TID.

Our experiences on the diskless blade server installation has been published in the XIFI Blog (<https://blog.fi-xifi.eu/openstack-deployment-on-diskless-blade-servers-the-sgi-altix-ice-story-part-i/>), and a further extended description is expected in March 2015 (published in the XIFI blog and other OpenStack blogs).

The team of Wigner RCP has also taken part in the Level-1 support team of XIFI, providing direct support to potential stakeholders. During this activity we had to get familiar with other FIWARE components and terms as well.

2.17.3 Current Status

Currently, the Budapest node works properly on OpenStack Grizzly running on 14 blade servers and the manual preparation of the IceHouse environment and the migration is ongoing. After it is finished, 28 blades will be dedicated to the IceHouse environment and 4 other blades will be allocated to testing purposes.

The Budapest node is accessible through the FIWARE cloud portal and it is monitored by the XIFI monitoring infrastructure shown in the Infographics and node status pages. As noted the resources of the Budapest node have been extended during the operation since the huge interest from the user communities, now this node hosts 149 active VMs.

2.17.4 OpenStack Configuration

The applied quotas are the following:

Property	Value
metadata_items	128
injected_file_content_bytes	10240
ram	25000
floating_ips	3
key_pairs	100
instances	3
security_group_rules	30
injected_files	5
cores	6
fixed_ips	10
injected_file_path_bytes	255
security_groups	20

Table 29: Default quotas Budapest node

2.17.5 User-base

The Budapest node provides resources to various user tenants that are associated with the FIWARE platform. After Prague node, it was one of the first new nodes that has successfully federated. In total, for now, 165 unique users have used the node, while the table below presents in details the resource usage per tenant.

List of the 10 over 165 first tenants Usage from 2015-02-03 to 2015-03-04:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk GB-Hours
11233	33	883014.54	431.16	12934.78
10778	12	1689961.98	1307.95	19927.56
9784	10	1357952.71	1302.39	13498.66
11911	9	13400.53	6.61	195.60
4259	8	2413.58	42022	35.36
8027	7	2887431.17	1409.88	42296.35
9740	7	1092421.19	533.41	16002.26
10776	7	943753.24	461.39	13818.81
1143	6	1935515.32	1762.36	20179.39
9655	5	1797525.26	965.24	25455.56

Table 30: Tenants Budapest node

2.18 Intellicloud – Crete (Associated Partner)

2.18.1 Description

The node of Crete is composed of 4 Dell C6220 II with a total of 160 Cores, 128GB of RAM and 4TB of Disks. The node was deployed in HA using ITBox v1.3.4.0. It is composed of 2 computes and 2 controllers (HA deployment). A Dell Networking N4032 switch is used to connect the servers based on a 10 Gb links.

The node of Crete relies (for Internet and MDVPN connectivity - currently on 1Gb link) on the network center of Technical University of Crete (TUC) which is connected to XIFI through GRNET via dark fiber at speeds up to 10Gbps that has access to the pan-European GEANT. Currently, there is 1 external network deployed on our configuration with a /24 subnet of Public IPs and a MDVPN network with a /20 subnet.

An extension of the node is ongoing. We are going to add to the current configuration 8 more Dell R320 servers with 48 Cores, 256GB of RAM and 8 TB of Disks. The final node when deployed will have 2 controllers and 10 computes (208 Cores, 384GB of RAM and 12 TB of Disks).

2.18.2 Experience on Support and Maintenance

Our first OpenStack installation was manual. We installed OpenStack on grizzly over Ubuntu 12.04 LTS and KVM as hypervisor with one server as Controller node, one server as Network node and 6 servers as cinder and compute nodes. This installation is still running and is used for the purposes of the FI-STAR project. It is a quite stable installation, after resolving a lot of performance problems, and now is deploying more than 70 different VMs with tenants coming from the FI-STAR consortium as well as TUC (graduate students). A new installation was made (for the purposes of XIFI federation) using ITBox v1.3.4.0 - after attending the XIFI training session, which took place on June 24 - 25, 2014 at Universidad Politécnica de Madrid. The configuration was as following:

- 1 server as master node (ITBox)
- 2 servers as Controllers (High Availability) with Monitoring (in one of the two Controllers) and Cinder (in the other one),
- 2 servers as Compute Nodes with Cinder.

The main problems we had with our node were due to the following:

- The installation through ITBox was problematic and a lot of components either were not installed properly or not installed at all even though ITBox showed successful installation. More specifically:
- Cinder was not installed in one of the compute nodes. Could not re-install it afterwards (too complicated).
- Nagios was installed but was not working properly. We had to stop the existing service and install a new version.
- `ngsi_event_broker`, `ngsi_adapter`, `openstackdatacollector` (ODC): Never installed. We had to install them manually.
- The HA (High Availability) service (using Pacemaker and Corosync) is working in a non-controllable way:
- A lot of unknown errors which affect the deployment (by the HA service) of service agents between Controllers.
- A lot of problems when we manually try to restart certain agents (e.g. `quantum-dhcp-agent`).
- HA also caused problems in the deployment of Nagios web interface (when Nagios is installed in one of the Controllers as in our case).

Beyond what proposed XIFI to monitor the federation, we choose Nagios to monitor our node. The check of the node is done as following:

- By host:
 - ping
 - CPU Load
 - Current Users
 - HDD Free space
 - Memory Swap
 - Zombie Processes
- By services:
 - cinder-api
 - cinder-scheduler
 - glance-api
 - glance-registry
 - nova-api
 - nova-cert
 - nova-conductor
 - nova-consoleaut
 - nova-novncproxy
 - nova-objectstor
 - nova-scheduler
 - quantum-metadata-agent
 - quantum-openvswitch-agent
 - quantum-server
 - quantum-dhcp-agent
 - quantum-l3-agent

Instead of backup we use mirrored disks in all servers of the node (computes and controllers).

2.18.3 Current Status

Since the correction of multiple bugs found, the node of Crete is quite stable. The support to user is handled through the Jira Helpdesk.

2.18.4 OpenStack Configuration

Currently, the node of Crete is a High Availability node running on grizzly over Ubuntu 12.04 LTS and KVM as hypervisor created by the master node (ITBox v1.3.4.0). Nagios, CB and ODC were installed in the one of the two Controllers. A list of modifications on OpenStack configuration file has been made in order to use the security proxy of Spain instead of the keystone provided by default by OpenStack. By changing the local keystone to the keystone proxy, the OpenStack dashboard (horizon) is no more usable. The cloud portal is taking over the functionalities. Quota: The default quota has been configured as follows:

Property	Value
metadata_items	1024
injected_file_content_bytes	10240
ram	51200

Property	Value
floating_ips	3
key_pairs	100
instances	4
security_group_rules	20
injected_files	5
cores	6
fixed_ips	-1
injected_file_path_bytes	255
security_groups	10

Table 31: Default quotas Crete node

2.18.5 User-base

The node of Crete is a new XIFI node and currently is used by fi-lab users for testing reasons. On the other hand, we provide an independent Openstack infrastructure (Intellicloud) for Specific Enablers development and deployment for the FI-STAR project. This includes deployment of SEs by SE owners and testing and development of FI-STAR applications by Use Case owners. The two infrastructures are scheduled to be merged as long as the existing SEs migrate from Intellicloud to the new XIFI node. The utilization of the node is described in the following table.

List of the 10 over 26 first tenants Usage from 2015-02-04 to 2015-03-05:

Tenant ID	Instances	RAM MB-Hours	CPU-Hours	Disk GB-Hours
4980	16	7816207.92	3834.34	76211.30
1213	15	4323018.72	2593.06	39002.24
9270	12	1462859.72	714.29	14285.74
5921	7	1821956.69	1542.51	13440.00
102	6	1617537.68	1585.61	10490.94
2553	5	605197.94	295.51	5910.09
11257	5	785.07	1.53	0.00
8485	3	1032192.04	2016.00	0.00
8916	3	3440640.12	2688.00	26880.00
256	2	45676.24	89.21	0.00

Table 32: Tenants Crete node

2.19 Infotec Mexico (Associated Partner)

2.19.1 Description

The Mexican node was deployed using the Icehouse version of OpenStack and the system operating Ubuntu 14.04. This Node is physically located in the INFOTEC datacenter in the state of Aguascalientes, Mexico. This is a Tier III datacenter (Certification of the Uptime Institute) that

ensures an availability of 99,982% and a redundancy of N+1. The following equipment composes the current node:

- 2xServer HP ProLiant DL 380 CG6, processor Intel Xeon E5630 with 16 cores, 2.53 Ghz, 50 GB RAM, Volumen 300 GB.
- 3xServers HP ProLiant BL 460 C G6, processor Intel Xeon E5530 with 16 cores, 2.4 Ghz, 36 GB RAM, Volumen 146 GB.
- 1xServer HP ProLiant BL 460 C G6, processor Intel Xeon E5530 with 16 cores, 2.4 Ghz, 36 GB RAM, Volumen 300 GB.

An extension of infrastructure was designed in order to extend the capacity of the current node. This new infrastructure was designed to have two main purposes: one dedicated to a generic architecture of FIWARE and other, which is complementary to the general purpose architecture that was designed to support BigData applications. The new hardware, which will be added in the following weeks, is the following:

- 2xServers Relion 1900, Dual Intel Xeon E5-2630 v3, 8 Cores, 2.4GHz, 32 GB RAM DDR3, Volume 1 TB, NIC Intel X520-DA2 2x SFP+/10GiGE.
- 2xServers Relion 1900, Dual Intel Xeon E5-2630 v3, 8 Cores, 2.4GHz, 32 GB RAM DDR3, Volume 286 GB, NIC Intel X520-DA2 2x SFP+/10GiGE.
- 1xServer Relion 1903 GT, Dual Intel Xeon E5-2640 v3, 8 Cores, 2.6GHz, 64 GB RAM DDR3, Volume 1,2 TB, NIC Intel X520-DA2 2x SFP+/10GiGE.
- 15xServers Relion 2900, Dual Intel Xeon E5-2698 v3, 16 Cores, 2.3GHz, 512 GB RAM DDR3, Volume 7,2 TB, NIC Intel X520-DA2 2x SFP+/10GiGE.
- 1xServer Relion 900, Dual Intel Xeon E5-2603 v3, 4 Cores, 1.6 GHz, 32 GB RAM DDR3, Volume 500 GB, NIC Intel X520-DA2 2x SFP+/10GiGE.
- 2xServers Relion 1900, Dual Intel Xeon E5-2630 v3, 8 Cores, 2.4GHz, 64 GB RAM DDR3, Volume 286 GB, NIC Intel X520-DA2 2x SFP+/10GiGE.
- 1xServer Relion 1900, Dual Intel Xeon E5-2640 v3, 8 Cores, 2.4GHz, 64 GB RAM DDR3, Volume 7,2 TB, NIC Intel X520-DA2 2x SFP+/10GiGE.
- 5xServers Relion 2900, Dual Intel Xeon E5-2698 v3, 8 Cores, 2.6GHz, 64 GB RAM DDR3, Volume 37 TB, NIC Intel X520-DA2 2x SFP+/10GiGE.
- 1xIceBreaker Storage Fibre Channel 8Gbps host interface module, 4+4 ports, 20 x 2TB 7200rpm SAS drive, 200GB Solid State Drive.
- 3xSwitch Top of Rack (TOR), Artica 4804x, 10GiGE, +48 ports.
- 5xSwitch Uplink TOR, Artica 4804x, 10GiGE, +48 ports.
- 2xSwitch Administration, Artica 4804i, 1GiGE, +25 ports.
- 4xSwitch Uplink Administration, Artica 4804i, 1GiGE, +25 ports.

2.19.2 Experience on Support and Maintenance

During the period of creation of Mexican Node, lots of maintenance and support activities have to be done due to that the creation of node was a completely new activity for the INFOTEC technical group. The creation of the Mexican Node implied the complete configuration of some local equipment that was dedicated originally to commercial services. This equipment was dedicated to create the first instance of the Mexican Node. Below bullets summarise those activities:

- Test and configuration of network of high availability related to current INFOTEC equipment.
- Installation of the operating system Ubuntu 14.04 LTS
- Configurations of the required network and storage for the installation of OpenStack (Icehouse) in the servers.

- Configuration of the core switch for the interconnection of the servers, by resulting in the activation of VLAN's, data network and flat network for the administration of OpenStack and also the admin network for the equipment management.
- Installation of OpenStack (Icehouse) by following the steps described in <http://docs.openstack.org/icehouse/install-guide/install/apt/content/>
- Creation of the system logs in the DNS servers to sign in the project: <http://filab.infotec.net.mx>
- Configuration of the network and security to enable the access, via SSH, to the Mexican nodes from Spain.
- Validation of the nodes at OpenStack level.
- Activities related to addressing the authentication to the main node of FILAN in Spain.
- Assignment of the types of fees to users of the Mexican node.
- System Operations: Upgrade and maintenance.
- Security: Checking log security, and verifying the security problems and notifications of
- Configuration files to adapt to changes in the architecture.
- FIWARE Lab Migration: Migration of the whole Mexican node from previous physical hosts in Mexico City DC to the INFOTEC DataCenter located in the State of Aguascalientes.

2.19.3 Current Status

The current status of the Mexican Node is the following:

- 1 Controller: 2xServer HP ProLiant DL 380 CG6, processor Intel Xeon E5630 with 16 cores, 2.53 Ghz, 50 GB RAM, Volumen 300 GB.
- 3 Compute Nodes: HP ProLiant BL 460 C G6, processor Intel Xeon E5530 with 16 cores, 2.4 Ghz, 36 GB RAM, Volumen 146 GB.
- 1 Neutron: HP ProLiant BL 460 C G6, processor Intel Xeon E5530 with 16 cores, 2.4 Ghz, 36 GB RAM, Volumen 300 GB.
- 1 Storage: 2xServer HP ProLiant DL 380 CG6, processor Intel Xeon E5630 with 16 cores, 2.53 Ghz, 50 GB RAM, Volumen 300 GB.

Following, the extension of infrastructure that was designed to extend the capacity of the current node is presented. As commented, this new infrastructure is composed by a general purpose architecture and an architecture designed to support BigData applications.

The generic FIWARE infrastructure	
Quantity	Description
2	Relion 1900 Server-Controller: Dual Intel Xeon E5-2630 v3, 8Cores, 2.4GHz, 32 GB RAM DDR3, RAID I Volume 1 TB, NIC Intel X520-DA2 2x SFP+/10GiGE.
2	Relion 1900 Server- Neutron: Dual Intel Xeon E5-2630 v3, 8Cores, 2.4GHz, 32 GB RAM DDR3, RAID I Volume 286GB, NIC Intel X520-DA2 2x SFP+/10GiGE.
15	Relion 2900 Server-Compute: Dual Intel Xeon E5-2698 v3, 16C, 2.3GHz, 512GB DDR3-1600 ECC (32 x 16GB), RAID 5 Volume: 7,2 TB (6 x 1.2TB), NIC Intel X520-DA2 2x SFP+/10GigE.
1	Relion 900 Server-Monitored: Dual Intel Xeon E5-2603 v3, 4Cores,1.6GhZ,32 GB RAM DDR3, Volume 500GB SATA2, NIC Intel X520-DA2 2x SFP+/10GigE.
2	Relion 1900 Server-Storage: Dual Intel Xeon E5-2630 v3 8Cores 2.4GHz, 64 GB RAM

The generic FIWARE infrastructure	
Quantity	Description
	DDR3, RAID I Volume 286GB, NIC Intel X520-DA2 2x SFP+/10GiGE.
1	IceBreaker powered by EMC-Storage, Fibre Channel 8Gbps host interface module, 4+4 ports, 20 x 2TB 7200rpm SAS drive, 200GB Solid State Drive.
3	Switch Top of Rack (TOR), Artica 4804x, 10GiGE, +48 ports.
4	Switch Uplink TOR, Artica 4804x, 10GiGE, +48 ports.
2	Switch Administration, Artica 4804i, 1GiGE, +25 ports.
4	Switch Uplink Administration, Artica 4804i, 1GiGE, +25 ports.
1	Console Server, Avocent ACS6048, 48x RJ45, w/ dual AC PSU
1	Rackmount 17" LCD/ US keyboard / USB TochPad, 1U, Avocent

Table 33: Server equipment Mexican node

The BigData-specific FIWARE infrastructure	
Quantity	Description
1	Relion 1903 GT Server-Visualization node: Dual Intel Xeon E5-2640 v3, 8C, 2.6 GHz, 1866 MHz, 64 GB DDR3, RAID I Volume 1,2TB, NIC Intel X520-DA2 2x SFP+/10GiGE.
1	Relion 1900 Server- Name & Mgmt Node: Dual Intel Xeon E5-2460 v3, 64GB DDR3, RAID I Volume 7,2TB, NIC Intel X520-DA2 2x SFP+/10GiGE.
5	Relion 2900 Server - Worker Node: Dual Intel Xeon E5-2640 v3, 8C, 2.6 GHz, 1866 MHz, 64GB DDR3, RAID I Volume 37TB, NIC Intel X520-DA2 2x SFP+/10GiGE.
1	Switch Administration: Artica 4804x - 10Gige Compute Traffic. Penguin Artica 4804x Chassi, 2x 460w Power Supplies, PSU to faceplace airflow.
1	Rackmount 17" LCD/ US keyboard / USB TochPad, 1U, Avocent
1	Console Server, Avocent AC"6048, 48x, RJ45, w/dual AC PSU, w/modem

Table 34: Big-data-specific Server equipment Mexican node

The synthesis of the new capabilities to be added to current equipment is the following:

- Total cores: 704
- Total RAM (TB): 8,16875
- Total HD (TB): 352,85

Finally, in order to carry out the configurations of the infrastructure a set of OpenStack flavours was configured. We have configured 5 flavours in the infrastructure, each flavour have defined different sizes of RAM, disk storage space and number of processing cores. The configurations are showed in the next table.

OpenStack flavour in Mexico				
ID	Name	VCPUs	Memory(MB)	Disk(GB)
1	m1.tiny	1	512	1
2	m1.small	1	2048	20
3	m1.medium	2	4096	40
4	m1.large	4	8192	80
5	m1.xlarge	8	16384	160

Table 35: OpenStack configuration Mexican node

Operating systems: FI-WARE Lab Mexico was configured with 32 images of different operating systems based on Linux, only 2 new images were added to the FIWARE Lab image distribution. The OS are free versions and ready to be installed in the virtual servers. The configurations are presented in the next table.

New images in Mexico Mexican node of usage in the Spain node					
Image	Type	Status	Enable	Container format	Disk format
Trusty	~	active	public	BARE	QCOW2
cirros-0.3.2-x86_64	~	active	public	BARE	QCOW2

Table 36: New images configurations Mexican node

Currently, access to the node FI-WARE Lab Mexico's LaNIF can be made through the central portal of FI-LAB: <http://lab.fi-ware.org/>.

2.19.4 OpenStack Configuration

In order to install the final infrastructure of OpenStack is realized a test concept in the Data Center. In order to carried out the test was necessary to solicited an STAAI infrastructure that is composed for 5 servers sunfireX and 14 IBM Blade Center HS22.

In the phase were carried out a set of test and high availability configuration network in the IBM Blade Center HS22. Then, It was realized the installation of operating system Ubuntu 12.04 LTS. It was realized the network configuration to install the OpenStack (Havana) in 19 servers.

A level of network infrastructure and routing configurations were activated 3 VLAN's, DATA NETWORK and FLAT NETWORK to achieve the communication of Openstack. The installation of OpenStack is achieved following the steps of the "Install guide" <http://docs.openstack.org/havana/install-guide/install/apt/content/>

For the Test Laboratory, were configuring the DNS registers the link is: <http://filabsf.infotec.net.mx/horizon>

The equipment employed to install OpenStack is described in the next table.

Equipment used in Mexican node of usage in the Spain node					
ID	Component	Servers	Port	Role	Characteristics
1	Controller	1	2 Network cards	MySQL Rabbit Keystone Glance Dashboard	HP Proliant DL 380 C G6 Processor Intel Xeon E5630 16 cores 2.56 GHz 300 GB Hard Disk 50 RAM
2	Compute	3	2 Network cards	Generate the virtual machines	HP Proliant BL 460 C G6 Processor Intel Xeon E5530 16 cores 2.40 GHz 146 GB Hard Disk 36 RAM
3	Neutron	4	2 Network cards	Provide the address and the output to the Network to the virtual machines	HP Proliant DL 460 C G6 Processor Intel Xeon E5530 16 cores 2.40 GHz 300 Gb Hard Disk 36 RAM
4	Storage	1	3 HBA cards 2 Network cards	Provide the storage of the virtual machines.	HP Proliant DL 380 C G6 Processor Intel Xeon E5630 16 cores 2.53 GHz 300 Gb Hard Disk 50 RAM

Table 37: OpenStack equipment Mexican node

2.20 University of Messina – IT (Associated Partner)

2.20.1 Description

The node of UniMe is composed of 14 IBM Blades (LS21 class) with a total 52 Cores, 112GB of RAM and ~1.1 TB of Disks.

2 OpenFlow-enabled Cisco 3850 switches have been procured to connect the servers based on a 10G (fiber) links, as well as a 10G NIC-populated Intel WildCat server running Vyatta as full featured soft router, in order to replace current 1G ethernet-class network equipment.

UniMe relies on a local NREN (GARR) PoP for Internet (and in the future, MDVPN) connectivity, as well as Public IP Allocation.

The node has at the moment a /24 Public external network allocated, which is going to be subnetted (probably either a /26 or a /27 at the beginning, to be expanded later) for XI-FI-related Public IP availability pool.

2.20.2 Current Status

All administrative tasks have been completed, the hardware procured, and the servers already installed on-site.

Setup of the node for OpenStack is queued, pending the prerequisite connectivity enablement, through a third-party site-hosted VPN (either Lannion or Red.es, yet to be confirmed), in order to avoid to keep waiting for the long activation time for MD-VPN services by the national NREN.

2.20.3 OpenStack Configuration

The UniMe node is expected to be deployed as an IceHouse instance, running over an officially supported distro (to be decided) and KVM as hypervisor. The node will be deployed using the latest version of ITBox.

All other details subject to deployment experience and other constraints, thus to be planned.

2.20.4 User-base

The UniMe Node is going to provide resources for multiple projects coming from different areas:

- (to be planned) H2020 Projects related to FI-WARE, where UniMe (or an institutional partner) is involved
- Projects related to FI-WARE Accelerators (at least FrontierCities is planned)

2.21 Wroclaw University of Technology – Poland (Associated Partner)

At the time of writing this document, this associated partner has been just formally accepted by the Office to join the federation.

Update procedures related to the Management of the node has been completed in section 3 and technical activities are ongoing.

2.22 Report on Assistance and Support

This section gives a short report on what the main type of questions and support requests have been, and what issues have been encountered. For a detailed report on the organisation of FI-Developer support and a respective analysis and statistics please see section 6.

The questions received from FI-Developers can be grouped in below categories:

- Access problems to portal: The user can't access the portals for some reason:
 - The problem is usually a down time for any reason, a bug or the system was overloaded at that time.
 - The disk was full and/or some services were down.
- The portals don't work as expected.
 - No more floating IPs.
 - The users have reached their quotas.
 - No more free resources in the compute nodes.
 - Instability of Essex and the problems that causes it.
 - Problems with volumes.
 - Problems with Storage (swift).
- Virtual machine access problems.
 - The network is down for any reasons (fibre cut, blackouts, ...).
 - The network was down and DHCP leases in Virtual hosts were lost.
 - Incorrect configuration of .pem files (usually permissions...).
 - The Virtual host didn't boot and stayed in "grub" menu.
 - Errors in the Image files which does not allow booting instances.

- The network is slow.
 - User abusing the service (e.g. using a film streaming proxy in a VM).
 - Virtual host compromised with installation of trojans which sends lots of traffic.
 - Security breach behaviour has been detected from the IP addresses pool you are responsible for (botnet).
- Requirements from users
 - Increasing Quotas (floating IPs, disk, number of instances, etc...).
 - Names in DNS.
- Questions about various Generic Enablers
 - Usually redirected to the StackOverflow (tags created at the moment in StackOverflow are: fiware, fiware-orion, fiware-wirecloud, filab).
 - Rest of GEi redirected to the owner.
 - Occasional requests for pointers to the GE documentation
 - Enquiries to know the location of the source code of the GEs
- Questions about Linux administration.
 - Using the disks (ephemeral or volumes).
 - Changing partition tables.
- Unfamiliarity with the services provided.
 - Questions about images, security groups, organizations, keypairs, etc...
- Generic feedback.

3 UPDATE ON GENERAL PROCEDURES

In this section we define a set of general procedures that are required to execute the procedures for maintenance and user / developer support defined further down in this document. These procedures are related to the identification and assignment of operational roles for:

- Management of nodes
- Developer support;
- Infrastructure support.

3.1 Management of Nodes

Each node has provided a reference person for each of the following roles:

- Node Manager: the main contact for the node and the person in charge of decision on how to apply XIFI policies and procedures in the node.
- Authoritative Contact: This is the individual who either has to request ALL new connections to a particular VPN or has to approve all new connection requests to a particular VPN that could come in through various NRENs².
- System Administrator: the person in charge for the physical set-up of servers, the installation of server management software and its configuration.
- Network Admin: the person in charge for the physical set-up of network (internal and external access), the installation of network management software and its configuration.
- Node Help Desk: the person in charge for the support of user requests specific to a node.

For each reference person the following information has been provided:

- Full name
- Email contact
- Phone contact
- Availability

The full information is stored on the secure part of the XIFI Wiki [11]: [Fi-ppp:Management_of_XIFI_Nodes](#)

A copy of the data is given in below tables. For privacy reasons however (as D5.3 is a public Deliverable) email and phone contact details were not included in below tables.

3.1.1 Berlin

The Berlin node consist of two distinct sites contributing distinct services: The Fraunhofer site adds datacentre capacities while the DT site adds wireless infrastructure capacities. Efforts have been made to hide that functional separation and to provide Berlin node services as an integrated offer. Nevertheless, the functional separation is creating the need to maintain distinct help desk contact points providing complementary support services for datacentre and wireless access issues.

² Essentially an NREN may receive a request from one of their customers to be connected into an existing VPN. However, the NREN in question does not know whether that request has actually been sanctioned by the rest of the users of that particular VPN instance (who can be in countries served by other NRENs). This is the role of a per-VPN-instance “authoritative contact”.

Role	Contact	Availability
<i>Fraunhofer Site</i>		
Node Manager	Bernd Bochow	weekdays, 09:00 - 18:00 CET
Authoritative contact	Bernd Bochow	weekdays, 09:00 - 18:00 CET
System and Network Administrator	Support team	weekdays, 09:00 - 18:00 CET
<i>DT Site</i>		
Node Manager		
Authoritative contact	Matthias Baumgart	weekdays, 09:00 - 18:00 CET
System and Network Administrator	Nico Bayer	weekdays, 09:00 - 18:00 CET
<i>Both sites</i>		
Node Help Desk	Support team	weekdays, 09:00 - 18:00 CET

Table 38: Berlin contact details

3.1.2 Brittany

Role	Contact	Availability
Node Manager	Sergio Morant	9h-18h CET
Authoritative contact	Sergio Morant	9h-18h CET
System and Network Administrator	Engineering team	9h-12h30/13h30-18h CET
Node Help Desk	Support Helpdesk	9h-12h30/13h30-18h CET

Table 39: Brittany contact details

3.1.3 Spain Node

Role	Contact	Availability
Node Manager	Antonio Fuentes Bermejo (Red.es), Fernando López (TID)	weekdays, 08:00 - 20:00 CET
Authoritative contact	Antonio Fuentes Bermejo (Red.es), Fernando López (TID)	weekdays, 08:00 - 20:00 CET
System and Network Administrator	Enrique de Andres (Red.es)/Francisco José Martín (Red.es)/José Ignacio Carretero (TID)	weekdays, 08:00 - 18:00 CET

Role	Contact	Availability
Node Help Desk	Enrique de Andres (Red.es)/Francisco José Martín (Red.es)/José Ignacio Carretero (TID)	weekdays, 08:00 - 18:00 CET

Table 40: Spain contact details

3.1.4 Trento

Role	Contact	Availability
Node Manager	Ivan Biasi	Mon-Fri 9:00 – 17:00 CET
Authoritative contact	Ivan Biasi	Mon-Fri 9:00 – 17:00 CET
System and Network Administrator	Trento Node Team	Mon-Fri 9:00 – 17:00 CET
Node Help Desk	Trento Node Team	Mon-Fri 9:00 – 17:00 CET

Table 41: Trento contact details

3.1.5 Waterford

Role	Contact	Availability
Node Manager	Eamonn Power	10:00 - 18:00 CET, Mon - Fri
Authoritative contact	Eamonn Power	10:00 - 18:00 CET, Mon - Fri
System and Network Administrator	Joe Tynan	10:00 - 18:00 CET, Mon - Fri
Node Help Desk	Joe Tynan	10:00 - 18:00 CET, Mon - Fri

Table 42: Waterford contact details

3.1.6 IMINDS

Role	Contact	Availability
Node Manager	ThijsWalcarius	Mon-Fri 9h-17h CET
Authoritative contact	ThijsWalcarius	Mon-Fri 9h-17h CET
System and Network Administrator	Vicent Borja Torres	Mon-Fri 9h-17h CET
Node Help Desk	iMinds XIFI Node Help Desk	Mon-Fri 9h-17h CET

Table 43: IMINDS contact details

3.1.7 ZHAW

Role	Contact	Availability
Node Manager	Seán Murphy	Mon-Fri 8:00-18:00 CET
Authoritative contact	Thomas Michael Bohnert	Mon-Fri 8:00-18:00 CET
System and Network Administrator	Seán Murphy	Mon-Fri 8:00-18:00 CET
Node Help Desk	ICCLabXIFIsupprt	Mon-Fri 8:00-18:00 CET

Table 44: ZHAW contact details

3.1.8 PSNC

Role	Contact	Availability
Node Manager	Wojbor Bogacki	Mon-Fri 9:00-17:00 CET
Authoritative contact	Bartosz Belter	Mon-Fri 9:00-17:00 CET
System and Network Administrator	Marek Zawadzki	Mon-Fri 9:00-17:00 CET
Node Help Desk	Local XIFI Team	Mon-Fri 9:00-17:00 CET

Table 45: PSNC contact details

3.1.9 Neuropublic

Role	Contact	Availability
Node Manager	John Koufoudakis	Mon-Fri 08:00-16:00 CET
Authoritative contact	John Mavroudis	Mon-Fri 08:00-16:00 CET
System and Network Administrator	Theofanis Katsiaounis	Mon-Fri 08:00-16:00 CET
Node Help Desk	XIFI Support Team	Mon-Fri 08:00-16:00 CET

Table 46: Neuropublic contact details

3.1.10 CESNET

Role	Contact	Availability
Node Manager	Rudolf Vohnout	Mon-Fri 10:00-16:00 CET
Authoritative contact	Jan Gruntorad	Mon-Fri 09:00-15:00 CET

Role	Contact	Availability
System and Network Administrator	Jan Kunderat	24/7
Node Help Desk	Local XIFI Support Team	24/7

Table 47: CESNET contact details

3.1.11 UPRC

Role	Contact	Availability
Node Manager	Aristi Galani	Mon-Fri, 7h-14h CET
Authoritative contact	Support team	Mon-Fri, 9h-17h CET
System and Network Administrator	Support team	Mon-Fri, 9h-17h CET
Node Help Desk	Support team	Mon-Fri, 9h-17h CET

Table 48: UPRC contact details

3.1.12 Com4Innov

Role	Contact	Availability
Node Manager	Philippe Badia	Mon-Fri 9:00-18:00 CET
Authoritative contact	Claude Hary	Mon-Fri 9:00-18:00 CET
System and Network Administrator	Bastien Putegnat	Mon-Fri 9:00-18:00 CET
Node Help Desk	Support Team	Mon-Fri 9:00-18:00 CET

Table 49: Com4Innov contact details

3.1.13 ACREO Swedish ICT

Role	Contact	Availability
Node Manager	Jonas Lindqvist	Mon-Fri 9:00-17:00 CET
Authoritative contact	Anders Berntson	Mon-Fri 9:00-17:00 CET
System and Network Administrator	Roland Elverljung	Mon-Fri 10:00-18:00 CET
Node Help Desk	Switch board	Mon-Fri 9:00-17:00 CET

Table 50: ACREO contact details

3.1.14 WIGNER

Role	Contact	Availability
Node Manager	Sandor Laki	Mon-Fri 10-18 CET
Authoritative contact	Support team	Mon-Fri 10-18 CET
System and Network Administrator	Support team	Mon-Fri 10-18 CET
Node Help Desk	Support team	Mon-Fri 10-18 CET

Table 51: WIGNER contact details

3.1.15 UTH

Role	Contact	Availability
Node Manager	Ioannis Igoumenos	Mon-Fri, 09h-17h CET
Authoritative contact	NITLab team	Mon-Fri, 09h-17h CET
System and Network Administrator	NITLab team	Mon-Fri, 09h-17h CET
Node Help Desk	NITLab team	Mon-Fri, 09h-17h CET

Table 52: UTH contact details

3.1.16 BTH

Role	Contact	Availability
Node Manager	Kurt Tutschku	Mon-Fri, 09h-17h CET
Authoritative contact	Eva-Lisa Ahnström	Mon-Fri, 09h-17h CET
System and Network Administrator	Patrik Arlos	Mon-Fri, 09h-17h CET
Node Help Desk	Switch Board	Mon-Fri, 09h-17h CET

Table 53: BTH contact details

3.1.17 Intellicould (Crete) – Associated Partner

Role	Contact	Availability
Node Manager	Spyros Argyropoulos	Mon-Fri, 09h-17h,CET
Authoritative contact	Support Team	Mon-Fri, 09h-17h,CET
System and Network	Support Team	Mon-Fri, 09h-17h,CET

Role	Contact	Availability
Administrator		
Node Help Desk	Support Team	Mon-Fri, 09h-17h,CET

Table 54: Intellicloud contact details

3.1.18 Infotec (Mexico) – Associated Partner

Role	Contact	Availability
Node Manager	Hugo Estrada Esquivel	Mon-Fri, 09h-17h, CST
Authoritative contact	Gabriela Diaz Ocampo	Mon-Fri, 09h-17h, CST
System and Network Administrator	Sergio Martínez Pacheco Leonel Reyes Rosales	Mon-Fri, 09h-17h, CST
Node Help Desk	INFOTEC Node Team	Mon-Fri, 09h-17h, CST

Table 55: Infotec contact details

3.1.19 University of Messina (Italy) – Associated Partner

Role	Contact	Availability
Node Manager	Giovanni Merlino	Mon-Fri, 09h-17h,CET
Authoritative contact	Giovanni Merlino	Mon-Fri, 09h-17h,CET
System and Network Administrator	Antonio Puliafito	Mon-Fri, 09h-17h,CET
Node Help Desk	Switch Board	Mon-Fri, 09h-17h,CET

Table 56: University of Messina contact details

3.1.20 Wroclaw University of Technology (Poland) – Associated Partner

Role	Contact	Availability
Node Manager	Pawel Swiatek	Mon-Fri, 09h-17h,CET
Authoritative contact	Pawel Swiatek	Mon-Fri, 09h-17h,CET

Table 57: Wroclaw University of Technology contact details

3.2 Developer Support

Different roles to run the developer support are required and have been defined. These roles participate in the developer support process which is defined in detail in section 6 of this Deliverable. The following roles were defined:

- Level 1 Help Desk: the team in charge to filter tickets incoming to the shared facility.

- **Node Help Desk:** the persons in charge for the support of user requests specific to a node (a XIFI user is a developer). These persons provide Level 2 support for the node. Level 3 support is provided by system or network administrators if need be, in case of more complex issues.
- **Software Component Support:** the person in charge of providing the support for a specific GE.

Each person assigned to above roles should provide:

- Full name
- Email contact
- Register in the shared facility that is used as Level 1 support tool (persons will be assigned to an area according to their role)

For the definition of the Level 1 Helpdesk team please see section 6.4.

Level 1 helpdesk will be in charge of the following activities:

- Initial contact point for all incoming tickets that are not directly assigned to a node, FIWARE Ops tool or GE.
- Providing support to general issues that can be easily solved by pointing out to FAQ, stack overflow groups or other documentation.
- Contribute to the creation/update of FAQ by handling generic requests by users (e.g. Why I cannot reach the following port on my VM? Answer: You need to set the correct security group).
- Forwarding requests that cannot be answered by Level 1 to the proper Level 2 team.
- Routing general requests - not of technical nature or not related to FIWARE Lab - to the proper contact point.

Node Helpdesk

The node helpdesk is in charge of handling developer requests that are specific to a XIFI node; i.e. this team provides Level 2 support for the node. The node helpdesk consists of representatives from all nodes that have joined the federation. The contact persons are those indicated in section 3.1 "Management of Nodes" in the tables listed as "Node Help Desk".

In case of infrastructure issues that cannot be solved by the level 2 helpdesk, requests will be forwarded by the level 2 helpdesk to the level 3 support, provided by system and network administrators, as well as the responsible developers for each respective GE.

Software Component Support This support role is in charge of providing the support for a specific GE that has been developed by FI-Ware project partners. Support is provided by the GE owner, i.e. the respective FI-Ware / FI-Core partner, and is thus out of the scope of XIFI. Incoming tickets will be routed directly to the GE owner whenever possible, or forwarded by Level 1 / 2 helpdesk in case an automated assignment to Level 3 was not made.

3.3 Infrastructure Support

The infrastructure support is based on a joint facility (JIRA, to be introduced further down in the document) that is shared with developer support. Different roles to run the infrastructure support are required and have been defined and assigned:

- **Level 1 Help Desk:** the person in charge to filter tickets incoming to the shared facility.
- **Federation Manager:** the person in charge of the federation office and of the process of including new nodes, as well as the process of withdrawing nodes.

- Federation Deployment Help Desk: the person in charge of providing the support for node deployment.
- Software Component Support: the person in charge of providing the support for a specific XIFI federation tool (FIWARE Ops)

All persons assigned to above roles should provide:

- Full Name
- Email contact
- Register in the shared facility that act as Level 1 support tool (persons will be assigned to an area according to their role)

Infrastructure support team

Role	Name	Organisation
Level 1 Help Desk	Florian Rommel	EURES
Federation Manager	Anastasius Gavras	EURES
Federation Deployment Help Desk	Daniele Giai Pron	Telecom Italia

Table 58: Infrastructure support team

It should be noted that the Level 1 helpdesk for infrastructure support, defined here, is provided by a different team than the Level 1 helpdesk for developer support, as detailed in section 6.

Software Component Support

Software Component Support is provided by the persons in charge of the respective Software Components of the XIFI federation tool suite (FIWARE Ops). The list of software components is available on the public XIFI Wiki: http://wiki.fi-xifi.eu/Public:Software_Components. For each component the responsible person is listed. These are in charge of providing level 2/3 support for the components.

4 UPDATES ON PROCEDURES FOR OPERATING THE FEDERATION

4.1 Stakeholders and Roles in Establishing and Maintaining Operational Level Agreements

Deliverable D2.2 has defined basic roles and stakeholders in the scope of federation. These definitions are in line with the overall XIFI stakeholder definitions [17]. This list must now be revisited under the scope of operational level agreements, node and federation operations (cf. sect. 6.2) and node and federation maintenance (cf. sect. 5.2).

D2.2 defined the **Federator** (also known as federation manager) as a role with full access to the federated XIFI infrastructure in charge of the control and the management of the federation. In scope of the operation level agreements (OLA) **the Federation authority** complements this technical role by a legal representative of the federation in the internal relationship between infrastructures. Federator and Federation Authority are jointly complementing the role of an infrastructure owner since this role does not exist for the federation (that is, something like a 'federation owner' does not exist). It should be noted here, that the Federator role usually falls onto an individual or entity while the Federation Authority usually denotes an office. The incumbent in turn may be an individual or entity.

D2.2 defined the **Infrastructure Owners** (also known as **Node Providers** or, acting in the scope of OLA as an **Infrastructure Management Authority**) as a role that takes responsibility to expose the offerings of an infrastructure node including GEs to the federation and for providing and enforcing policies regarding the utilization of their resources and components. **Infrastructure Operators** are particular roles that work with Infrastructure Owners to keep the federated infrastructure working and available. In general, each Infrastructure Owner steps into a bilateral relationship with the Federation Authority in the scope of implementing operational level agreements. That is, a bilateral agreement is established that defines the obligations of both. This is usually done implicitly when joining the federation since terms and conditions of the federation have to be agreed in the course of this process. Although out of scope for the time being, it must be noted that Infrastructure Owners may also enter into bilateral agreements if needed. This is likely the case when particular national laws apply, or particular services are brought into a bilateral relationship.

In addition to the roles discussed above and in the scope of D2.2, we here additionally need to consider the **Network Provider** role complementing that of the **Node Provider** in the federation. In general Node Provider and Network Provider already are in a bilateral agreement that affects the operational level agreement between Infrastructure Owner and Federator: Infrastructure Owners have to respect 'their' agreement with their Network Provider prior to agreeing with the Federator or other Infrastructure Owners. Furthermore, there might be multiple network providers that federate to provide network connectivity for the federation as a whole, which is the case for the XIFI MD-VPN. Such 'network federation' may demand for a distinct **Network Federator** role. In case of the XIFI MD-VPN, GEANTs role comes closest to that of a Network Federator. There may or may not exist bilateral agreements between the Federator and the Network Federator if needed.

In the following, collaboration between Federator, Infrastructure Owners, Network Providers and (optionally) Network Federator is assumed for the federation being able to provide SLAs in its interaction with the User/Developer. Operational Level agreements between these roles and stakeholders are considered to be the means that enable sufficient trust to allow sharing and delegating infrastructure's responsibilities to the federation. Procedures and workflows to implement OLAs in the federation as well as implementation status are described in subsequent sections.

4.2 Update on Support Process and Procedures for Joining the Federation

The Federation Office, as defined in [10], is the first point of contact for the nodes to join the federation and is in charge of administrative matters. After administrative acceptance for joining the

XIFI infrastructure federation, a new node that is joining the XIFI federation is technically supported by XIFI with the methodological support level defined in [5] and described in detail in Deliverable D5.6 [8]

In Figure 17 the support process is shown for a new node that is having an issue. Also the role of the XIFI support (Fi-Ops Level 2 Helpdesk) is defined in the figure.

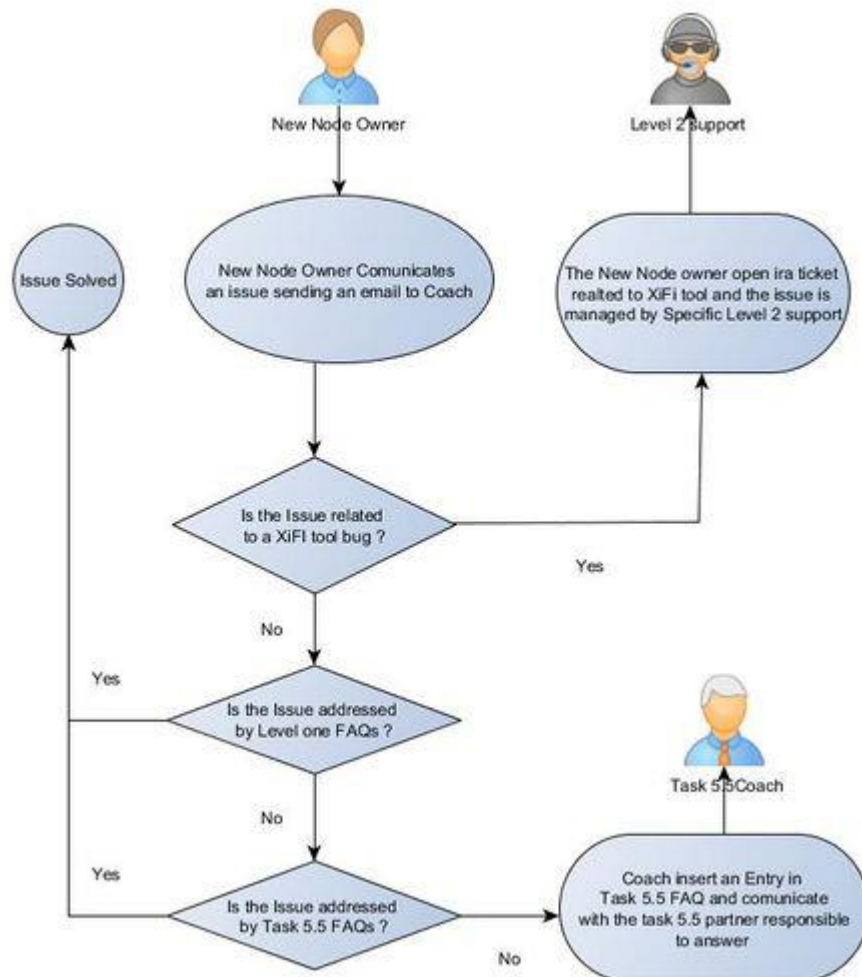


Figure 17: Federation support procedures

The engage of the Level 3 of FIWARE Ops is managed inside Level 2.

4.3 Scope and Purpose of Operational Level Agreements

Infrastructures provide resources to the federation to enable the federation to commit on service level agreements (SLAs) between the federation and its users. A number of factors, implicit and explicit, contribute to the implementation of operational agreements.

- Implicit agreements, for example, are put in place by an infrastructure when joining the federation through fulfilling the minimum requirements set forth in D5.1 (cf. XIFI:Wp5:d51#Requirements).
- Explicit agreements, for example, are put in place by an infrastructure agreeing to participate in the federation help-desk, implementing node support to the user (cf. XIFI:Wp5:d53#Support_to_FI-Developers_.E2.86.92_TI) and implementing maintenance

procedures for the node to maintain the service level experienced as initially agreed upon (cf.

XIFI:Wp5:d53#Maintenance_process_.28updated:_15.07.2014.29_.E2.86.92_Fraunhofer)

Operational level agreements are considered two-sided and mostly apply to the relationship between federation and node. An example for a two-sided implicit agreement is identity management. Here the node agrees implicitly on the delegation of part of its user management to the federation authority by implementing the federation identity management sub-system as required, while the federation authority commits to make the process of user relationship management transparent and revisable. This example makes clear that trust between stakeholders is essential for implementing operational level agreements. Furthermore, measurable and quantifiable key performance indicators are required for a reliable implementation, which is agreed upon through implementing the federated monitoring sub-system.

Thus, upon joining the federation a new node steps into an operational level agreement by accepting the terms and conditions set in place by the federation authority equally for all infrastructure nodes (with distinct parameters for prospective master nodes). Summarizing D5.1, these parameters are formulated in terms of minimum requirements regarding network bandwidth, computing resources, storage resources, availability targets, and configuration capacities (e.g. though PaaS requirements). No commitments are currently made regarding 'non-conventional resources'.

The main purpose of operational level agreements is, as mentioned earlier, to enable user-side SLAs, which here is broken down into several sub-targets that may be considered as a categorization of agreement purposes. The list given in the following is clearly non-exhaustive and contentiously discussed regarding the technical targets and metrics as well as their legal and technical framework requirements.

Purpose of an OLA	Stakeholders involved in a mutual agreement	Prerequisites for agreeing in an OLA
Maintaining network connectivity	Federator and IO	Agreements between Network Provider and IO
Maintaining computing and storage resources	Federator and IO	None, if under control of IO
Maintaining availability of 'non-conventional resources'	User and IO	Agreements between IO and National or Local Authorities / Third Parties (e.g. in case of spectrum licenses)
Maintaining infrastructure availability targets	Federator and IO	Agreements between IO and Third Parties (e.g. local facility manager)
Maintaining user support	User and IO	Agreement between Federation Authority and IO to provide user support
Maintaining mutual Infrastructure support	Federator and IO(s)	Agreements between Federator and IOs regarding the implementation of a maintenance process
Maintaining maintenance contact points	Federator and IO(s)	Agreements on help-desk availability (i.e. availability of supporters)
Maintaining software maintenance	Federator and IO and between IOs	Agreements on provisioning and availability of a shared software repository

Purpose of an OLA	Stakeholders involved in a mutual agreement	Prerequisites for agreeing in an OLA
Maintaining communication security	Federator and IO, between IOs, between IO and Network Provider, and between Federator and Network Federator	Mutual agreements regarding the implementation of secure protocols and secure credential exchange, potentially including the operation of a certification authority
Maintaining user privacy	Federator and IO	Agreements on the quality assurance process (e.g. regarding identity management)
Maintaining tenant isolation	Federator and IO, and between IOs	Agreements on federation management security and implementation of Infrastructure's commitment to implement tenant isolation

Table 59: OLA categories

4.4 Operational Level Agreements

When joining the federation a node implicitly agrees on – or already has implemented as a prerequisite for joining – a number of rules, for example, to install conformant cloud management, monitoring and access control services. Hence, the joining node enters into a set of operational level agreements (OLAs) between node and federation in an early state of federation when agreeing to implement common operations and maintenance procedures set forth by the federation.

By disclosing node monitoring data to the federation and to the federation users the node enables tracking and evaluation of its performance data as a prerequisite for validating the node's conformance with these rules. Since most of these parameters are quantifiable, they can form the basis for evaluating a node's conformance level as its potential to implement and maintain service level agreements (SLAs) towards the user.

A number of OLAs that do not apply to any direct quantitative evaluation are complementing the set of performance parameters and are equally important for a viable federation. For example, nodes agree to join the help-desk and implement procedures for mutual collaboration through the help-desk. Utilising the help-desk then becomes a subjective OLAs that cannot be measured directly but may also receive quantifiable judgement from an observer that has to implement evaluation rules to map a subjective OLA to one or more performance indicators, such as the number of tickets received or resolved through the help-desk. This approach is similar to a symbolic reasoning process and can be automated.

The implementation of OLAs in the XIFI federation is outlined in subsequent sections based on the node maintenance status at the time of writing.

In a parallel approach it was considered already to base the definition of OLAs on workflows, which would allow to split complex scenarios involving many OLA stakeholders and parameters into smaller processes with limited scope and better manageability. A workflow-based approach would at the same time decrease complexity, increase scalability and increase security. The latter is of particular interest since when a workflow is defined it is only applicable within the boundaries set through rules, policies, actions, stakeholders, and observable and controllable parameters bound with this workflow. In consequence, the workflow could be authenticated, authorised and certified as an operational unit thus.

Although some of the maintenance tasks developed in scope of the Work Package 5.4 "Node Maintenance" have been designed relying on a business process model, which can be mapped easily to a workflow, workflow procedures have not been implemented so far in a formalised way. Hence, their basic principles are discussed further in the annex. The findings and procedures suggested in this discussion may come handy later on when the XIFI federation gains maturity and the number of nodes

increased that much causing the current help-desk based methods to become tedious. The workflow approach thus is considered as a contribution on sustainability making it possible to automate, to evaluate and to secure basic tasks in the node's operations and maintenance including the implementation and management of OLAs.

4.5 OLA implementation in XIFI

4.5.1 OLA Purpose

The goal is to explain the design and specification of Operational Level Agreement for a federated cloud infrastructure.

In short it is proposed to implement federation OLA as a set of uniform federation-wide policies (e.g. based on the federation utility preservation and increase) and their mapping to service-specific policies for all services offered by a federation.

A federated cloud infrastructure is an attractive option for multiple providers to join their resources and to appear to their customers as a (federated) provider of virtually unlimited capacities.

For a sustainable federation FitSM recommends [24] to care about offered services in the following way:

1. each service is to be described by a service portfolio entry (a template with the three sub-records);
2. each service is to be protected by an SLA (assuming that even best-effort service offering must have an SLA that specifies e.g. planned service outages and service maintenance schedule);
3. service evolution proceeds in the direction of enhancing service capabilities (increasing maturity levels) reflected by the corresponding modifications of service portfolio records.

This way, at any moment of time the actual service offering by a federation is reflected by a set of valid records in the service portfolio.

The usage of these services is not limited to that of a single service entity, the trend is service chaining (composition); notably Microsoft Corporation includes P&F in its Cloud Design Patterns [25].

The challenge however is to allow dynamic composition of services in a federated environment in such a way that a composed service is also protected by a newly created SLA.

The service protection by an SLA here means the following: service scope and service levels defined in an SLA are being monitored by a provider, can be observed by a consumer, and in case of [predicted] violation an action is taken by a provider to reduce the harm. SLA violation here is such degradation of prescribed KPI's that their values are not within the prescribed ranges. SLA's per se are not in the focus of this document and should be detailed elsewhere, while in this document we shall concentrate on

- methods to predict SLA violations, and
- actions to be taken to avoid SLA violations.

We conjecture that the above two items are largely defining the OLA design, because methods for SLA violation predictions must inevitably be based on KPI monitoring and the actions to be taken in response to the predictions must be applied to root causes of the KPI's violations. Accordingly, the KPI set must be defined and clustered into two groups:

1. Monitored KPI (mKPI) set that is being defined by the offered service portfolio, and
2. Target KPI (tKPI) set to be acted upon.

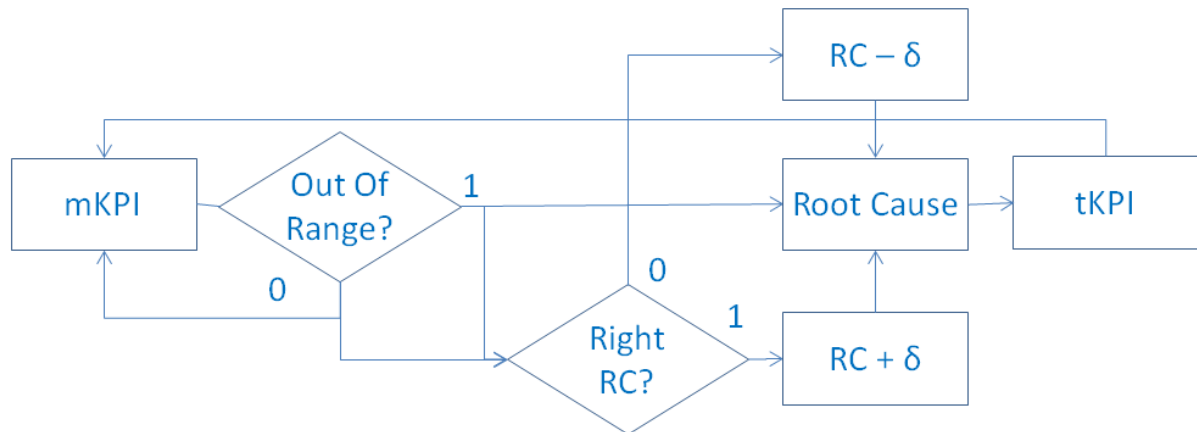


Figure 18: Generic root cause adaptation scheme

In Figure 18 we sketch a very generic scheme that shows the intended relation between mKPI and tKPI and outlines a simplistic prediction algorithm, where Root Cause is a database containing records of type

$$mKPI1 \rightarrow (\delta_1, tKPI1), (\delta_2, tKPI2), \dots (\delta_N, tKPIN).$$

where the root cause mapping $mKPI(i) \rightarrow tKPI(j)$ is initially set based on experience and/or best current practices, $\delta(j)$ – is a weighting coefficient either incremented by δ if a RCA was successful as observed within a pre-defined monitoring period after a pre-defined action on KPI8i)I violation did happen, or decremented otherwise.

4.5.2 Root Cause Adaptation (RCA) Mechanisms

An OLA like any business agreement should be able to cope with change. Indeed the change is ubiquitous in a complex federation: members and customers come and go, technologies are being deployed and abandoned, etc., etc. To cope with this relatively long-term changes a federation must deploy the right management and governance structure, to which federation members – per Agreement noted above – would delegate the right to make respective decisions. Frequently such management and governance structure is a three-tier one: Assembly – Board – Executive Committee, and also frequently a mechanism to define and to implement needed changes is policy.

We intend to demonstrate how OLA can be systematically implemented based on federation policies in a robust yet sensitive way, understanding robustness with respect to a set of monitored KPI's and sensitivity with respect to a set of offered services.

The root cause analysis (implemented right) plays the central role in achieving robust and sensitive OLA, however it is fundamentally hard because of the multiplexing that happens at various levels. It largely depends on the load known to be self-similar (at least because any cloud includes networking), but also it depends on the cloud architecture and configuration. In a federated cloud infrastructure it is hard to expect a solution that fits all heterogeneous components, hence we have to look at adaptation mechanisms from the two viewpoints.

An adaptation mechanism of the 1st type is required to translate mKPI's as well as tKPI's from a cloud specific to a federation generic form; this adaptation mechanism can be termed syntactic adaptation, because no change of the meaning is required.

Second, an adaptation mechanism is required that shall modify the RCA mechanism in response to the observed behaviour of mKPI. Under the behaviour we understand the following observed sequences (each sequence is a time series, not necessarily representing a casual relation):

- mKPI violation → action on tKPI → mKPI improvement (positive)
- mKPI violation → action on tKPI → mKPI violation (negative)

For the initial deployment of OLA we would assume that prediction is one of the roles of a Federation Board; this body generates and maintains federation policies constituting OLA.

4.5.3 Operational Definitions for OLA

Our pragmatic approach to implement and to maintain OLA is rooted in the envisaged three-tier governance structure of a sustainable federation, where the Federation Board is empowered by the Assembly with the right to define and maintain Federation-Wide Policies within the prescribed range of agreements. The set of FWP constitutes federation OLA and is enforced by the Federator. As any business agreement OLA, meaning a set of FWP must be based on operational definitions.

Following [26] the FWP should be operationally defined through a set of KPI's that are uniformly measured federation-wide, meaning that their semantics and the process of monitoring are the same throughout the federation. It is reasonable to suggest that the set of these uniform KPI's (uKPI) is based on utility metrics. However we need to recognise that typically policies are associated with services, no surprise because they are strongly connected to respective SLA's.

4.5.4 Service Policy

Indeed, in a service oriented world an OLA can be seen as a placeholder for policies. A federation may use different methods to keep a record of all policies that are agreed upon and that are maintained, that is managed, enforced, modified, etc. in full accordance with the policy life cycle. However keeping policies aligned with an OLA helps federation to migrate from pure access control to a process control, which in turn facilitates another migration path, namely from point correctness to process correctness as required by the sustainability of a federation.

In the below we demonstrate pragmatically how to associate policies to services and how to differentiate them from those policies that constitute OLA.

Consider a hypothetical service CCA - Cost-efficient Cloud Analytics, which can be used either by a cloud administrator or by a cloud user – developer. Typically, CCA policies will be defined as

- Service access policies : Administrator: no constraints; Developer: access on valid credentials;
- Service usage policies: Administrator: per offered SLA's; Developer: that of resource usage (quota|time|reservation<...); in both cases may include additional dependability policies such as the number of concurrently used resources, etc.;
- Service pricing policies: Administrator: none; Developer: Flat | Reputation based | Load based | Utility based |

Thus defined service policies are generic enough to be applied to a reasonably wide spectrum of cloud services.

Obviously the above three types of policies are service specific. For SLA monitoring they will need to monitor service specific mKPI, and, as explained above, decision on target KPI for service improvements may be hard.

4.5.5 OLA Policies

We explain our approach starting with the separation of two concerns:

- Action on tKPI in case of mKPI violation (under assumption that RCA is known and valid);

- RCA adaptation.

This is demonstrated in Figure 20 (which is just a rearrangement of blocks shown in Figure 18), where internal module abstracted in future as $AR(m,t)$ – action result on t -th $tKPI$ after violation of m -th $mKPI$ under and assumption that RCA is done right. The outer module provides RCA adaptations but not after each $mKPI$ violation (which would be service specific $mKPI$) however based on smaller set of indicators, namely on violations of $uKPI$ i.e. those that serve the basis of federation-wide policies.

Indeed, in order to be able to achieve the above separation the mapping outlined in Figure 19 must be understood.

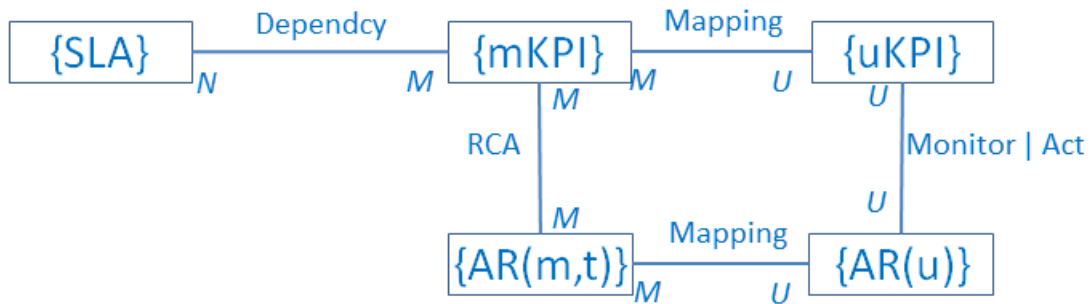


Figure 19: Mapping between service and utility KPI's

We denoted as $mKPI$ a set of all KPI's that constitute all SLA's offered by a federation, $uKPI$'s are those that constitute the OLA; the $AR(u)$ is the result of action performed under the violation of $uKPI$, the mapping $AR(u) \rightarrow AR(m,t)$ is the one that makes the RCA relatively easy.

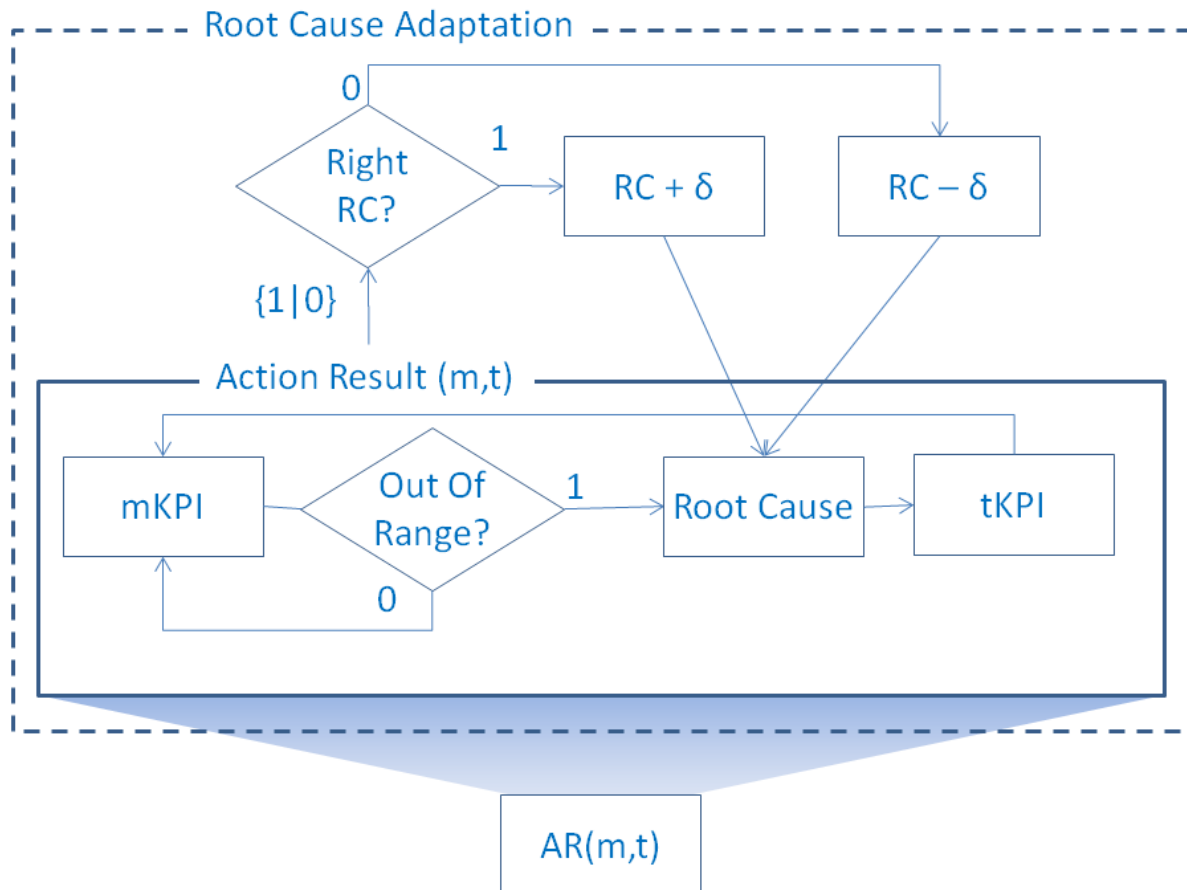


Figure 20: Separation of concerns in KPI mapping

Finally, Figure 21 outlines the Big Picture of the approach.

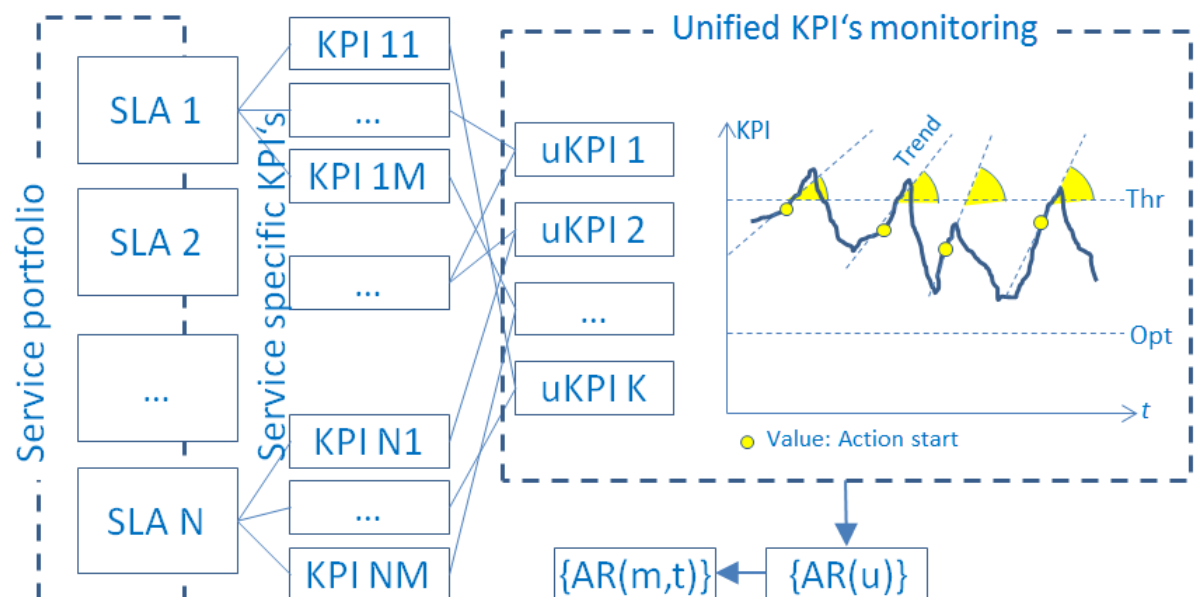


Figure 21: OLA via uniform KPI's

SLA set covering the offered services is based on mKPI set that contains service specific KPI's; one can assume that service degradation is experienced under violation of a single one mKPI from that set. However, as it is well known the sets of mKPI's pertaining per each service are not disjoint and/or independent for all offered services, thus it is always possible to define generic (unified) KPI's based on service specific ones, for example as it was done for LTE SON in 221[27] based on utility concept. Thus obtained uKPI constitute federation's OLA and enjoy uniform monitoring procedures. In essence, uKPI are all being targets for actions under violation of any of uKPI thresholds, because the federation utility is then the only "monitored KPI". This does not mean that – much simplified in this case – RCA is not needed, however it translates into a set of parameters, on which the uKPI depend.

4.5.6 OLA Implementation Status

The Operational Level Agreement (OLA) objectives are in general fulfilled today. The OLA encompasses a number of well-defined, concise and unified Key performance indicators, which are applicable for the entire XIFI service catalog.

The required API tools [28], service policies, mindset for federation monitoring related to the operational level agreement, are in place.

- All services are possible to measure [29] individually in a quantitative and quantifiable way and can be directly linked to the OLA and the underlying service level agreement in a clear and unified context relation to each key performance indicators.
- At the present moment, the violation monitoring can be performed, by initiating and running verification scripts on the nodes within the federation. The output file of the script can confirm a complete fulfilment of the OLA on a simplified level for hardware, connectivity and software requirements as a good starting point. Further work is foreseen to automatically manage the operational level agreement including: violation prediction, actions to be taken to avoid violations and root cause of operational level agreement violation.

The operational level agreement has unified key performance indicators that are divided into three layers: Connectivity (e.g. Available Connectivity between nodes), Hardware (e.g. minimal hardware capacity and status of running wms) and software (e.g. open stack service components and indirect GE / SE)

All Key performance Indicators for the services have their own individual conditional and allowed maximum level of deviations in terms of the operational level agreement. The major purpose of the operational agreement is to steer the XIFI federation in a direction of always keeping the operational levels of all service in a high availability state.

4.6 Operational Requirements and Procedures

Below a number of procedures are defined that are intended to describe the Infrastructure Owner operations. This definition is built on the experience of XIFI partner gained from running the FIWARE Lab legacy platform in FIWARE project. The information in this section complements the information in the handbook Deliverables D2.1 and D2.4.

Sections related to identity management and resources management has been added to solve uncontrolled expansion of resource consumption that appeared since the openness of FIWARE Lab to FI-Developers.

4.6.1 Policy for Identity Management in FIWARE Lab

Tenant User Identification

This procedure is used for all the associations in which the data of tenants have been stored only in the Telefonica database.

It is required by the national policy authority to keep the log of who was using a certain public ip address at a certain time, a detailed and specific process that let Infrastructure Owner to retrieve this information is needed. At the time being this info should be present in Telefonica Database.

The data (logs) needed to identify the user of a certain public IP addresses have to be available for the last 12 months.

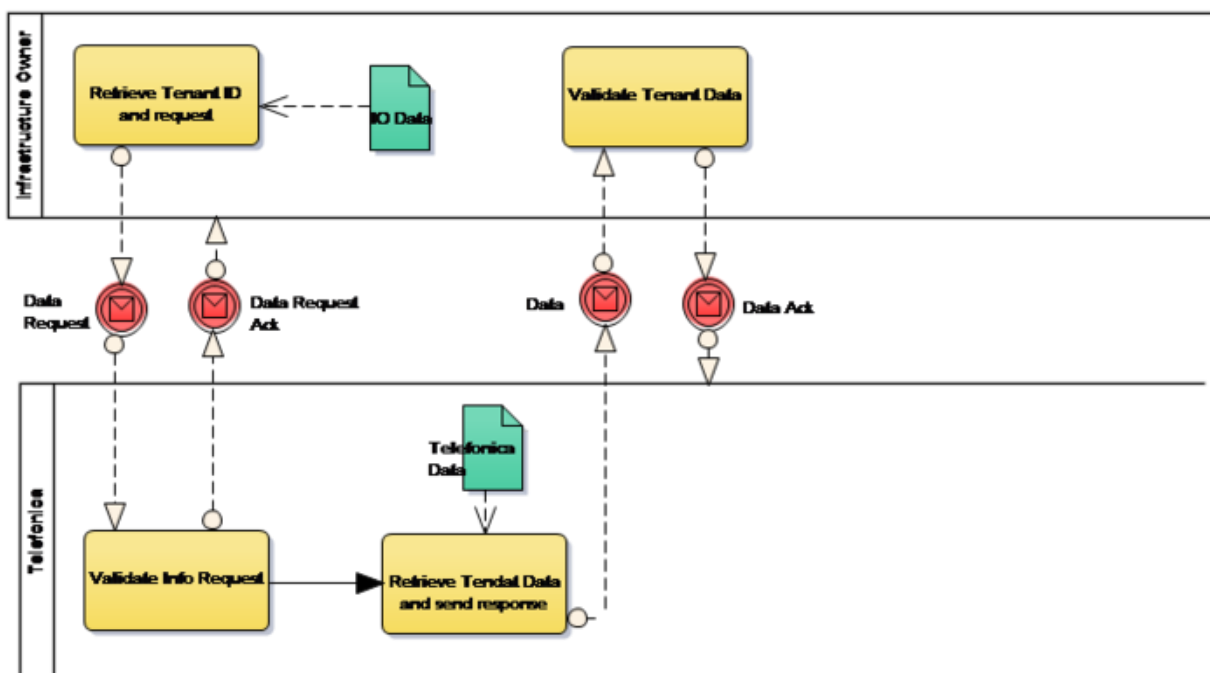


Figure 22: Tenant user identification

Infrastructure Owner

The Infrastructure Owner is providing the pool of public ip addresses and has to be able to retrieve the data identifying the user of a certain public IP address in the period of the 12 months previous the date of the policy authority request.

- **Retrieve Tenant ID and request**
The infrastructure Owner retrieves the Id of the tenant that was using the public IP address at a certain interval time (dd/mm/yy hh:mm:ss of the start, dd/mm/yy hh:mm:ss of the end of the period) and sends to Telefonica the request to identify the legal representative of the tenant.
- **Data Request**
The data request will contain:
 - tenant-id
 - start of the period (dd/mm/yy hh:mm:ss)
 - end of the period (dd/mm/yy hh:mm:ss)
- **Validate Tenant Data**
Infrastructure Owner validates and acknowledges the data.

- Data Ack
The Infrastructure Owner validates and acknowledges the receipt of the data within 1 business day via e-mail.
- IO Data
The Infrastructure Owner local data is a log where are recorded for the last 12 months:
 - the Ids of the tenants that have utilized or are utilizing the node resources (computing, storage, ip addresses public and private)
 - the correspondence between the tenant Id and the resources utilized at a certain time at least in the previous 12 months.

Telefonica

Telefonica is keeping at the time being the database of the users. Telefonica will be responsible to keep the correspondence of the Tenant User Data with Tenant Ids at least for the previous 12 months.

- Validate Info Request
Telefonica validates and acknowledges the data request.
- Data Request Ack
An ack to the data request has to be sent within the next business day.
- Retrieve Tenant Data and send response
Telefonica retrieves the user requested user data and sends an e-mail (same sender-receiver e-mail address list).
- Data
The data has to be sent via e-mail within 3 business day from the data request and will contain:
- Telefonica Data
The user data have to be enough to identify in a precise, clear and secure way the user of a certain ip address:
 - First and Last Name
 - Date and place of birth
 - Legal Address
 - Identity Number (Identity card or tax code)

Business Process - Identity Management

This document describes the Identity Management process handled autonomously by every XIFI node.

The main objective of this document is to provide a set of procedures for the identity management of the XIFI nodes.

There are legal issues that regard the identity of users that utilize the public IP in the each infrastructure. For the Infrastructure Owner it is very important to keep trace of the history, to whom a certain IP address was assigned at a certain time (detailed and precise data). This process has also to deal with identity thefts, an increasing phenomenon nowadays, and should make data accessible at any time for the Infrastructure Owner.

XIFI-users can be not only companies or developers, but can also be projects. It is fundamental to identify a person who is legally responsible for the resource usage.

It is assumed that under normal conditions the IOs have free public IPs at his disposal.

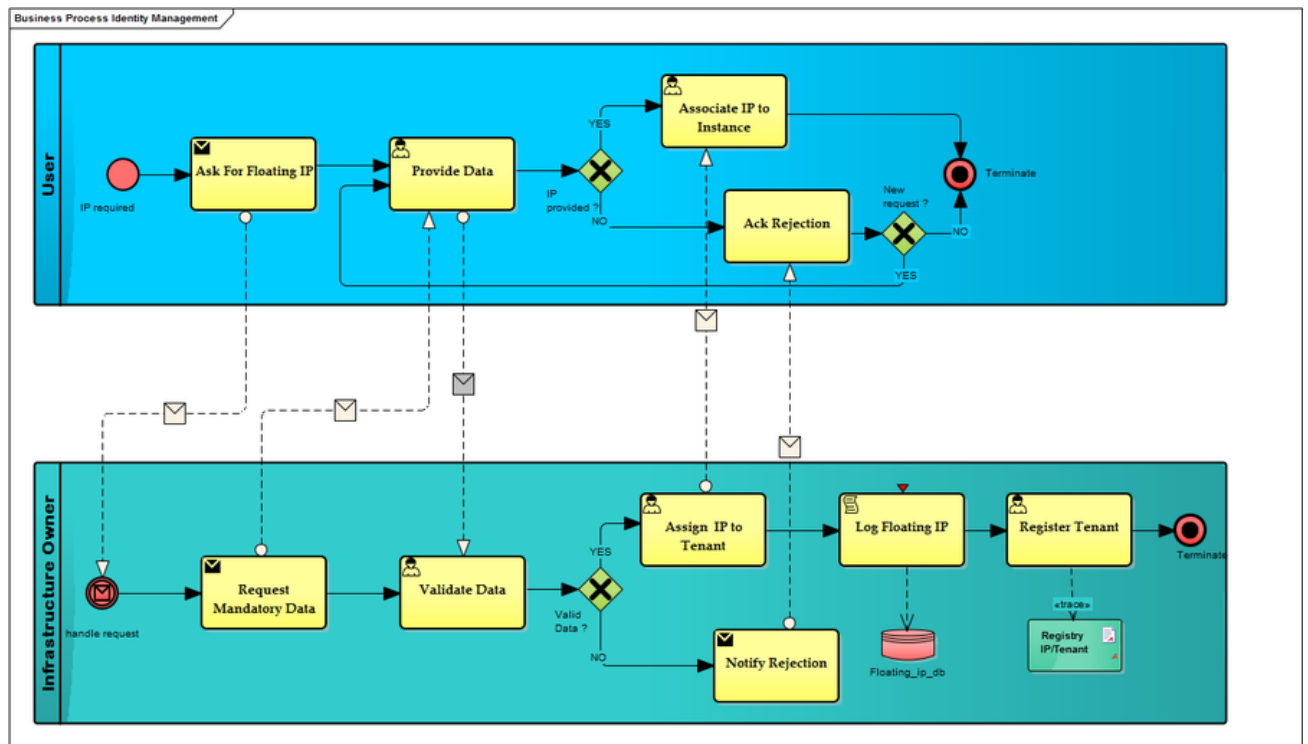


Figure 23: Identity management

ELEMENTS OWNED BY Infrastructure Owner

- **Request Mandatory Data :**
After receiving the request, the IO evaluates and should answer the request within 2 working days, by listing the necessary documents to be provided for the release. The Infrastructure Owner informs the user about the documentation to be provided in order to gain access to the requested resource:
 - Copy of an identity document (identity card, passport or driving license)
 - Mobile phone number
 - Institutional e-mail address (in case of academic projects and so on)
- **Validate Data :**
In this phase, all the documents submitted by the user are analyzed, validated and stored by the IO (or company legal office). As mentioned, during this phase it can happen that additional details/documents must be requested to the customer for the identity verification. The validation step can be completed within 3 working days from the user request. Every Infrastructure Owner can have its own validation procedure for the documents provided, complying with the laws of the country.
- **Valid Data? :**
After evaluating the data provided the Infrastructure Owner decides whether the data is sufficient to assign the resource to the user.
- **Assign IP to Tenant :**
In case of positive validation of the documents provided by the user, a public IP address will be assigned to the project/tenant.
- **Notify Rejection :**
In the case of negative response from the validation phase, the user will be notified with the rejection of the request by e-mail and the justification of the rejection.

- Log Floating IP :

By using the floating_ip_log script, the Infrastructure Owner is allowed to log the usage of the floating IPs. By viewing the table it is always possible to identify the tenant that was using a public IP at any time.

- Register Tenant :

Also, the IO could decide to fill also a document in which the information about the tenant and the IP used in a certain period are stored in order to keep trace of the assignments.

- Floating_ip_db :

MySQL database where all the associations tenant/Ip are plotted:

ip_address	tenant_id	port_id	start_date	end_date
193.205.211.165	10d303d5ed1c468a92d19fc6e18117c7	3d09b8d3-c21f-4756-8461-8704409bd5b8	2015-01-26 11:50:52	2015-02-03 14:02:37
193.205.211.167	10d303d5ed1c468a92d19fc6e18117c7	46aa53c0-995c-4dd6-b2c7-e9b6e2c645ea	2015-01-26 11:50:52	NULL
193.205.211.168	10d303d5ed1c468a92d19fc6e18117c7	4ff62263-bd6f-448e-b9e9-75923da1df0	2015-01-26 11:50:52	2015-02-04 10:44:39
193.205.211.166	10d303d5ed1c468a92d19fc6e18117c7	ad1d8a2d-0240-48b1-a96a-f7d0639dbd5a	2015-01-26 11:50:52	NULL
10.0.40.11	10d303d5ed1c468a92d19fc6e18117c7	ed46d433-d92f-4290-bfff-397bd9f10a96	2015-01-26 16:21:37	2015-01-26 16:34:49
10.0.40.12	10d303d5ed1c468a92d19fc6e18117c7	e448022e-d9bc-4a59-a89d-7ae3cafe1e07d	2015-01-26 16:22:59	2015-01-27 14:17:09
193.205.211.172	56ba55df184747a8a95abf8b511623b3	03a06b56-eb7b-43f4-9c81-062201f51892	2015-01-27 14:52:32	2015-01-27 14:55:22
10.0.40.12	10d303d5ed1c468a92d19fc6e18117c7	51b9c1b4-4d3e-4a27-847f-61dec68fa652	2015-01-27 17:01:56	2015-01-27 17:26:22
10.0.40.15	10d303d5ed1c468a92d19fc6e18117c7	4076b624-e4fd-47de-b49a-a31f1cfda612	2015-01-27 17:35:32	2015-01-27 17:55:40
10.0.40.18	10d303d5ed1c468a92d19fc6e18117c7	d3383f3b-a952-484f-926e-6e05c0b9b21a	2015-01-27 17:54:42	2015-02-03 13:59:44
10.0.40.10	10d303d5ed1c468a92d19fc6e18117c7	6473581c-bb89-4d41-8262-38494996f6da	2015-02-03 15:00:41	NULL
193.205.211.173	10d303d5ed1c468a92d19fc6e18117c7	e466d121-25fe-4ccc-b075-ecd30d19c786	2015-02-03 16:11:05	NULL
193.205.211.174	56ba55df184747a8a95abf8b511623b3	8dd2a26d-3ab1-4de2-857c-b4d83a73d2d9	2015-02-03 17:00:03	2015-02-05 09:58:09
193.205.211.165	56ba55df184747a8a95abf8b511623b3	b0650953-0bba-4bb5-8a53-e8c7bf2485eb	2015-02-04 12:04:07	2015-02-05 09:58:09
193.205.211.168	56ba55df184747a8a95abf8b511623b3	1df47679-728a-42e2-9a61-ebce267ea643	2015-02-05 09:59:32	2015-02-05 10:00:20
193.205.211.169	56ba55df184747a8a95abf8b511623b3	74f15cb9-c5e0-45a7-ad0b-72cb597e4b89	2015-02-05 09:59:50	NULL
193.205.211.170	56ba55df184747a8a95abf8b511623b3	4c88f0f0-c62a-4010-b6a3-7f9f728c6954	2015-02-05 10:17:03	NULL
193.205.211.171	56ba55df184747a8a95abf8b511623b3	e94d3b74-dcc5-443d-a77a-d659eda3ae93	2015-02-05 10:20:03	2015-02-05 10:22:35
193.205.211.165	10d303d5ed1c468a92d19fc6e18117c7	0e0f750f-a4ca-47c8-b854-df097c08f6dd	2015-02-09 14:34:47	NULL
193.205.211.171	19ae6963a5c845d39f9149384214b6db	cf5f02e0-46c8-449d-ac8a-317044504d93	2015-02-12 14:14:13	2015-02-12 14:14:42
193.205.211.172	19ae6963a5c845d39f9149384214b6db	4f495c90-1826-47a6-83bf-8ca4d8f8127c	2015-02-12 14:40:01	NULL
193.205.211.174	19ae6963a5c845d39f9149384214b6db	d775bb06-b31e-4ca7-8a56-71a7a4188c86	2015-02-13 15:04:21	NULL

Figure 24: Floating IP database

ELEMENTS OWNED BY User

- Ask For Floating IP :

The request for a new public IP release must be performed from the interested user or in case of projects by the person indicated as legal responsible. The request must be submitted via e-mail.

- Provide Data :

The user provides the requested data to the Infrastructure Owner. This operation could end up after several interactions between the interested parts.

- Associate IP to Instance :

After being notified about the positive evaluation of the request the user proceeds with the association of the IP to the proper instance.

- Ack Rejection :

Once notified about the rejection, the user can anyway decide to continue providing the missing data to the Infrastructure Owner or to terminate the request.

- IP provided? :

Based on the IO reply, so if the public IP is provided or not, the user proceeds with the allocation or the acceptance of the rejection notification.

- New request? :

After receiving the rejection notification the user can decide whether to provide the requested data or to end up the request.

4.6.2 Policy for Resource Management in FIWARE Lab

Resource management in FiWARE lab is generally complex as it must strike a balance between the capabilities of the technologies, the capabilities of the nodes, the capabilities of the federation and the needs of the users. A short discussion on each of these is below. It is worth noting that the resource management issue primarily arises regarding IPv4 floating IP addresses as these are very constrained, although it is also likely that congestion will appear in compute and storage resources as usage of the system grows.

The FIWARE cloud chapter is strongly based on Openstack which was not originally designed to operated in a federated model. While resource management is an essential aspect of the design of Openstack - resource management is probably the most important issue in a cloud management stack - models of users and groups of users are not so sophisticated in the system. Openstack essentially assumes another service will perform user management and group users in a sensible fashion to which the resource management policies can be applied. Further, the mechanisms by which Openstack can track usage and interact with users relating to their usage is very basic: currently, the primary mechanism to do this is via the Ceilometer database which was only introduced in the Havana release of Openstack and even this has minimal active mechanisms - it is mostly a data store which can be used to analyze usage.

Another challenging issue which Openstack has no support for is dealing with resource hogs - users who allocate resources at some point in time (and are within quota) but never free these resources. Of course it is not easy to differentiate between a valid user who is not currently active and a resource hog. In Openstack, it is assumed that the cloud provider can communicate with the system users and understand their requirements in this fashion; if the cloud provider is not satisfied with use or the usage patterns, then the cloud provider can remove specific resources or, in the worst case, revoke access to the cloud resources entirely.

Apart from the technology considerations, each of the nodes has a slightly different perspective, depending on their available resources. For example, each of the nodes has a different number of available floating IP addresses, ranging from a very limited number (~20) to a limited number (~250). Further, depending on the particular node, there can be legal obligations relating to these IP addresses, which impact the node's perspective on how they are managed.

The federated solution also introduces further complexity as it necessitates making system-wide resource management policies for a loosely coupled system. The XiFi federation solution involves leveraging the user and project/tenant models of Openstack: more specifically, each account on Filab has a dedicated user and tenant which is available on each node. This means that each user of the federation has independent quotas on each of the nodes. Further, these quotas are managed separately, according to the policy of each node. As such, the XiFi federated solution does not inherently support a holistic resource management solution and consequently, a solution which can be implemented reasonably within the nodes and is reasonably clear to the system users is required.

Finally, the system users have some expectations regarding how the system should operate: they should be given a reasonable quota to enable them to use the system from the outset; they should be able to understand when they try to exceed this quota and they should obtain feedback from the system if they are consuming resources which could be better allocated elsewhere.

With these considerations in mind, XiFi developed a resource management strategy for FiWARE Lab. The approach used was as follows:

1. clearly articulate the resource management issues in the project
2. propose a set of policies which are realistic in light of the issues identified
3. propose mechanisms by which these policies could be implemented

The issues identified were as follows:

Issue number	Description
1	Currently, there is no way way to distinguish “important” users from “try-and-escape” users. This may lock unused (or poorly used) resources. This does not help us to know what the user do in FIWARE Lab and does not help us pushing users in doing something nice and let us know about that.
2	Even though we defined resource quotas for Openstack, these quotas are valid only per a single node (i.e. if N is the number of nodes, and Q the quota, a single user can get NxQ resources)
3	The most limited resources are Public IPv4 IPs
4	Most of the nodes, for legal national reasons, need to be able to trace who get allocated a Public IP. Consequently, nodes are being conservative with the issuing of public Ips eg you can obtain a Public IP if you issue a ticket

Table 60: Resource management strategy –issues identified

Having identified the above issues, the following set of policies were proposed (Table 61). Essentially, the solution is to differentiate between users: Trial users who have less rights on the system and Community users who have more rights. In general, the needs of the Community users take precedence over the needs of the Trial users and the FIWARE Lab nodes can be more aggressive with reclaiming resources from Trial users.

Policy number	Description	Related issue
1	Define 2 (or more) category of users and define different resource policies for the the 2 category. e.g. “Trial” and “Community”. Within “Trial” category, nodes are free to assign additional resources when they think it is justified.	1
2	“Trial” users can access FIWARE Lab for 14 days, then their account is disabled and their resources released - any backup is up to the users. Reminders are sent one week before the expiration and the day before, including. To keep alive the account they need to apply for “Community”.	1
3	“Community” and “Trial” users cannot create Organizations beyond the one that maps to their account.	2
4	“Trial” users can access only a limited number of nodes. Nodes will be established according to resources they can share and their will do to so. This will be a criteria for FI-Core Open Call.	2
5	A “Trial” user can use only 1 node at time. (He needs to release resources on one node before using another)	2
6	“Community” users are assigned a default node. The node is selected according to their preference and/or association with a given accelerator that is linked to a default node. This would also “improve” the support chain accelerator FIWARE Lab, by user needing to interact only with specific node support.	2

Policy number	Description	Related issue
7	Accelerators contribute to manage “Community” users assignment (i.e. they guarantee that user X is one of their startups)	1
8	“Trial” users to become “Community” need to be part of an accelerator or describe a meaningful project to they plan to run in FIWARE Lab	1
9	“Trial” users do not have a private network (for node where this is possible), they can use only a shared network.	3 / 4
10	Users (no matter which category) to be able to allocate a Public IP needs to provide a verifiable identity information (e.g. ID card number, Credit Card)	4
11	“Community” users can access additional nodes if well motivated by issuing a ticket. They can also request additional resources using this mechanism.	2

Table 61: Resource management policies description

Then, the specific mechanisms and processes to realize the above policies are listed below.

Action number	Description	Related Policy
1	Prepare an email introducing that there will be a change on policy starting from a given date. Users not willing to accept the policy have 30 days to download their data (if possible), then will be removed. Email should explain how to download their data (if possible).	ALL
2	Implement ways to differentiate users. Best option is creating additional roles in the Cloud Application registered in IDM.	1
3	Modify Keystone Proxy / Keystone (if we can use already the new keystone when we launch the policy) to restrict resources according to the user “category” and associated policy.	1
4	Ability to list “Trial” accounts older than 30 days and delete them in one click from all nodes. (initially nodes can do this manually, but we need to have an easy one for them to know who is “Trial” and who is not among their tenants).	2
5	Ability to send reminders / messages to user with a specific category. (inform before actual deletion)	2
6	Limit the number of organizations a user can create according to a role.	3
7	Enable ways to limit the access to nodes to specific user category, user or organization in the catalogue (keystone) and in the portal. (e.g. by creating a user role per node in IDM - AccessBerlin).	4 / 5 / 11
8	Create an organization per Accelerator and enable the organization to assign the role “Community” and “AccessNodeX” to the users or add them to their organization.	6 / 7 / 8
9	Set-up a process to submit request for “Account upgrade” through JIRA and evaluate the requests.	8
10	Define cloud resource quotas per user category.	1
11	“Monitor” total number of used resources per “tenant” on multiple nodes.	11 / 5

Action number	Description	Related Policy
12	Enable Public IP only to users that provide “full data” (they get by default a “PublicIP” role in IDM)	10
13	“Trial” users cannot “create networks” and “create routers”. Nodes supporting “Trial” accounts need to implement a shared private network.	9

Table 62: Resource management mechanisms and processes

4.6.3 Resource Management Implementation Status

Request for a FIWARE LAB account upgrade

Participants to the FIWARE Accelerator programme and Individuals and Companies willing to develop innovative applications based on FIWARE need to request for free for an account upgrade.

Once submitted the request, the process works as follow:

- The request is assigned to a responsible person, depending of the fact that the applicant selected a given accelerator or none.
 - 1 When the request is associated to an accelerator, the responsible person is the coach assigned to the accelerator.
 - 2 When the request is not associated to an accelerator, the idea is evaluated by a dedicated team.
- The responsible person validates if the request is eligible.
 - 1 When the request is associated to an accelerator, the coach verifies with the accelerator the eligibility.
 - 2 When the request is not associated to an accelerator, the team analyze whether the idea proposed will be relevant in terms of FIWARE adoption.
- If eligible, the responsible person, in agreement with the nodes - for resource availability check - and with the user request, assigns a node.
- If eligible, the account is upgraded by the FIWARE Lab admins.
- Resource information is shared with assigned node that upgrades accordingly quotas if possible (this applies only for the reference account)
- Users are notified of the result of the request (accepted/rejected).

More details about this process are available on :

[http://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE Lab: Upgrade to Community Account](http://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE_Lab:_Upgrade_to_Community_Account)

Floating IP management - Implementation requirements

Floating IPs are scarce resources. They are associated dynamically with tenants and virtual machines in a tenant. This by default is a self-service for the developer only limited by the tenant quota assigned. Since floating IPs are drawn from a shared pool in a first-come first-serve manner, fairness is not guaranteed and overbooking is the regular case. OpenStack by default does not provide any means to implement fine-grained control over floating IP pools. For management efficiency reasons, additional attributes must be associated with a floating IP (i.e. other than tenant ID and association virtual machine) such as firewalls (i.e. filter rules) of the infrastructure hosting the XIFI node.

In consequence, floating IP management as a maintenance task must be implemented both through proprietary tools and by manual procedures. Suitable tools may not be available easily from the community due to the unique requirements of the XIFI federation and the particular legal situation. This causes a significant impact on the response time for maintenance actions involving floating IPs.

The maintenance task for managing floating IPs has to consider technical, operational and legal requirements for all maintenance actions taken. This is due to the fact that, in contrast to other federation maintenance tasks, maintenance of floating IP pools and association (with tenants, routers or VMs) elevates a federated node into the legal role of an Internet Service Provider and opens up the platform for attacks and potential malicious use intended or by accident. In the following, a number of considerations arising in conjunction with providing public IP addresses are summarized to understand their impact on floating IP maintenance tasks.

- **Technical considerations** mainly apply to the availability and use of management information obtainable from the platform (i.e. node, node monitoring, and OpenStack management data accessible through APIs as well as non-disclosed / internal data). Capacity constraints and access control here determine if a certain procedure can be implemented or not.
- **Operational considerations** are with regards to user requirements, to the terms of use (e.g. user management), to existing SLAs and OLAs (e.g. lease conditions for a floating IP, i.e. how and for how long to allocate an IP address), as well as to the feasibility of certain solutions (e.g. completeness, response time). Operational constraints may require to disregard some procedures or options due to resource consumption constraints, maintenance effort required, or delay considerations. In consequence, a maintenance procedure may be impractical or too costly to implement.
- **Legal considerations** apply to the way of storing and processing information, how and for which time frame actions involving the use of floating IPs are logged or observed (e.g. for reasons of privacy, security, or legal interception). Legal requirements need to be considered since the floating IP is an "anchor" to identify, isolate and prevent malicious use of the node.

In the following requirements to the maintenance task are collected as an open list without further classification. The maintenance task given below aims to consider these requirements whenever possible. Remaining open issues will be discussed later on.

It is required to support allotment of floating IPs to tenants.

An infrastructure maintainer must be able to assign a particular IP to a particular tenant or to a particular VM. This feature is needed since the floating IP might be associated with a particular firewall of the node infrastructure. Also, the floating IP pool may be associated with a particular network, switch or route. Selection criteria thus may only be known to the infrastructure maintainer. Additionally, allotment of floating IPs is required to implement private IP pools for a particular organization (e.g. to lease a bulk of floating IPs to a particular use case project).

It is desirable to support self-service of tenants on floating IP pools.

An infrastructure maintainer is not able to manage all tenant's floating IPs manually. It is thus recommendable to allow some or all tenants to obtain a floating IP automatically upon request, preferably from an allotted pool. The self-service feature is the default behavior for OpenStack Nova and may need to be enhanced to coexist with allotted floating IPs. This feature requires maintaining reasonable floating IP quotas for tenants. It is preferred to maintain per tenant quotas.

It is required to support per floating IP lease times.

It must be possible to remove floating IP associations automatically after a well-defined period of time. It already has proven impractical to verify by management actions after some reservation time if an IP is still required by a tenant. This is due to the fact that scarce resources such as floating IPs are observed to be allocated greedy and that they are usually not released by users.

It is desirable to support lease-time based triggers for floating IP maintenance actions.

OpenStack does not provide a lease management for any resources including floating IPs. Lease times are a required feature to implement FIWARE Lab policies for trial users (regarding the 14-days trial-use period proposed). It is thus desirable to set a deadline for automatic removal of trial users and it is also desirable to implement such a feature for all resources such as instances, networks, routers, snapshots, and floating IPs since the removal of a tenant in the identity management does not automatically free up resources allocated by this tenant in the Identity Management. It is rather likely that removing a tenant from the cloud portal's list of accounts will require to remove resources allocated on every FIWARE Lab node prior to that. It is not recommendable to rely on a manual maintenance activity in case of a larger number of trial-users.

It is required to support per tenant quotas for floating IPs if self-service is enabled.

Support of user categories requires per-tenant quotas and a suitable quota management since trial users can advance into a community status. Trial users start from a default quota but may have individual quotas in case this is a requirement for conducting their trials. Past the trial period, a trial user might withdraw or might request an upgrade to a community user status. Community users likely will need individual quotas, which could be managed through a more fine-grained "user engagement level" management. It is necessary to avoid overbooking of resources given the limited node resources (in particular regarding the available floating IP pools). Additionally, it is not desirable to rely on completely individual management since this could create the need for more manual maintenance tasks and bears the risk of creating imminent unfairness and competition. The allowance for resource allocation then could become an individual and subjective decision of a maintainer. Hence, per-tenant quotas should follow clear rules and their way of management must be justifiable.

It is required to log user access to floating IP pools.

Any attempt to associate a floating IP with a tenant must be logged for operational reasons and is therefore stored only in the scope of a particular node infrastructure. It is sufficient to store pool name, date/time of an attempt, tenant ID and if the attempt succeeded or failed. It might be needed to log, if an associated floating IP is subsequently assigned to a VM. It is also required to log when a floating IP is released and returns into the pool. It is not required to log the originating user's clear name assuming that the tenant ID is unique and that tenant IDs are re-used not more frequently than the use pattern analysis time window. This information helps to identify busy hours and to optimize pool sizes. Use pattern analysis should determine the user behavior (e.g. if a user is collaborative or greedy) and should provide thresholds for invoking maintenance actions. The outcome of such analysis also might determine due maintenance actions to prevent resource outage situations.

It is required to log successful allocation of floating IPs.

Association of a floating IP with a tenant might be caused by explicitly requesting a floating IP from a pool or by defining a virtual router and setting this router as the gateway to a public network. The latter implicitly associates a floating IP from that network's allocation pool with the tenant. Since the router already provides NAT for all VMs of this tenant (i.e. this tenant's VMs will subsequently access the public network through an IP address owned by the hosting infrastructure), legal requirements are

imposed to the infrastructure owner to prevent liability issues. For this purpose, it is required to log the tenant ID, the IP address, and the date/time of associating and releasing a floating IP, and to obtain the real name and contact information of the user owning the tenant for being able to identify malicious users sailing under an IP of the infrastructure. In case of organization tenants liability cannot be split (i.e., the owners of particular VMs under that organization must be associated with the VMs of this tenant in the logs. This implies that not only tenant ID but also user IDs part of a VM's metadata must be logged for all VMs instantiated for this tenant. For Privacy reasons the concept of "Datensparsamkeit" (i.e. the paradigm of only handling that data needed for the purpose) must be respected. Therefore, the logging period must not extend the allocation period of any floating IP associated with any of the VMs of this tenant. Usually, a floating IP associated with a VM of a tenant can be utilized only if a router (i.e. the gateway) exists, it is sufficient to monitor the router's floating IP association in order to determine the logging period.

It is required to log association of a floating IP with an instance including the properties of an instance.

To enable incident forensics it is strongly advised to log also other metadata of a VM. For example, any virtual disks or data sets stored for the tenant or shared among them might be affected in case a (potentially malicious) behavior of a VM is under study. As outlined above, logging must ensure that for any time a VM was able to access a public network under the identity of an infrastructure, the association of VM and user must be recorded. Consequently, association of such VM with any other resource (e.g. a volume) must be traceable too. It should be verified further, if there exist any legal requirements that demand (e.g. for the purpose of proving guilty, or avoiding to) to take snapshots of VMs or volumes in case of evidence of malicious use.

Based on the requirements outlined above, section "Floating IP maintenance actions" defined in the wiki (<http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4>) summarizes the maintenance actions (i.e. the primitives) related to floating IP maintenance, their trigger condition (i.e. when to perform the action) and their particular logging requirements. These actions are further detailed below in terms of activities required to implement or utilize the action. A particular complex maintenance task may consist of a sequence of actions listed in the table. For example, removing a tenant may consist of removing all objects owned by the tenant including the return of freed up floating IPs to their respective allocation pool prior to removing the user from the central keystone.

4.6.4 Naming of Networks

External networks

The external network typically provides Internet access for your instances. These External networks are used to give XIFI internal Tenant networks outside IP access on the node. The two Quantum functions that come into play here are Network Address Translation (NAT) and floating IP's, where the tenant can access floating IP address and suitable security group rules. The only tenants allowed to manage these networks are XIFI OpenStack admin tenants.

XIFI External Networks

In XIFI infrastructures owners are to provide the tenant with access to two external networks. In some cases, as in WIT XIFI node, we have sub divided our Federated network (MDVPN) into smaller sub networks. For example 10.0.0.0/20 range (4094 ip hosts) into 10.0.0.0/22 (1022 host IPs) and the rest into /24 (254 host IPs). Also on WIT XIFI node we have been allocated a second /24 (254 host IPs) Public IP block. This means that we need to come up with a logical naming schema for our external networks.

Name Schema of External Networks

After some consideration we have come up with a network naming conversion that is donated as follows:

[IP purpose]-[cloud network]-[index]

- **IP purpose:** Currently we have two this is two flavors **federation** and **public** here. This denoted the two different networks available to the XIFI node. The tenant will be granted a public IP from the label depicting public and or a Federated IP address from the mdvpn label.
- **Cloud network:** here we use the label **ext-net**. This shows that this network is externally accessible. It will allow admins use tools like “grep” and the like from the command line.
- **Index:** We use an index number here. This index represents the subnet part of the external facing network. For example: **federation-ext-net-01** will be used for the first federated “mdvpn” subnet. In short the network label index will increase in correspondence with the instance of subnet.

As you now may notice these are generic and do not contain any node specific detail. So an example for the public network label is **public-ext-net-01** for the first network and public-ext-net-02 for the second defined network and so on. For the Federated network **federation-ext-net-01** for the initial and again **public-ext-net-02** for the second defined subnet and so on.

Details on how to change network name on provisioned networks

```
xifiuser@node-21:~$ quantum net-show public-ext-net
```

Field	Value
admin_state_up	True
id	932080d8-ec7e-4ed1-930a-a85f5571c6a1
name	public-ext-net
provider:network_type	gre
provider:physical_network	
provider:segmentation_id	2
router:external	True
shared	False
status	ACTIVE
subnets	8f52fd93-7cb5-48dd-8340-90592e161ac8
tenant_id	000000000000000000000000000000003013

Figure 25: Change of network name on provisioned networks 1

```
xifiuser@node-21:~$ quantum net-update 932080d8-ec7e-4ed1-930a-a85f5571c6a1
--name public-ext-net-01
```

[illegible]

Figure 26: Change of network name on provisioned networks 2

Internal shared network

Also XIFI have a requirement to deploy **shared networks** that are common to all tenant on the node and are kept uniform across all XIFI federated nodes as they server the same purpose. The first shared network is called **node-int-net-01** and is used to proved tenants with a network that has access to the public internet.

As an example, the Waterford node has implemented this network as follows:

```
quantum net-create --shared node-int-net-01
quantum subnet-create --ip_version 4 --gateway 10.101.10.1 node-int-
net-01 10.101.10.0/24 --allocation-pool
start=10.101.10.10,end=10.101.10.254 --name node-int-sub-01 --
dns_nameservers list=true 8.8.8.8
quantum router-create node-int-net-router-01
quantum router-gateway-set node-int-net-router-01 public-ext-net-01
quantum router-interface-add node-int-net-router-01 node-int-sub-01
```

The second shared network is **federation-int-net-01** and provides tenants access to the federation network. The Waterford node has implemented this network as follows:

```
quantum net-create --shared federation-int-net-01
quantum subnet-create --ip_version 4 --gateway 10.100.10.1
federation-int-net-01 10.100.10.0/24 --allocation-pool
start=10.100.10.10,end=10.100.10.254 --name federation-int-sub-01 --
dns_nameservers list=true 8.8.8.8
quantum router-create federation-int-net-router-01
quantum router-interface-add federation-int-net-router-01 federation-
int-sub-01
quantum router-gateway-set federation-int-net-router-01 federation-
ext-net-01
```

It was found that node router quota limits applied when creating these networks and had to increase the number of assigned router instances to the admin tenant.

```
quantum quota-update --tenant-id xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx --
router 10
```

4.6.5 DNS Service

Hosts

The DNS as a service component uses a number of hosts to provide the service. The overall DNS Service is described in D5.2 - XIFI Core Backbone [6].

XIFI DNS Domain Name Service (DNS) is an Internet service that maps IP addresses and fully qualified domain names (FQDN) to one another. An outline of DNS architecture for XIFI is represented in Split DNS diagram below. It comprises of primary master, secondary master, one internal authoritative slave DNS per XIFI node and two authoritative external DNS XIFI nodes for public queries on the XIFI domain. There is also an additional service requirement of Network Time Protocol (NTP), which is needed to push out of DNS zone files in sync.

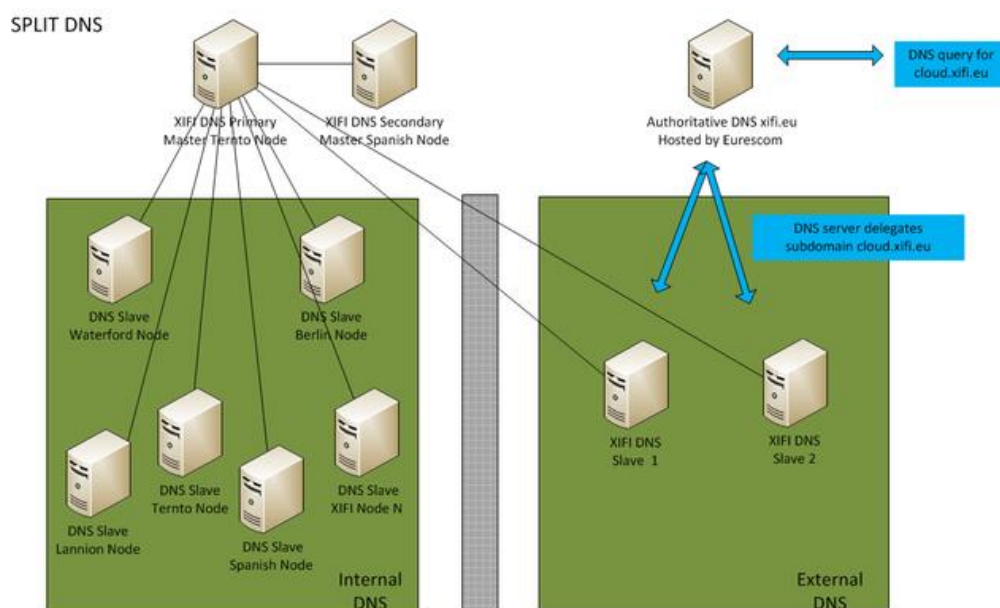


Figure 27: Split DNS

The application servers host the front end for the overall system.

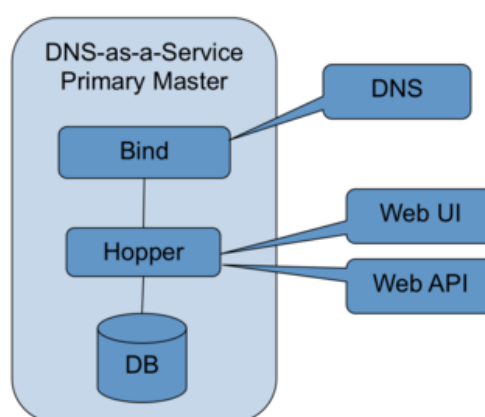


Figure 28: DNS-as-a-Service

Bind provides the DNS lookup service and provides the nsupdate service used by Hopper. Hopper provides the Web UI for Host DNS entry management and provides the Web API for Host DNS entry updates. PostgreSQL Database stores application state for Hopper.

The DNS masters and slaves host the different levels of DNS server to enable a fault tolerant deployment of Bind. The primary master is hosted in Ireland at the Waterford Node. The slaves are hosted in Trento, Italy and Waterford Ireland respectively. This population can be expanded and adjusted as required by the federation.

Deployment

The application server combines **Nginx**, **Gunicorn**, **Hopper** and a custom auth provider to work with the FIWARE LAB IdM GE.

Nginx is configured to run on port 80 as a reverse proxy connecting to a Gunicorn instance on a Unix socket of the localhost. Gunicorn is an application container for Hopper, a dynamic DNS webservice. This in turn uses the nsupdate facility present in DNS servers (in this case Bind) to dynamically apply changes to the zone files. The custom auth provider is a python package used to authenticate users to the service via the FIWARE LAB IdM GE.

4.6.6 Tenant Deployment

Below, a basic procedure is described of what must be deployed by a user (developer) in order to have a tenant with instances up and running.

- Register a user and a Organisation:

Explanation step by step to register a user on the cloud portal and the creation of an organization.

The creation of the user allows to access to all resources.

The creation of an organization is to associate multiple users to a project.

- Use of Network and IP addresses:

Public IP addresses are a scarce resource. They are assigned to tenants as a way to deploy a tenant with its own private router. Care has to be taken that IP addresses are used efficiently. The application of quota, i.e. the maximum number of IP addresses per tenant, is already in place by some nodes.

- Quota:

Default values for a Node are given in the D2.1 "XIFI Handbook v1" as following:

- quota_instances: 3 (number of instances allowed per tenant)
- quota_floating_ips: 3 (number of floating ips allowed per tenant)
- quota_cores: 6 (number of instance cores allowed per tenant)
- quota_volumes: 10 (number of volumes allowed per tenant)
- quota_gigabytes: 1000 (number of volume gigabytes allowed per tenant)
- quota_ram: 2034 (megabytes of instance ram allowed per tenant)

A System Administrator might change the default quotas. For example the disk space associated to user (or project) or a number of volumes per tenant.

As an example, the default number of floating IPs that is allowed by tenant is 3. This number could be judged by an IO as too big taking into account that FIWARE Lab has its portal accessible to anyone without really restrictive measures. A fair number could be put to 1 and be changed by an IO in a case by case manner.

- Security groups:

Security groups allow interaction directly on the firewall rules in authorizing the opening of desired ports on an IP address subnet (CIDR) defined.

After creating a security group, it is possible to associate it with one or more instances of the project.

- Keypair:

Creating keypair allows the association and the use of a public key on the instance.

- Volume creation and attachment:

Creating volume allows the user to benefit from a storage space.

The storage size is chosen by the user depending on use and from the choices offered by the node on which the instance is created.

Different quotas are proposed. Once the volume or volumes are creating the user the possibility to link that storage space to its instance.

- Virtual Machines & instances:

Users / Tenant can actually choose from a list of images identified in section 4.6.11.

Once the selected image and the indicated configuration data, the virtual machine enters in a building state during few seconds before activation.

Activated instance, allows the user to access to the FIWARE catalog and permit to install compatible generic enabler with installation of the selected image.

The user has the possibility of administering his machine by protocols defined in the security groups and available by the basic system selected previously.

Finally, the cloud portal also offers to the user the interaction with his virtual machine via VNC interface available in the details of the instance in use.

4.6.7 Basic Tenant Deployment Procedure

Create new organisation:

New organisations are created only in the federation portal, not at nodes level. To create a new organization, you must go to the Account part. In this field, you will also grant the different kind of accesses have the users of your organization.

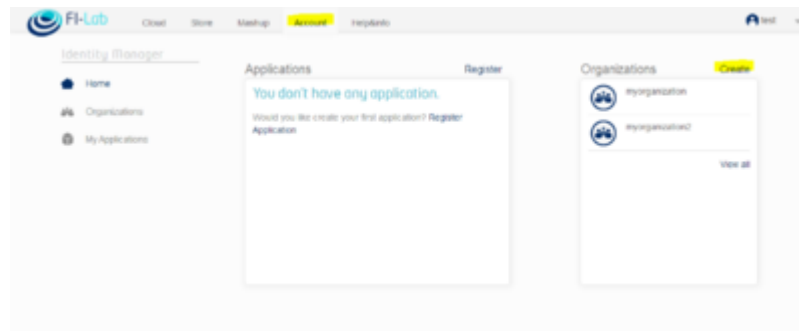


Figure 29: Account part

Once you are done with granting access, you can go back to the cloud portal and choose the organization you just created as "Project Name" and the Region where you want to start building your tenant.

Create network and subnet

First of all you must create a network with a private subnet. To do this, you should click on the "Networks" button then click on "Create Network" (see screen shot below)

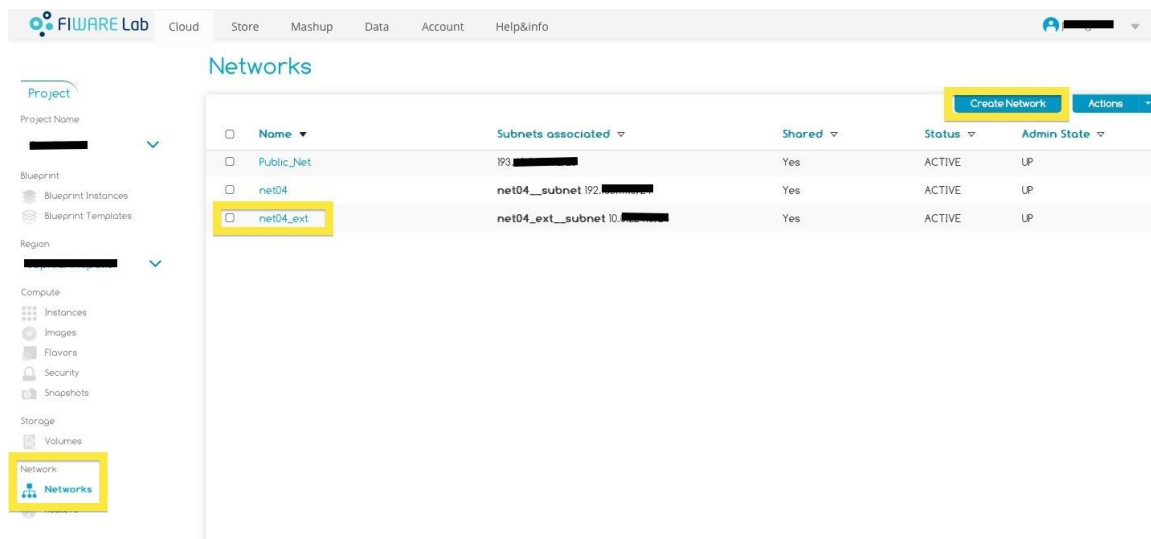


Figure 30: Create a network

Fill the information to create your network, e.g.:

Network name: mynetwork

Subnet name: mysubnet

network address: 192.168.0.0/24

Gateway IP: 192.168.0.1

and push the create button.

Create a router:

- Create a router by click on "Routers" on the bottom of the left menu, then "Create Router"

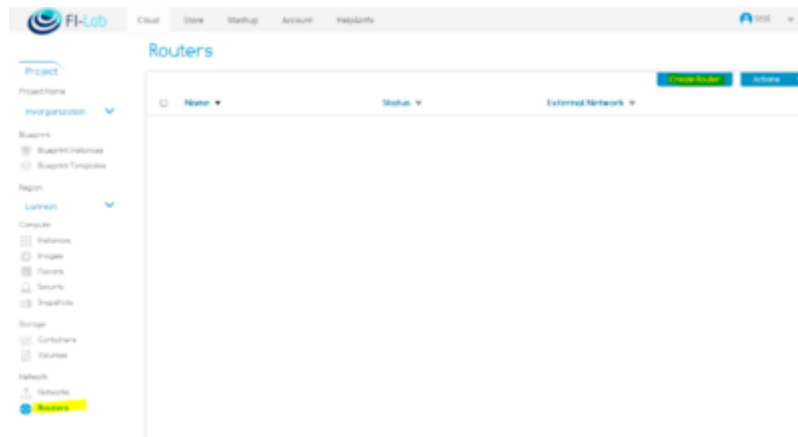


Figure 31: Create a router

- Add an interface: Click on the "router", then add an interface corresponding to your private subnet you created earlier
- Set Gateway: From the main router menu, click on set the gateway and choose the "Public External Network"

If you go back to your interface details of your router, you should at this stage, 2 interfaces: 1 corresponding to your private subnet and 1 for the external network.

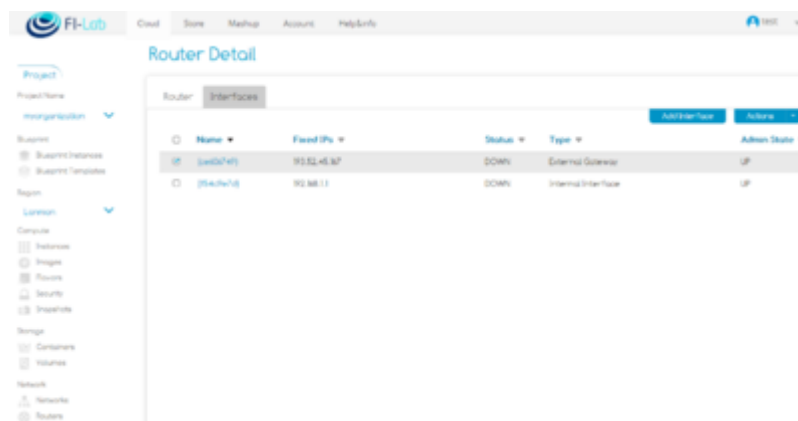


Figure 32: Add an interface

Security groups

To create your security group, you should click on the "Security" button then click on "Security Groups", then "Create Security Group" (see screenshot Figure 33).

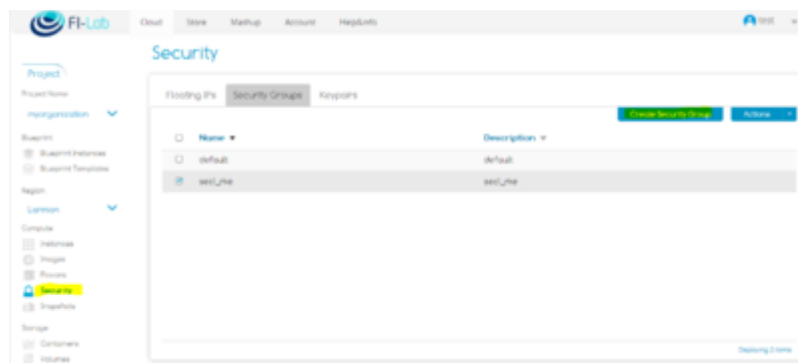
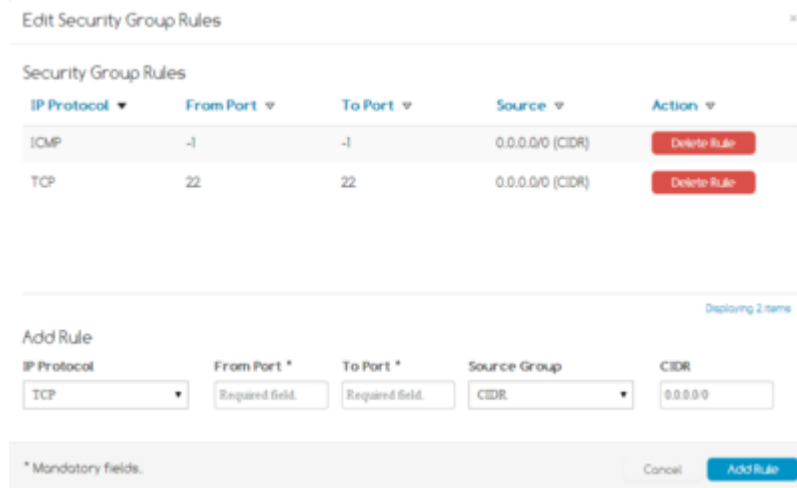


Figure 33: Create Security Group

Add (some) rules to your security rules, see Figure 34.



Edit Security Group Rules

Security Group Rules

IP Protocol	From Port	To Port	Source	Action
ICMP	-1	-1	0.0.0.0/0 (CIDR)	Delete Rule
TCP	22	22	0.0.0.0/0 (CIDR)	Delete Rule

Deploying 2 items

Add Rule

IP Protocol: TCP
 From Port: Required field.
 To Port: Required field.
 Source Group: CIDR
 CIDR: 0.0.0.0

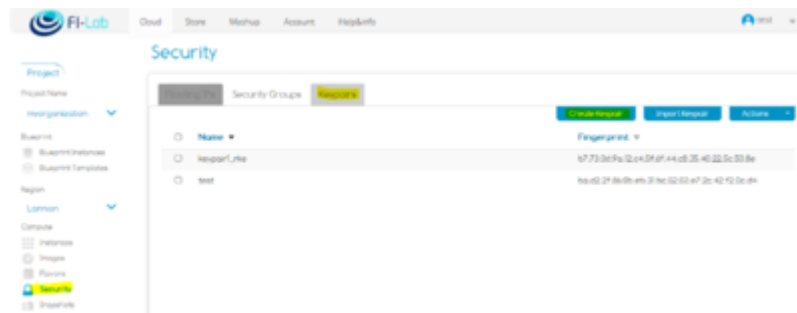
* Mandatory fields.

Cancel Add Rule

Figure 34: Add rules

Create the Keypair

Click on "Keypairs" Tab (Figure 35), create your own keypair. At the end of the creation, it is proposed to download the keypair, then you must answer "yes" as it is the only time you can do it.



Security

Security Groups

Keypairs

Name	Fingerprint
keypair1	1773:0a:Pa:2e:0a:0a:0a:0a:0a:0a:0a:0a:0a:0a:0a:0a
test	1a:02:2f:8b:0a:0a:0a:0a:0a:0a:0a:0a:0a:0a:0a:0a

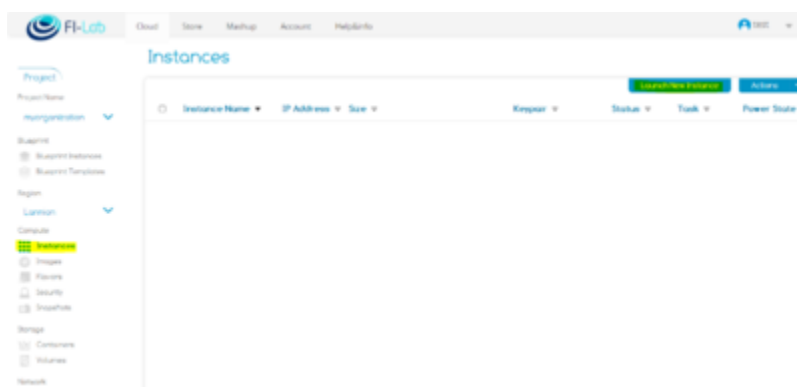
Buttons: Create Keypair, Import Keypair, Actions

Figure 35: Define the Keypairs

Note: It is important to know that a keypair is associated with a user and with a tenant. In other words, it means when you create keypairs, other users having access to the tenant won't see and won't be able to use keypairs you created.

Create your first VM

From the left menu (Figure 36), select "Instances", then click on "launch New Instance"



Instances

Buttons: Launch New Instance, Actions

Instance Name	IP Address	Size	Keypair	Status	Task	Power State
---------------	------------	------	---------	--------	------	-------------

Figure 36: Create an instance

- Image: Choose the cloud image you want to use to create your VM
- Name: Put the name of the VM of the VM you want to create
- Flavour: Choose the flavor you want to be applied for your VM
- Instance Count: 1
- Define the keypair and the security group: Select the one you just created
- Networking: Choose the private subnet you created earlier
- Then push the "Launch instance" button

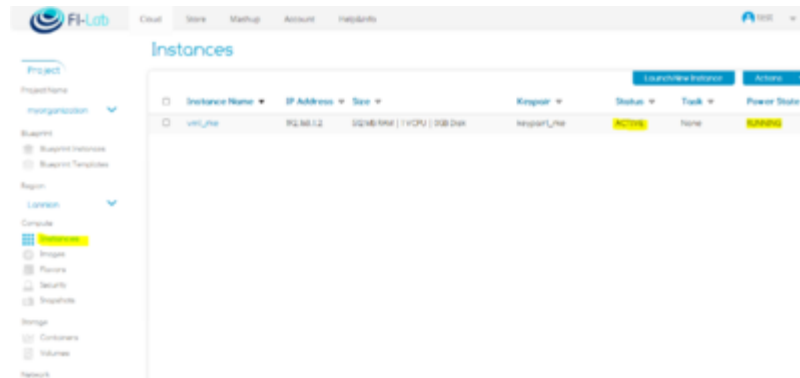


Figure 37: Launch an instance

Once launched (Figure 37), you can click on it, to check the logs and that you instance has been created successfully, Figure 38.



Figure 38: Instance Log

Floating IP:

- Allocate an IP to the project:

Under the left menu, click on "Security", choose the "Floating IPs" Tab, then click on "Allocate IP to Project"(Figure 39)

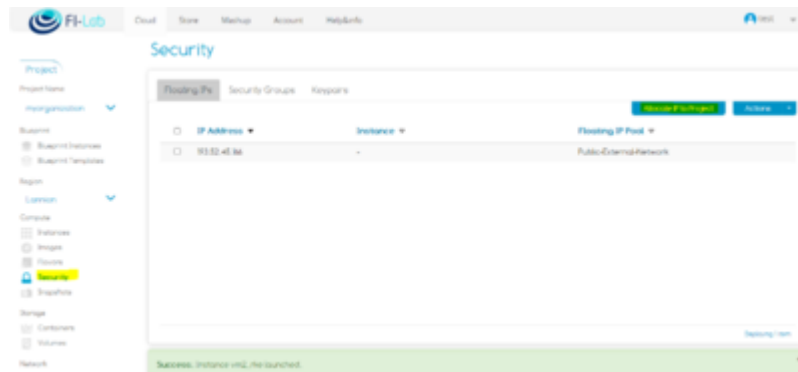


Figure 39: Allocate IP

- Associate the IP to your Instance

In the "Action" field, click on associate IP and select the instance you want to associate the IP (Figure 40)

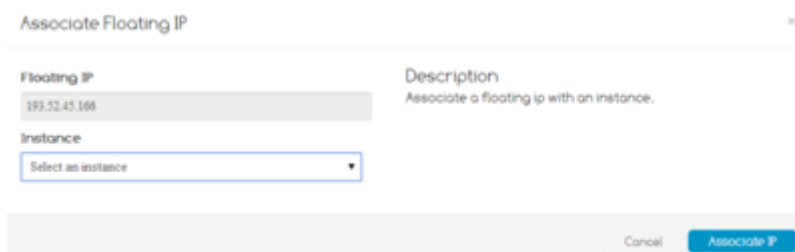


Figure 40: Associate IP

Once you have associated the IP to your instance, it is accessible through internet by SSH and ping. These are the only two protocols you allowed in your security group.

- Try to ping your instance from your personal computer

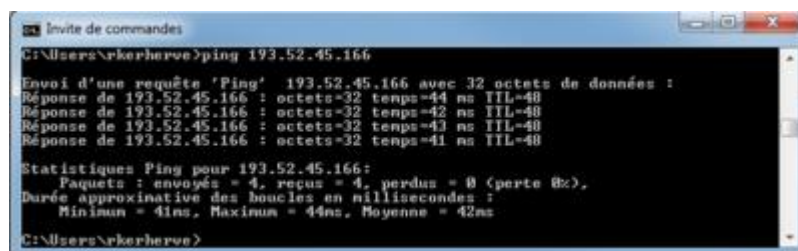


Figure 41: Ping

- Connect to your instance via SSH:

Do not forget to use the public key, you downloaded and used earlier for your instance.

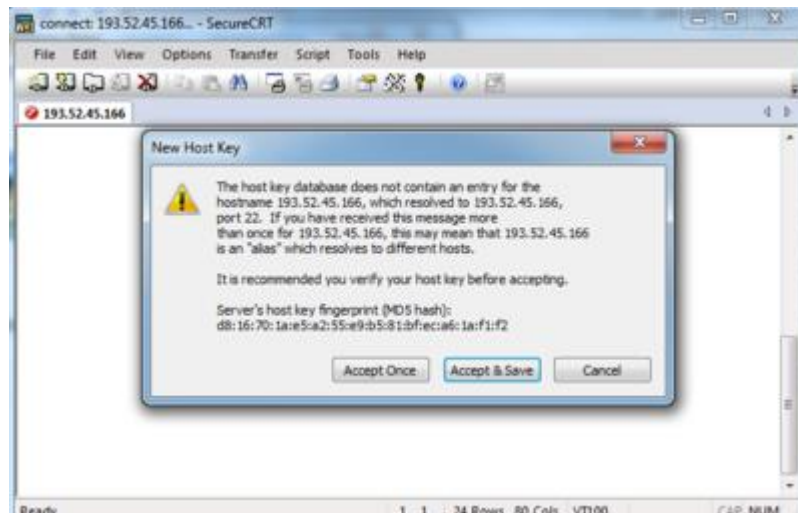


Figure 42: Connect via SSH

4.6.8 Tenant Life Cycle

This section describes the tenant life cycle and the actions that have to be done by the IOs. Actually, there are a few scenarios in which a definition of a tenant life cycle is needed and all of them are related to the identification of fraudulent use of resources. As the creation of a tenant involves no use of “real resources”, the problem comes with the use of the user inside this tenant. There are 3 scenarios:

- **Tenant with no use:** In this case we have a tenant with or without resources, but after a predefined period of time, there is no use of any resource. This period of time could be fixed as 3 months but could be redefined for each IO depending of the availability of resources or the misuse of them. Irrespective of whether this tenant uses resources or not, the system sends an email to the owner of the tenant in order to inform about the situation. Depending the decision of each IO this message could give details about the lifetime ineligibility if no activities are detected after a defined period (predefined with 3 months but IO could change it depending of their own management resources).

In case that there are user(s), the administrator will send an email to each user informing the about the situation in order to correct it or in other cases proceed to release those resources. If those resources continue not to be used, the admin will automatically release them after a period of time. If this situation applies to all users, the admin will proceed in the same way as in the previous one with no users.

- **Tenant with a user with a black email account:** This is a special situation in which a user has been created with an incorrect email address. After some period of time the IO administrator detects that the email corresponds to an email generator and proceeds to include it into the email black list in order that it cannot be used. The IO notifies the tenant owner of the situation in order to correct it and not to repeat it in the future. There is also the possibility to delete the tenant if the situation continues in the future.
- **Tenants with fraudulent users:** In this case the IO administrator detects that a user is fraudulently using resources. The procedure is to send an email to the users in order to inform them to resolve the problem, or the IO administrator could deactivate the resources. In the same way a notification is sent to the tenant owner informing of the situation in order to resolve it and not repeat it in the future. If malicious use continues, the IO will deactivate the tenant and release the resources associated to the tenant.

4.6.9 Traceability of Deployed Instances

This section deals with the IO's capability to identify who has allocated resources on the IO's infrastructure:

List all instances with their status, connected networks, private and public IPs using nova command `nova list --all-tenants`. As shown in Table 63, the output provides for each instance:

- Its unique ID
- Its display name, as given by the tenant
- Its status: **ACTIVE** (the instance is running), **SHUTOFF** (the instance is powered off) or **ERROR** (the instance has encountered a problem and requires administrative attention from the node owner).
- The networks that the instance is connected to, with the network name (as given by the tenant), the private IP address allocated to the instance in the network and the public IP address that the user has attached to the instance.

List all volumes with their status, size and attachment using nova command `nova volume-list --all-tenants`. As shown in Table 64, the output provides for each volume:

- Its unique ID
- Its status: **available** (if not attached to an instance), **in-use** (if attached to a tenant) or **error** (the volume has encountered a problem and requires administrative attention from the node owner).
- Its display name, as given by the tenant.
- Its size, in GB
- Its type
- the unique ID of the instance it is attached to, if its status is "in-use".

Provide details about an instance using the command `nova show instanceID`. As shown in Table 65: Output of `nova show 9389febd-bcc2-4e2f-83ba-c4c9356dd211`, where 9389febd-bcc2-4e2f-83ba-c4c9356dd211 is an instance ID the output provides more details than command `nova list --all-tenants`. Some information worth mentioning includes the following instance properties:

- **updated:** The last time the instance was changed (for example: restarted)
- **image:** The virtual machine image that the instance is based upon
- **OS-EXT-SRV-ATTR:hypervisor_hostname:** The compute node that the instance is running on
- **flavor:** The virtual machine flavor that the instance is based upon. The flavor determines the amount of resources (number of CPU cores, amount of memory, user disk size) allocated to the instance.
- **user_id** is the user ID as registered in the user database (see below the point about identification of an instance owner).
- **tenant_id** is the tenant ID as registered in the user database (see below the point about identification of an instance owner).

Provide details about a volume using the command `nova volume-show instanceID`. As shown in Table 66, the output provides more details than command `nova volume-list --all-tenants`. Some information worth mentioning includes:

- **os-vol-tenant-attr:tenant_id:** the Tenant ID.
- **attachments:** information about the volume attachment. In particular, it states where OpenStack believes that the volume is mounted on the instance (field `u'device'`). Although

this information is erroneous (the instance mounts the volume on the next free mounting point and discards the mount point provided by OpenStack), OpenStack refuses to mount a volume to an instance if it believes that the mount point is already in uses.

- **os-vol-host-attr:host:** name of the cinder host providing the volume.

Identification of an instance owner: An IO is not in charge of the user database. This is managed by the administrator of the IDM component. On the time of the release of this document, the IDM component is only deployed in Spain. As a consequence, User information is subject to Spanish law and according to this, user data cannot be disclosed except if requested by legal authorities.

ID	Name	Status	Network
9389febd-bcc2-4e2f-83ba-c4c9356dd211	MainServer	ACTIVE	MainNetwork=192.168.11.3, 194.28.122.35
84399fc8-65fa-4793-8212-d9a33c13727b	SecondServer	SHUTOFF	MainNetwork=192.168.11.3, 194.28.122.36
0ec146dd-4996-4a3e-935d-8f855443bd8e	MyServer	ERROR	TheNetwork=192.168.122.4.3, 194.28.122.45
9389febd-bcc2-4e2f-83ba-c4c9356dd211	MainServer	ACTIVE	MainNetwork=192.168.11.3, 194.28.122.35

Table 63: Output of nova list --all-tenants

ID	Status	Display name	Size	Volume type	Attached to
7c615192-4020-4521-9120-827946cec4db	in-use	MainVolume	10	None	9389febd-bcc2-4e2f-83ba-c4c9356dd211
b28ca07a-045c-4d07-bd61-ef651bb7e359	in-use	SecondVolume	50	None	9389febd-bcc2-4e2f-83ba-c4c9356dd211
4b1c8591-a8dc-47a7-ad5d-36e8c97aad51	available	MyVolume	20	None	
7c615192-4020-4521-9120-827946cec4db	in-use	MainVolume	10	None	9389febd-bcc2-4e2f-83ba-c4c9356dd211
b28ca07a-045c-4d07-bd61-ef651bb7e359	in-use	SecondVolume	50	None	9389febd-bcc2-4e2f-83ba-c4c9356dd211
4b1c8591-a8dc-47a7-ad5d-36e8c97aad51	available	MyVolume	20	None	
7c615192-4020-4521-9120-827946cec4db	in-use	MainVolume	10	None	9389febd-bcc2-4e2f-83ba-c4c9356dd211
b28ca07a-045c-4d07-bd61-ef651bb7e359	in-use	SecondVolume	50	None	9389febd-bcc2-4e2f-83ba-c4c9356dd211
4b1c8591-a8dc-47a7-ad5d-36e8c97aad51	available	MyVolume	20	None	
7c615192-4020-4521-9120-827946cec4db	in-use	MainVolume	10	None	9389febd-bcc2-4e2f-83ba-c4c9356dd211

Table 64: Output of nova volume-list --all-tenants

Property	Value
status	ACTIVE
updated	2015-02-06T23:03:42Z
OS-EXT-STS:task_state	none
OS-EXT-SRV-ATTR:host	node-15
key_name	my_keypair_1
image	Ubuntu12.04-server-x86_64 (a9f56be9-c993-4890-8116-35cbc5cbfa77)
MainNetwork network	192.168.11.3, 194.28.122.35
hostId	4f240acee7bce2d219d25f926f394f325108fec2cf6f5f647d8dc470
OS-EXT-STS:vm_state	stopped
OS-EXT-SRV-ATTR:instance_name	instance-0000017a
OS-EXT-SRV-ATTR:hypervisor_hostname	node-15.domain.tld
flavor	m1.small (2)
id	9389febd-bcc2-4e2f-83ba-c4c9356dd211
security_groups	[{'u'name': 'u'donald_sec'}]
user_id	donald-duck
name	MainServer
created	2015-01-29T14:33:27Z
tenant_id	00000000000000000000000000000005584
OS-DCF:diskConfig	MANUAL
metadata	{'u'region': 'u'Stockholm'}
accessIPv4	
accessIPv6	
progress	0
OS-EXT-STS:power_state	1
OS-EXT-AZ:availability_zone	nova
config_drive	

Table 65: Output of nova show 9389febd-bcc2-4e2f-83ba-c4c9356dd211 , where 9389febd-bcc2-4e2f-83ba-c4c9356dd211 is an instance ID

Property	Value
status	in-use
display_name	MainVolume
attachments	[[{'u'device': 'u'/dev/vdc', 'u'server_id': 'u'84399fc8-65fa-4793-8212-d9a33c13727a', 'u'id': 'u'7c615192-4020-4521-9120-827946cec4db', 'u'volume_id': 'u'7c615192-4020-4521-9120-827946cec4db'}]]
availability_zone	nova
bootable	false
created_at	2015-01-30T20:12:28.000000
snapshot_id	None
display_description	
os-vol-host-attr:host	node-5
volume_type	None
os-vol-tenant-attr:tenant_id	000000000000000000000000000000005584
source_volid	None
size	10
id	7c615192-4020-4521-9120-827946cec4db
metadata	{}

Table 66: output of nova volume-show 7c615192-4020-4521-9120-827946cec4db , where 7c615192-4020-4521-9120-827946cec4db is an instance ID

4.6.10 Local catalogue management

Introduction

Glancesync is a command line tool to solve the problem of the images synchronisation between nodes. It synchronises glance servers in different regions taking the base of a master node, which is in that case the Spain node. It was designed for FI-Core project, but it has been expanded to be useful for other users or projects and the first version was started to be used in XIFI.

Glancesync synchronises all the images, or a subset of them, with certain metadata owned by a tenant from a master node to each other node in the federation. This feature works out of the box without any configuration. It requires only the same set of environment variables, which are needed to contact the keystone server, in the same way that we need to contact the glance tool. It is also possible to set these parameters in a file instead of using environment variables. These parameters are:

- OS_USERNAME
- OS_PASSWORD
- OS_AUTH_URL
- OS_TENANT_NAME
- OS_REGION_NAME

Glancesync by default does not overwrite the content of existing images. If an image checksum in a node is different from the master node to be synchronised, a warning message is emitted. The user has the option to define a whitelist or a blacklist in order to force or ignore the overwriting of a specific image (optionally renaming the old one) including the checksums in a configuration file.

Glancesync has special support for AMI (Amazon Machine Image). Amazon images include a reference to a kernel image (AKI) and to a ramdisk image (ARI), but they are identified by their Universally Unique Identifier (UUID) on each node. Therefore, Glancesync has to set in the AMI image the kernel-id and ramdisk-id properties with their respectively UUID images in order to link with the AKI and ARI images.

This tool does not synchronise using UUID but names (i.e. an image has the same name in all regions, but not the same UUID). Using a UUID to synchronise is generally a bad idea, because some problems may arise with the restriction that a UUID must be unique. For example, a user in a region might upload a image with this UUID before the synchronisation, or a previous upload may end with an error but with the UUID created. If something similar to an UUID is required in order to identify an image, is better to use a metadata field that identifies universally the different images in different nodes.

Properties

Currently, images in glance are provided with a set of metadata properties that help us to identify, work and filter with them. Due to it, it is very important that the metadata service is up and running on each node. The following keys, together with the values to which they are specific, can be used with the property option for both the glance image-update and glance image-create commands. For example:

```
$ glance image-update IMG-UUID --property nid=142
```

The property keys that we are managing are the following:

Property keys			
Specific to	Key	Description	Supported values
	nid	Provide a unique identification of the corresponding GE image.	*See [52] for more details
All	type	Provide a classification of the images in order to allow filtering of them in the cloud portal.	*baseimages, images to be used to deploy new services. *fiware:apps, images from the FIWARE Application chapter. *fiware:data, images from the FIWARE Data chapter. *fiware:i2nd, images from the FIWARE Interfaz to the Network chapter. *fiware:iot, images from the FIWARE IoT chapter. *fiware:security, images from the FIWARE Security chapter. *fiware:userinterface, images from the FIWARE User Interface chapter. *fiware:utils, images prepare to be used with Sagitta (SDC Aware).
All	sdc_aware	Show us the possibility to be used	*True *No value

Property keys			
Specific to	Key	Description	Supported values
		this image by the Sagitta component (SDC Manager).	
All	kernel-id	The ID of an image stored in the Image Service that should be used as the kernel when booting an AMI-style image.	Valid image ID
All	ramdisk-id	The ID of image stored in the Image Service that should be used as the ramdisk when booting an AMI-style image.	Valid image ID

Table 67: Property keys

4.6.11 Managing Images

Images to add on the local catalog

Glancesync does not require an explicit images list to synchronise. Nevertheless, any public image upload by the tenant admin with NID or type metadata is automatically synchronised on all the FIWARE Lab nodes. The following table show us the set of images currently synchronised:

Property keys			
Image	Type	nid	sdc aware
wirecloud-img	fiware:apps	194	-
wstore-img	fiware:apps	512	-
iot-broker-R3.4	fiware:iot	476	-
eid-asbc-img	fiware:iot	696	-
eid-asvmlinuz		696	-
eid-asramdisk		696	-
repository-image-R3.2	fiware:apps	58	-
kernel_repository-image-R3.2		58	-
ramdisk_repository-image-R3.2		58	-
marketplace-ri-R2.3	fiware:apps	95	-
cep-r3.3.3-img	fiware:data	146	-
orion-psb-image-R3.4	fiware:data	344	-
orion-psb-image-R4.1	fiware:data	344	-

Property keys			
Image	Type	nid	sdc aware
ofnic-image-R2.3	fiware:i2nd	497	-
ofnic-image-R2.3-ramdisk		497	-
ofnic-image-R2.3-kernel		497	-
kurento-image-R5.0.33	fiware:data	855	-
kurento-image-R5.0.4	fiware:data	855	-
MiWi-POI server	fiware:userinterface	1170	-
augmented-reality-img	fiware:userinterface	1176	-
kernel_ub1204_3.2.0-29-amd64		1302	-
ramdisk_ub1204_3.2.0-29-amd64		1302	-
2d-ui-r3.3.3	fiware:userinterface	1304	-
3D-UI-XML3D	fiware:userinterface	1204	-
cloud-rendering-r3.3.3	fiware:userinterface	1286	-
VirtualCharacters	fiware:userinterface	1188	-
interface-designer-r3.3.3	fiware:userinterface	1292	-
2D3DCapture-3.3.3	fiware:userinterface	1257	-
GIS-3.3.3	fiware:userinterface	1215	-
RealVirtualInteractionGE-3.3.3	fiware:userinterface	1249	-
CentOS-6.3-x86_64	baseimages		-
CentOS-6.5-x64	baseimages		-
CentOS-7-x64	baseimages		-
Ubuntu Server 14.04.1 (x64)	baseimages		-
Ubuntu12.04-server-x86_64	baseimages		-
CentOS-6.3init	fiware:utils		True
CentOS-6.5init	fiware:utils		True
Ubuntu14.04init	fiware:utils		True
ubuntu12.04init	fiware:utils		True

Table 68: Set of images currently synchronised

Procedure to add the required images

At the moment, Glancesync is designed to run in the glance server of the master region, because it reads the images content directly from disk. This will be fixed in a feature version.

It's not necessary to install the software, after unzipping the package or running 'git clone' the tool is ready to work.

Glancesync works mainly as a front-end to the glance and keystone python tool, therefore they must be installed (note that in Essex OpenStack release, python-glanceclient was named as glance-client):

```
$ apt-get install python-glanceclient python-keystoneclient
```

First, you need the credentials to authenticate with the keystone server. You can put this credentials in a configuration file or set the following standard OpenStack environment variables: OS_USERNAME, OS_PASSWORD, OS_AUTH_URL, OS_TENANT_NAME, OS_REGION_NAME. The value of OS_REGION_NAME will be the master region (in FIWARE Lab this region is Spain).

If you prefer a configuration file, edit the file glancesync.conf. In section [main] the parameter 'master region' must be set with the region from which the images are synchronised. In section [master] the parameter 'credential' must include the following in this order: user, password, keystone_url, tenant. A difference in the configuration file, is that password must be encoded with base64.

After this, if you simply need to synchronise all the regions with the master region, run glancesync/sync.py.

The tool first obtains the list of regions, contacts with the master region to obtain its list of images, each one with its metadata, expands this metadata with the checksum of each image and finally prints the set of images to synchronise. Then it iterates through the list of regions. For each region, glancesync obtains the list of images with their metadata and checksums and compare with the results of master region. If an image is present in both regions but with different metadata fields nid, type or sdc_aware, it updates the image in the region with the values of the images in the master region. After this, it uploads sequentially (ordered by size in ascending order) the missing images to the region. If an operation with a region fails, for example the upload of an image, Glancesync passes to next region, due to if there is a problem uploading an image, there will be also problems uploading another one bigger.

It is possible that an image is present both in the region and in the master region, but with different content (i.e. the checksums are different). The default behaviour of Glancesync is only to print a warning (safety is a big concern with a synchronisation tool: it never should touch content without user knowledge). The user can specify a list of images that it is right to overwrite by setting a list of checksums (the old content ones) using parameter 'replace' in the section [master] of the configuration file. Another option is the parameter 'rename'; in this case the old image is not deleted but renamed (and its properties nid and type are renamed also). Both parameters can include the 'any' keyword. In this case the parameter 'dontupdate' works as a blacklist. The algorithm is:

- Is the checksum in dontupdate? print a warning only
- Is the checksum in rename? rename old image and upload the master region's image
- Is the checksum in replace? replace the old image with the master region's image
- Does the parameter 'rename' include the keyword 'any'? rename old image and upload the master region's image
- Does the parameter 'replace' include the keyword 'any'? replace the old image with the master region's image
- Otherwise: print a warning only

What images are synchronised?

The images to be synchronised from master region are selected by its metadata. Each selected image must has got a nid value and/or a type value. This choice is not arbitrary; all FIWARE images have at least one of these properties. A feature version of Glancesync will allow and arbitrary selection criteria.

The images also must be public and owned by the tenant.

It is possible to add manually, to the synchronisation set, additional images that:

- do not include the required metadata
- they are not public or are not owned by the tenant (but not both, because then they are not accessible)

Additional images are included appending their UUIDs to forcesyncs parameter at [master] section.

The regions list. Multitarget support

By default Glancesync synchronised the images from the master region to all the others regions whose glance server is registered in the same keystone server than the master region. However, another option is to pass the exact list of regions as parameters.

Additionally, Glancesync iterates sequentially with the region glance servers in the same order they are get from the keystone server. If the user prefer a different order, they can modify the parameter 'preferable_order' of section [main]. The value of this parameter does not need to include all the regions available, nor all of them has to exist at this moment. The algorithm works iterating through the list: if the region exists, it is append to the new ordered list and removed from the original list. At the end, the remaining regions are append to the new ordered list.

Post-image managing

The uploading of an image is done using the standard glance tool, for example:

```
$ glance add disk_format=raw name=image_name is_public=Yes container_format='bare' < image
```

If the infrastructure use Ceph as backend to Glance and Cinder, it is strongly recommended to use raw images. The kvm-img tool may be use to do the conversion from several formats to raw:

```
$ qemu-img convert image.qcow image.raw
```

It is also possible to upload qcow images to ceph and then convert to raw: <http://www.sebastien-han.fr/blog/2014/11/11/openstack-glance-import-images-and-convert-them-directly-in-ceph/>

If you need to manage an image, i.e modify it permanently, you can use guestfish. If you want to mount an image with read-write mode as root, use the following:

```
$ guestfish --rw -a <my_image.img>
```

You should then get a ><fs> prompt. First thing do to then is to type run, that will launch a virtual machine used to perform all the file manipulation. You can now list file systems with the list-file systems command.

```
><fs> run
```

```
><fs> list-file systems
```

```
/dev/vda1: ext4
```

```
/dev/vg_centosbase/lv_root: ext4
```

```
/dev/vg_centosbase/lv_swap: swap
```

Then mount your selected fs:

```
mount /dev/vda1 /
```


And then you can operate inside your image. When you're done, just type exit to go out of the guestfish tool. You can now use your modified image file.

4.6.12 Managing Blueprints

The management of Blueprint is based in the utilization of the PaaS Manager together with the SDC Manager. The corresponding recipes have to be incorporated into the SDC Recipes Catalogue in order to make use of these functionalities. These recipes currently are based in chef distribution programme. Nevertheless a new version of the SDC is being deployed in the Spain Node which allows the instantiation both Chef and Puppet recipes. In the following paragraphs we see the normal management operations over blueprint.

Create a blueprint template.

First of all you must create a blueprint template or take a predefined template previously defined in the catalogue. If we decided the first option, you should click on the "Blueprint Templates" button, then click on "Create New Template" (see screenshot below).

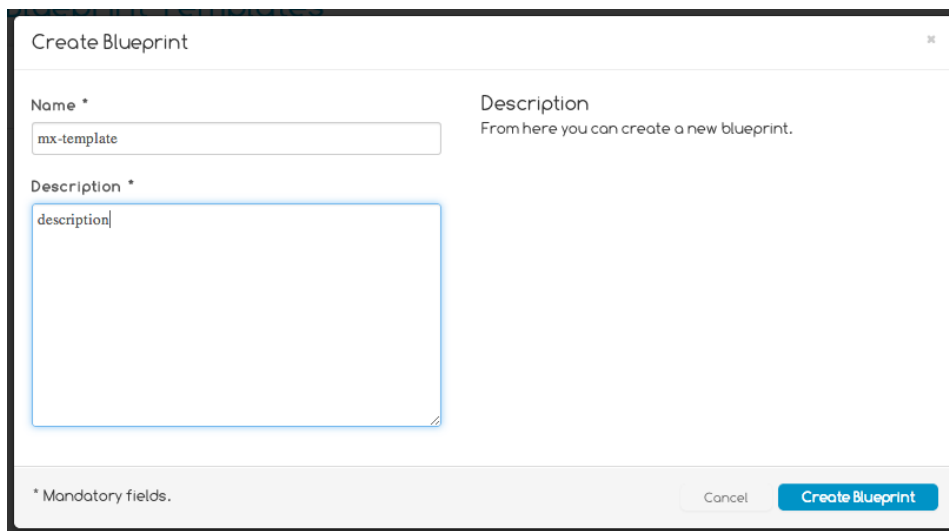


Figure 43: Create blueprint template

You have to complete the information related to the name of this template and a description in order to know afterward what the purpose of this template was. Then you should click on the “Create Blueprint” button to finalize the creation of the template.

If you decide to take one template from the catalogue, you should click on the "Blueprint Templates" button and then click on "Open Catalog". This shows you a list of predefined templates, take the one and click on the “Clone” button. This will create for you a new Blueprint template to work with.

Adding Tier(s) to your blueprint template

Secondly, if we want to add some Tiers to our blueprint template, we should click on the “Actions” button or over the right button of the mouse in order to go to the windows to add/edit/delete Tiers associated to this blueprint template (see the screen shown below).

Blueprint Templates

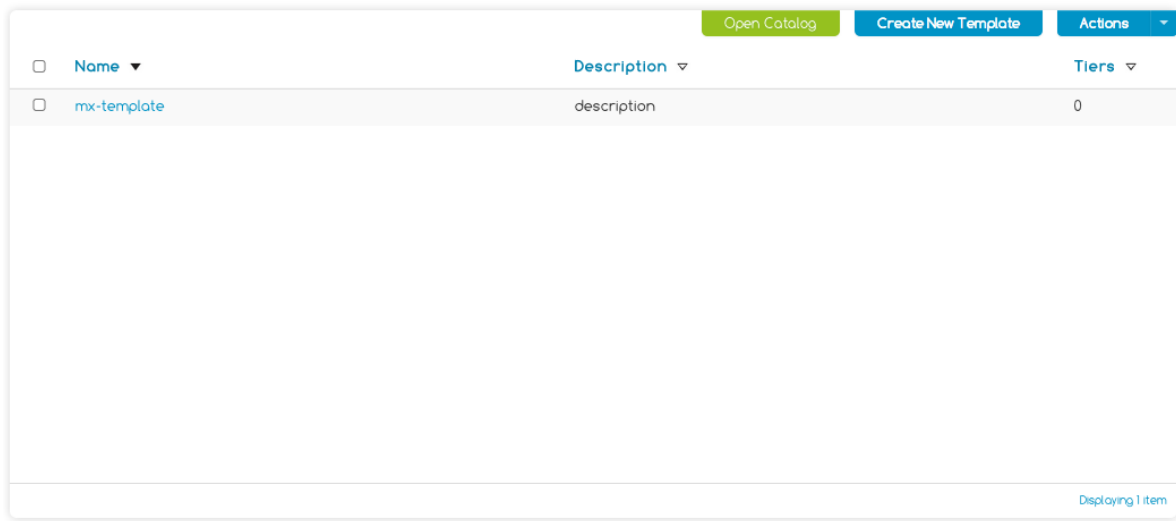
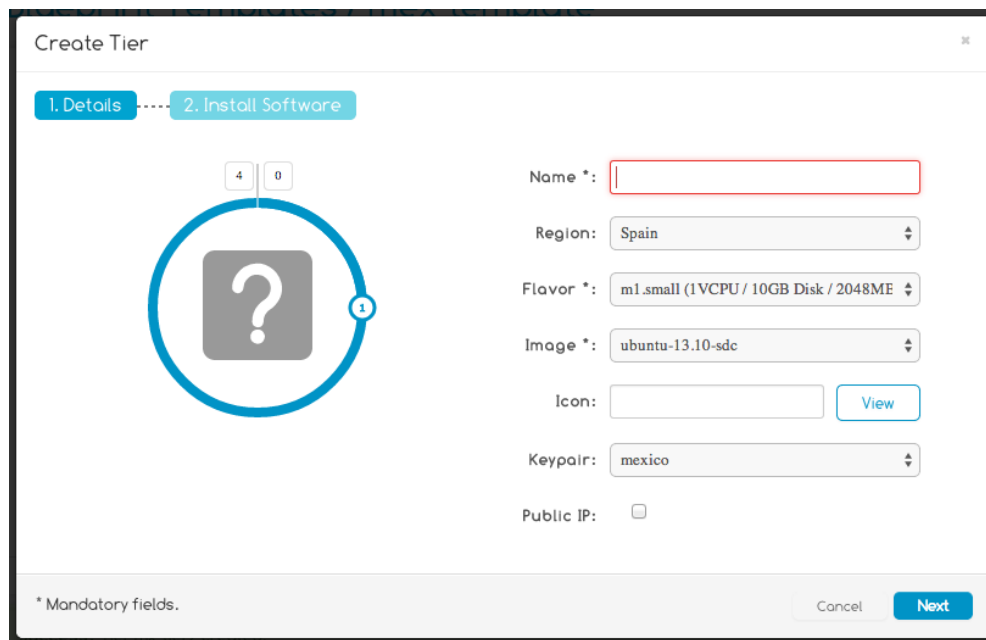


Figure 44: Adding tier(s) to a blueprint template

If we want to add a new tier, click on the “Add Tier” button to open a new window (see the screen shown below). It is a modal window in which you need to select the appropriate data. The marked attributes are mandatory. The selection of the Regions means that this Tier will be deployed in those regions. The data of flavours, Images and keypairs correspond to the data contained in the selected region (by default Spain Node). It is not mandatory to select an icon to represent the purpose of the tier but it is a good practice to know in a simple view what we are doing on this tier. Last but not least, you should indicate the minimum, maximum and current number of instances to be deployed using this template. This is made in the circle located to the left of the window (see the screen shown below).



Create Tier

1. Details 2. Install Software

4 0

Name *:

Region: Spain

Flavor *: m1.small (1VCPU / 10GB Disk / 2048ME)

Image *: ubuntu-13.10-sdc

Icon: View

Keypair: mexico

Public IP: ☐

* Mandatory fields.

Cancel Next

Figure 45: Create a tier

If we finish the introduction of data, we should click on the “Next” button, which moves to the next window in which we can select the corresponding recipe(s) to be installed on these instances. It corresponds to the list of Software Catalogue included in the SDC Manager. We can drag & drop from

the list of “Software in Catalogue” and translate it to the “Software in Tier” list (see screenshot below).

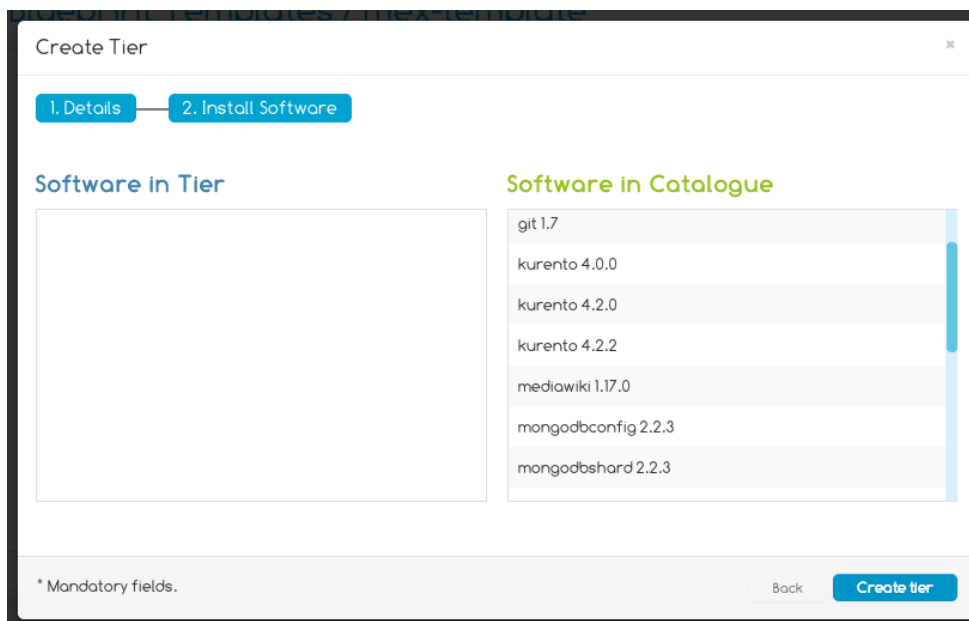


Figure 46: Adding software to a tier

To finish the template, click “Create Tier”.

Editing/Creating the software attributes.

In some cases, the software to be installed has an attribute or a group of attributes (ports, installation directory, and so on) that they are leaving by default. By contrast, if you want to change them, it is possible by clicking over the right button of the mouse over the selected software on the “Software in Tier” list (see the screen bellow) and click on “Edit Attributes”.

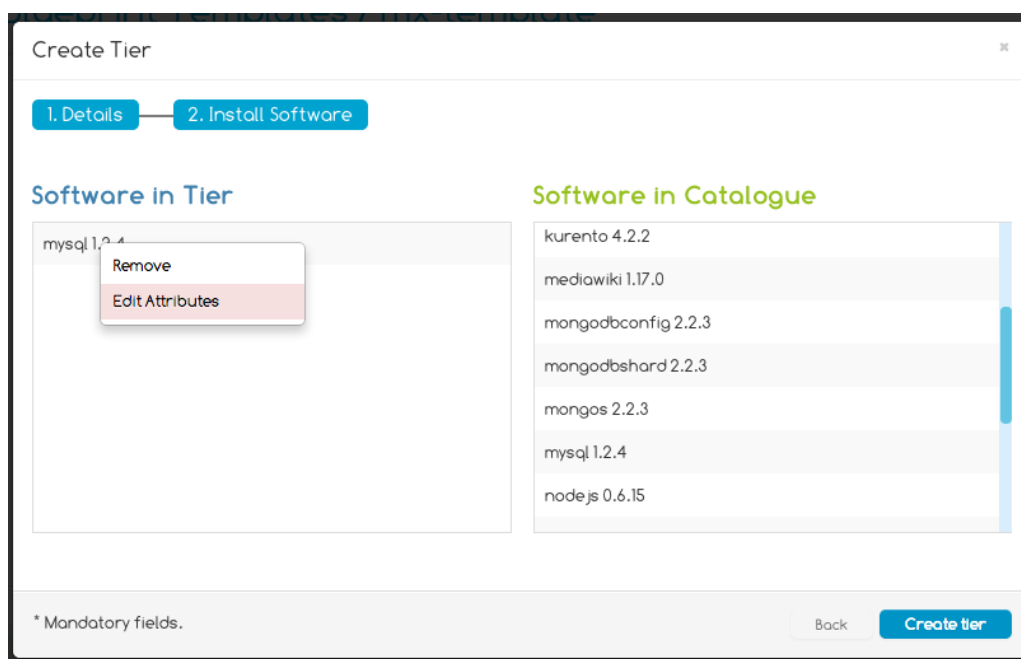


Figure 47: Selecting the menu to change the software attributes

This shows a modal window in which we can introduce the attributes to be used for the installation of the software (see the screen below). Please, refer to each product in order to know which the attributes for each case are.

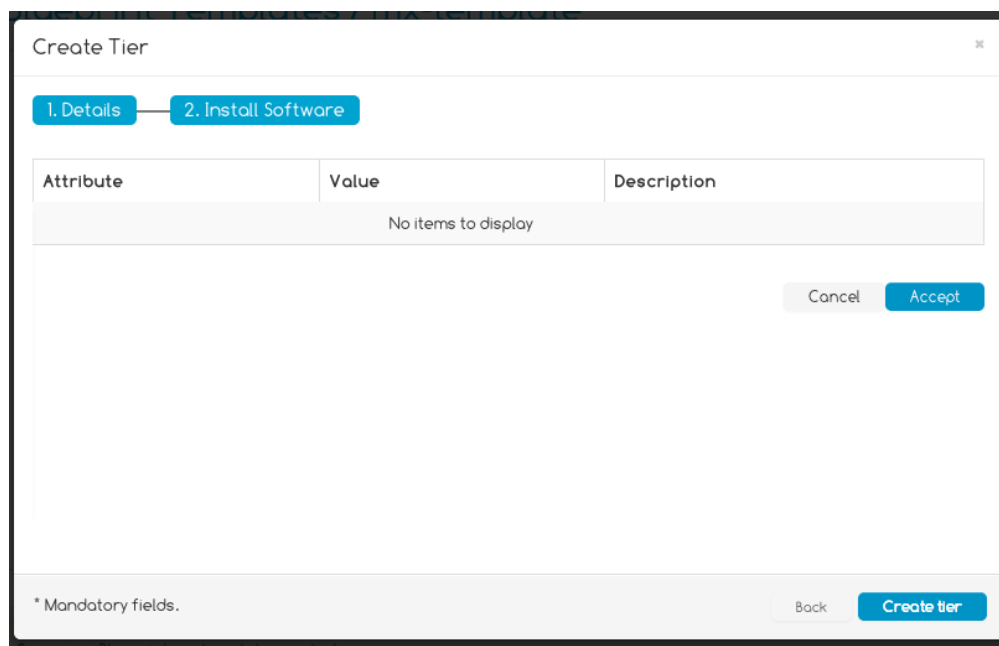


Figure 48: Editing the software attributes

Launching an instance.

After the creation of a blueprint template, if we want to launch it, we should click on the “Action” button, see Figure 44. It shows a menu with the option “Launch Instance” in which we click on in order to launch it. It shows a screen in which it asks us about the name of the blueprint instance and a brief description of it (see the image below).

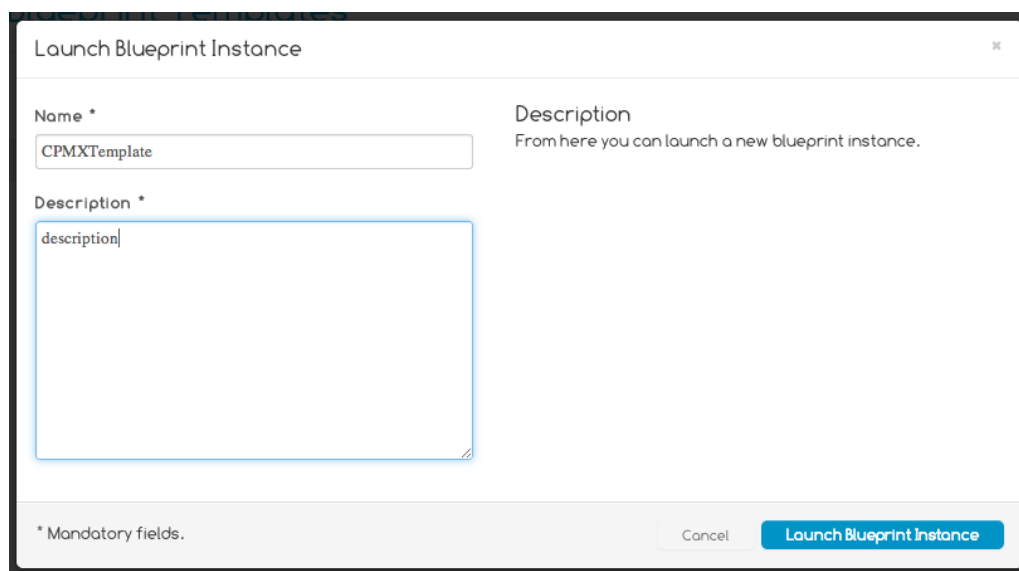


Figure 49: Launch a blueprint template

If we have finished the introduction of data we should click on “Launch Blueprint instance” in order to launch it. The screen changes to the main windows in which we can see the different states of the instantiation process (see the image below).

- Deploying the required infrastructure (deploying).
- Installing the selected products (installing)
- Installed (installed), which corresponds to the final status.

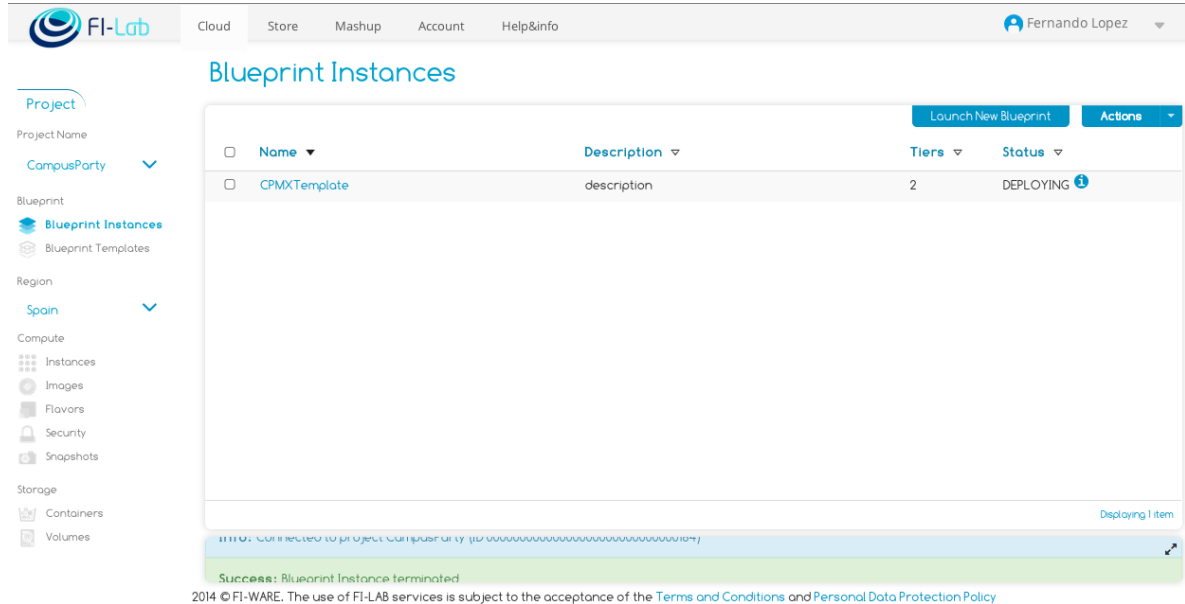


Figure 50: Blueprint instances

4.6.13 Use Case Handling

Information to get

When a Use Case wants to be deployed on a region, the person in charge of the Use Case needs to contact the node if the quota applied to the node does not fit the resource requirements. Indeed, most of the time, quota and flavors on a node are limited to prevent any user abuse. These can easily be changed or extended as best effort by the IOs, but it need to contact and provide to them a manual intervention.

Below is a list of basic information that an IO needs in order to adapt quota, flavor or configuration applied to a Tenant:

- Global Architecture presentation from the Use Case.
- Description of Instances:
 - Numbers of instances needed.
For each, give the dimension needed:
 - Memory.
 - Disk.
 - Number of processors.
 - Image type.
 - Etc.
 - Snapshot availability in case of migration from an existing server to XIFI.
- Network connectivity:
 - Number of network interfaces per instances and network.
 - Configuration of the interfaces wanted.
 - Ports to be open:
 - Management ports: Give the public IP needed in order to do the filtering.

- Service ports that will need to be open (Service provided by the Use Case).
- Login:
 - FI-Lab login ID (in order for the IO to add the ID to the granted list of users of the Use Case Tenant)
 - Snapshot login/pwd

4.6.14 Tenant Customization

The tenant customization is process that involves the definition of the default quotas. The resources available are divided in three sectors : Compute, Block Storage and Network .

Compute quota :

- instances
- cores
- ram
- floating_ips
- fixed_ips
- metadata_items
- injected_files
- injected_file_content_bytes
- injected_file_path_bytes
- key_pairs
- security_groups
- security_group_rules

Block Storage quota :

- gigabytes
- snapshots
- volumes

Network quota :

- floatingip
- port
- router
- subnet

The default quota provided by Openstack is :

Property	Value
metadata_items	128
injected_file_content_bytes	10240
ram	51200
floating_ips	10
key_pairs	100
instances	10

Property	Value
security_group_rules	20
injected_files	5
cores	20
fixed_ips	-1
injected_file_path_bytes	255
security_groups	10

Table 69: Openstack default quota

However the tenant's quota can be customized following the **FIWARE** policy.

The **default quota values** are a key aspect for a cloud infrastructure and the values of quotas have to be calculated in order to optimize the cloud resources. Using the command-line interface, the quotas can be managed for the OpenStack Compute service, the OpenStack Block Storage service, and the OpenStack Networking service. The default values can be modified in order to give more resources to a specific tenant to fulfill the requirements.

The federation is composed of many nodes with different features so the quotas has been defined in a common way, based on the considerations made for the quotas is necessary then set the flavours. The default flavours proposed by Openstack is the following:

ID	Name	Memory_ MB	Disk	Ephemeral	VCPUs	extra_specs
1	m1.tiny	512	1	0	1	{}
2	m1.small	2048	10	20	1	{}
3	m1.medium	4096	10	40	2	{}
4	m1.large	8192	10	80	4	{}
5	m1.xlarge	16384	10	160	8	{}

Table 70: Openstack default flavours

In order to provide the required flexibility for a heterogeneous set of platforms, several common flavours has been defined and each Infrastructure Owner has adopted in their environment. A **specific flavour for tenants with specific needs can be defined in order to keep a high level of flexibility**. An important thing to consider when the flavour is created should be the name format (e.i. m1.tiny). Openstack define each flavour with a specific name format and that convention has been preserved following the same format in every OpenStack distribution.

To **list the default quotas**, use the following commands (respectively for compute/network/block storage) :

- nova quota-defaults
- quantum quota-show
- cinder quota-show

To **update a quota for a particular tenant**, use the following commands :

- nova quota-update --<quotaName> <quotaValue> <tenantID>
- quantum quota-update --tenant_id <tenantID> --<quotaName> <quotaValue>
- cinder quota-update --<quotaName> <quotaValue> <tenantID>

Some examples :

- nova quota-update --ram 4096 <Tenant_ID>
- quantum quota-update --tenant_id <Tenant_ID> --network 3
- cinder quota-update --gigabytes 20 <Tenant_ID>

To **list the default quotas**, use the command:

- nova flavor-list

To **add a new flavor**:

- nova flavor-create --is-public <true/false> <flavor_name> <ID> <ram> <disk> <vcpu>
E.g:
nova flavor-create --is-public false Test-flavor auto 2048 0 2

Then **the created flavor must be associate to the tenant**, use the following command:

- nova flavor-access-add <flavor> <tenant_id>
E.g:
nova flavor-access-add a6abb411-9172-4641-974e-19d4ca044fdc
0000000000000000000000000000xxxx

4.6.15 Node Administration

Levels of administrative access (users, local admins, federation admins):

By joining the federation some administrative tasks are delegated to the master node, in particular identity management and authentication of OpenStack management actions. In consequence, some administrative commands (e.g. fetching user information or tenant information for all tenants) are no longer admitted for nodes but require collaboration with the federation (e.g. with the federation maintainer in case of maintenance procedures). This is a restriction imposed by federating, is a part of the operational level agreement between node and federation and causes a number of inconveniences including the need for continuous exchange of “service catalogues” and “authentication tokens” across the federation network infrastructure for almost all actions.

It could be disconcerting, but since the keystone component of OpenStack has been replaced by federation components (the keystone proxy and the IDM), some of the administration tasks that an IO needs to do on his node cannot be done anymore after joining the federation. The management of the user database is then removed from the tasks that an IO usually manages on his node and it is moved under the responsibility of another stakeholder: The Federation Maintainer (for roles c.f. section 5.2).

- Grants on the node (API, Cloud portal)

To administrate a node, three different types of access are available.

- CLI: This is the most common way to manage a node. This is a SSH connection on the controller that provide your CLI interface.

- OpenStack API: OpenStack provides a normalised API to manage its cloud. These APIs (Nova API, Cinder API, Quantum API, Swift API, Glance API) can be made accessible via the Internet or not. For the federation to work, these APIs must be openly accessible at least for cloud portal requests.
- Cloud Portal: It has access to the different OpenStack API (Nova, Cinder, Quantum, Swift, Glance) that are made accessible from the node. This access provide only a basic administration: user oriented.
- Roles

Here the level of administrative access, depending on the type of user, is defined.
- Users
 - Have basic access to tenants that he created.
 - Manage and administrate their tenants through cloud portal. It is the only administrative access they can have.
 - Manage and grant access of their own tenants for other users.
- Local admins
 - Is an IO, and is in charge of administration of its own node (IO)
 - Has basic access to his tenants through the cloud portal like other users has.
 - Has CLI access to the node and can administrate Nova, Glance, Swift, Quantum and Cinder
- Federation admins
 - Manage keystone proxy and IDM
 - Administrate users and tenants

Procedures:

- How to change the administrative level for a given user
 - Privilege on a tenant created by an IO

An IO can only manage a tenant he created. To manage a tenant, go to the Identity Manager and click on the arrow near the name of the user (Figure 51). Then choose "switch session" and click on the tenant you would like to modify roles for some users.

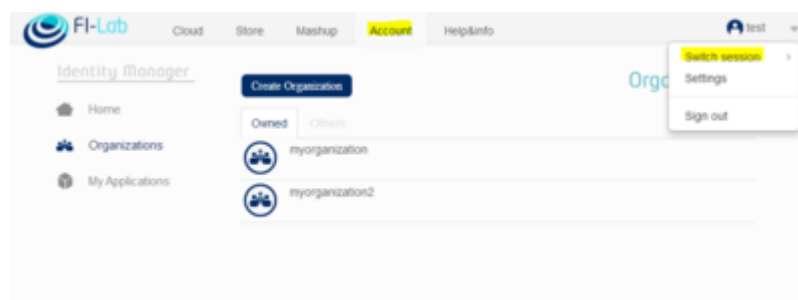


Figure 51: Manage tenant – select user

Click on members and then do the modification you would like to do, e.g. adding users or adding roles to a certain user.

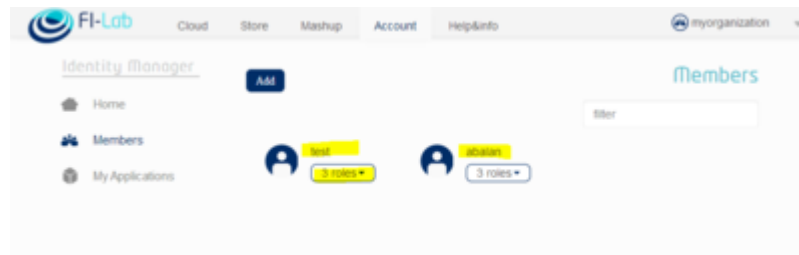


Figure 52: Modifications on a tenant

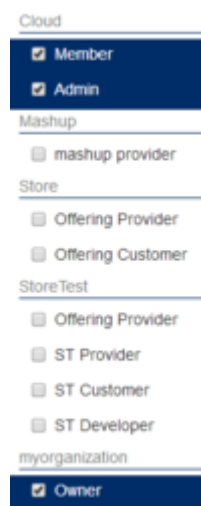


Figure 53: Modifications on a tenant II

- Privilege on a tenant not created by an IO

Please note that only persons in charge of IDM and the federation maintainers have the privilege to manage users on these tenants

- How to list tenants and users on a node:

The command below permits you to list all tenants in a given node

```
# sourceopenrc (to have nova rights)
# nova--os-region Lannion usage-list
```

See Table 71 for an example from Lannion node, 2014-07-15 - 2014-08-13.

Tenant ID	Instances	RAM MB-Hours	CPU Hours	Disk GB-Hours
00000000000000000000000000000009	2	53271.77	104.05	0.00
00000000000000000000000000000004	1	64407.91	125.80	0.00
000000000000000000000000000000356	2	5505024.48	2688.00	53760.00
0000000000000000000000000000002559	1	1376256.12	672.00	13440.00
0000000000000000000000000000002983	11	6720124.52	4993.41	54212.24
0000000000000000000000000000002988	3	2578811.02	1259.19	25183.70
0000000000000000000000000000003437	5	11010048.97	5376.00	107520.01
0000000000000000000000000000003449	9	33405.01	62.28	19.77
0000000000000000000000000000003478	3	24772610.17	8064.00	161280.01
0000000000000000000000000000003847	1	11010048.97	5376.00	107520.01
0000000000000000000000000000003851	1	344064.03	672.00	0.00
0000000000000000000000000000003940	1	195.70	0.10	1.91
0000000000000000000000000000003965	6	269624.50	388.78	918.84
0000000000000000000000000000003997	26	5758557.76	3816.22	0.00
0000000000000000000000000000004004	9	4930476.50	2411.06	58553.86
0000000000000000000000000000004012	1	344064.03	672.00	0.00
0000000000000000000000000000004019	1	344064.03	672.00	0.00
0000000000000000000000000000004098	1	2752512.24	1344.00	26880.00
0000000000000000000000000000004287	3	74.67	0.15	0.00
0000000000000000000000000000004291	1	1277297.91	623.68	12473.61
0000000000000000000000000000004351	2	595610.87	290.83	5816.51
38aec686f107485ebc1ca9763d96d958	5	6881280.60	3360.00	67200.01

Table 71: Lannion usage list

The command below permits you to list all VM created on your node as well as the name of the user who created it. Table 72 shows an example result.

```
# nova-manage vm list
```

instance	node	type	state	launched	image	kernel	ramdisk	project	user	zone	index
LeCloudC estLaVie	node-3	m1.s mall	active	22/04/2014	760d4409- 731c-4009- b368- 4a7ad78d83 35	38aec686f 107485eb c1ca9763d 96d958	f7b2f1315c4a47b284aa142fb0728d43			None	0
CEP- PTRAK	node-3	m1.m edium	active	21/05/2014	32f0120d-7533-4d3f-a7c4-0e492b93b740			3437	ptrak- syn	None	0
KURENT O2	node-1	m1.m edium	active	21/05/2014	25c3b46b-a91c-4bfb-8fd2-dc3d46858e57			3437	ptrak- syn	None	0
Fire2FIPP P01	node-5	fire2fi ppp	active	27/05/2014	dd5859f2-fbfa-4d12-9eac-2ec306685a75			3478	smorant	None	0
Fire2FIPP P02	node-5	fire2fi ppp	active	13/06/2014	b6d7fe2c-ee42-4e80-9978-d01a35f32d21			3478	smorant	None	0
Connecte dTV-v1	node-8	fi- cnt2- ctv	active	29/08/2014	88df7e0b-3cc6-4620-9e19-e76b832efb66			4004	smorant	None	0

Table 72: VM list

4.6.16 Openstack Release Upgrade

Openstack Release Roadmap

Starting with the Diablo release, the OpenStack release cycle abandoned its 3-month time-based cycle for a coordinated 6-month release cycle with frequent development milestones. You can see the development release schedule at https://wiki.openstack.org/wiki/Current_release_schedule

The Release Cycle is made of four major stages that will be described in this document.

Planning (Design, Discuss and Target)

The Planning stage is at the start of a cycle, just after the previous release, when we take a step back and focus on what we want to do for the next one. We discuss it with our peers to get their feedback and comments, in most cases proposing a spec document that precisely describes how we want to do it. It usually lasts 4 weeks, with the Design Summit on the third week.

Contributors may propose new specs at any moment in the cycle, not just during the Planning stage. However doing so during the planning stage is preferred, so that contributors can benefit from Design Summit discussion and PTLs can include those features into their cycle roadmap.

Once the spec is approved by the corresponding project drivers, implementation is tracked in a blueprint, where a priority is set and a target milestone is defined, communicating when in the cycle the feature is likely to land.

Implementation (Milestone iterations)

The Implementation stage is when we actually write the code (or produce the documentation...) corresponding to those blueprints. It is split into a number of milestone iterations.

Once your work is deemed ready to be proposed for merging into the master branch, it should be pushed to our Gerrit review system for public review. Note that in order to be fully reviewed in time for a milestone, the change should be proposed in the weeks before the milestone publication date.

At the last development milestone we apply three freezes: FeatureFreeze, DepFreeze and StringFreeze, in order to stop accepting new features and disruptive changes and concentrate on stabilization, packaging and translation.

Pre-release (Release Candidates)

After the last milestone and until the final release, we turn most of our attention to testing the result of all the development effort and to fixing release-critical bugs. So:

- Kick the tires on it like never before, and file bugs about everything you find, be it small bugs, big bugs, misfeatures, or missing features. Anything. It's better to have problems be known and understood than ignored.
- Help prioritize bugs (see BugTriage).
- Help write documentation. Our software can be as cool as it wants, but if our docs aren't any good, people won't be able to use it anyway.
- Last, but certainly not least, fix as many bugs as you can.
- See Bugs for a more detailed view.

Release day

On release day, the last published Release Candidate of each integrated project is collected and the result is published collectively as the OpenStack release for this cycle.

XIFI nodes update plans

In its first phase XIFI was based on the **Grizzly** release of Openstack. After December 2014 the XIFI nodes have started an upgrade procedure in order to implement the **newer release of Openstack Icehouse**.

Depending of the status of the nodes, and the use of individual resources from FIWARE lab developers the IOs had two upgrade paths to choose from:

1. Rebuilding of the node
Mainly concerning the nodes having little to no virtual machines from FIWARE lab users, this upgrade path included the rebuilding from scratch of the whole infrastructure. During the rebuilding period the node was removed from the Cloud Portal so no users might try to create an instance. Moreover the node was removed from the infographic and status pages so it would not give a wrong status.

2. Migration of the node

This upgrade path concerned nodes that have active instances from users deemed important enough so they should not be erased. In this case rebuilding the node is not possible. The nodes following this path had to procure enough resources to build another “copy” of their infrastructure in order to serve the migration process. After the new infrastructure was implemented it was federated in the cloud portal as a second node of the same name (eg for Trento, Trento2).

FIWARE Generic Enabler Support

The FIWARE GE's have been checked by the Spanish node and found to be compatible with Openstack versions up to Icehouse. Currently there is no node running Juno. The Spanish node is migrating part of its infrastructure to Juno and will test the compatibility of the FIWARE GE's as long as the infrastructure is operational.

Supported release by ITBox tool

The forthcoming release of ITBox tool (version 2.0) will be the one to support the deployment of nodes using the Icehouse rerelease of Openstack. The latest version of ITBox will be released by the end of February 2015 and will be available from: <https://github.com/SmartInfrastructures?query=fuel->

Node individual choice

The migration plan of all the nodes, that was defined on December, 2014, is listed in the following table.

Node	#VMs	Icehouse migration		Juno migration		Comment	Migration status
		Beginning	To be finalised by	Beginning	To be finalised by		
Gent	3	NA	NA	Feb 2015	Feb/Mar 2015	We plan to go straight to Juno to use IPv6. We would like to follow the consortium steps.	We have a Juno deployment and currently doing test with the components.
Poznan	41	january 2015	january 2015	not planned	not planned	Icehouse ready	

Node	#VMs	Icehouse migration		Juno migration		Comment	Migration status
		Beginning	To be finalised by	Beginning	To be finalised by		
C4I	4	February 2015	February 2015	not planned	not planned	We plan to migrate to Icehouse first. But if Juno is supported by the FIWARE GEs, we would prefer to migrate directly on it. We waiting for a final decision before we start the migration.	
Stockholm	TBD	NA	NA	Feb 2015	Feb/Mar 2015	Our plan is to go to the latest stable release (Juno).	
Budapest	117 active out of 272 total	beginning January 2015	Estimated end of February 2015	not planned, only if Juno is supported by the portal	not planned, only if Juno is supported by the portal	Our plan: we have 2 blades where we can install new version (icehouse) to test and operate in parallel. When new system passes all tests and is properly federated, migration begins. Since as far as we know Juno is not supported by the cloud portal, we are not planning to deploy it yet.	IceHouse environment has already been deployed in separate blade servers. The installation of monitoring components are on going. Experimentation with Grizzly-IceHouse migration is also on going.

Node	#VMs	Icehouse migration		Juno migration		Comment	Migration status
		Beginning	To be finalised by	Beginning	To be finalised by		
NITOS	0	January 2015	January 2015	NA	NA	We use Ceph as a shared storage. This framework supports only raw image formats. As a result the FIWARE images should be converted.	The migration to Icehouse is concluded. We are in the process of completing Monitoring (expecting tests from Attilio) and we requested the upload of the images from Fernando
Karlskrona	20	March	EoF April	April	June	Planning to go to IceHouse, but if Juno becomes supported will go there instead of IceHouse.	Upgrading test nodes for IceHouse and Juno Tests.
Zurich		January 2015	January 2015			No specific plans for Juno at present. Want to get Icehouse stable and operational before making such plans. Do have interest in exploring IPv6 capabilities which could drive this move, however.	
PiraeusU	0	January 2015	(Estimated) End of January 2015	NA	NA	Migration to Icehouse has been successfully finalized @ 21/01/2015	Icehouse successful setup and currently is running on PiraeusU node

Node	#VMs	Icehouse migration		Juno migration		Comment	Migration status
		Beginning	To be finalised by	Beginning	To be finalised by		
PiraeusN	0	01/12/2014	18/12/2014	NA	NA	Juno was set out of the question recently in some email discussion. For now there are no plans for Juno.	Currently running on IceHouse successfully.
Prague	TBD	NA	NA	2015-01-05	est. Q2 2015	There were compatibility issues talks according to the migration.	Working on it.
Trento	35	2014-12-17	Depends on the federation process	NA	NA	Before Xmas Holydays we plan to have a Parallel Production Environment based on Icehouse, but probably not Federated. At the moment the idea is to leave to the tenant the task to migrate VMs and nets. Waiting for a more transparent solution	Federation started

Node	#VMs	Icehouse migration		Juno migration		Comment	Migration status
		Beginning	To be finalised by	Beginning	To be finalised by		
Berlin	35					We did not finally decided to which OpenStack release to migrate. If Juno is supported by the FIWARE GEs, we would prefer to migrate directly to Juno. We will wait for a final decision of the consortium, before we start the migration activities.	
Waterford	46 (ish)			2015-02-12	2015-02-12 (+2 week)	At Waterford, if an upgrade OpenStack cycle is to be administrated, we would like to move to the latest stable (Juno) if possible. We have applied for a block an additional block of public IPs.	
Lannion	40-70	NA	NA	2015-03-01	2015-06-01	In our case, seing the projects timing plan (XIFI+ FI-Core), we d like to limit the number of upgrades, then jump straight to Juno.	

Node	#VMs	Icehouse migration		Juno migration		Comment	Migration status
		Beginning	To be finalised by	Beginning	To be finalised by		
Spain	2701	NA	NA	01-12-2014	31-01-2015	At Spain, we would not move to IceHouse due to we want to work with the last version of OpenStack due to our contribution to the community.	

Table 73: Node Openstack release upgrade

5 MAINTENANCE PROCESS

The Maintenance process grouping is dedicated to the execution of proactive and reactive maintenance activities to ensure that services provided to developers are continuously available and conform to SLA or QoS performance levels. As part of a continuous maintenance process it performs continuous resource status and performance monitoring to proactively detect possible failures. It collects performance data and analyses them to identify potential problems and resolve them without impact to the developer. It reacts on trouble reports from developers, informs the developers of the trouble status, and ensures restoration and repair.

The maintenance process involves the production environment and optionally the pre-production of the node. The implementation of a pre-production is under responsibility of a node owner and depends on internal resources.

The maintenance process interfaces with developers through the support and readiness grouping process and, in particular, through the help-desk. This section details on the core maintenance process in terms of procedures, stakeholders and roles involved in this process, and the components and sub-systems subject to the maintenance process.

First, relevant stakeholders are identified with a particular view on their role and obligations in the maintenance process. Next, the components that are subject to the maintenance process are identified. Finally the maintenance process is outlined further detailing how the roles involved are acting on the maintenance subject.

5.1 Relation to the eTOM Framework Objectives

D5.3 outlined the relationship between procedures of the e-business Telecoms Operation Map (eTOM) framework defined by the Tele Management Forum and the XIFI operations and maintenance processes. It was stated there that the eTOM framework is focusing on a customer perspective and thus has its main significance in the domains regarding quality assurance and service level agreements while XIFI is focusing on the operations and maintenance processes applied to platforms and node infrastructures.

Although there is a strong overlap and dependency between the two perspectives, it was decided that there is no urgent need to emphasize further on a potential alignment between the two perspectives. The topic thus is considered being of limited relevance for this deliverable and thus is not further elaborated upon.

Please note that some effort has been spent on the description of dedicated maintenance tasks which might be a good foundation if the relation to the eTOM framework will potentially come into focus again.

5.2 Stakeholders

Deliverable D5.1 has identified the following stakeholders relevant in the scope of the maintenance process grouping:

- The federation manager acting as a 'federation maintenance supervisor'.
A federation manager is considered the first point of contact in case a maintenance process involves more than a single XIFI node. The particular process must define if the federation manager has to be informed about the particular process being initiated, has to be involved as a mediator among different nodes or with the first level support of the federation, or has to be actively involved to coordinate independent node actions to avoid federation down-times.
- The node manager acting as a 'node maintenance supervisor'.

A node manager is responsible for the maintenance of a single node following both local maintenance processes and federation maintenance processes. In case a local maintenance process may affect operations of the federation, the node manager also responsible for interacting with the federation manager. In case of an incident, node managers play an active role in federation management to minimize the federation-wide impact of a local incident.

- End users acting as 'developers'.

Users may be involved in maintenance processes in various ways. Users may need to be informed about scheduled or unscheduled maintenance processes as soon as their use of the XIFI federation is affected. They may be involved to support the maintenance process by postponing activities, moving their activities across the federation to free up a particular node, or to backup and restore their results to bridge a certain foreseeable downtime of the federation, of a node or of a particular service. Users may also cause a maintenance process to initiate either through interaction with the first level support or through causing an incident.

For the maintenance process defined below, the roles these stakeholders can take had to be refined compared to what had been defined in D5.1 in a general sense (and reminded above). This was necessary for defining the maintenance process, in order to distinguish between actors (implementing the maintenance process), supporters (actively contributing to particular aspects of the maintenance process) and maintainers (taking responsibility for dedicated subjects).

The **federation maintainer** takes responsibility for coordinating the maintenance process in case a procedure involves multiple infrastructure nodes. This role is responsible for identifying and coordinating with infrastructure node's maintainers based on availability, capacity and technical requirements for a particular maintenance procedure and involvement of nodes in this procedure. This role requires mapping to an identity and must obtain suitable access rights to the federation.

This role

- can initiate the implementation of a maintenance procedure;
- responds to the request to implement a maintenance procedure;
- can delegate a received request to implement a maintenance procedure.

The **federation maintenance contact** is the single point of contact in case a maintenance procedure needs to request support from the federation maintainer to implement a maintenance process. The federation maintenance contact is the first point of contact for external requests (e.g. made through the help-desk) and for new nodes. It should be the first point of contact for all other roles to unburden the federation maintainer from handling misdirected requests.

This role

- can initiate the implementation of a maintenance procedure;
- can delegate a received request to implement a maintenance procedure.

Federation Maintenance Contact
Federico Facca (CreateNet), Miguel Carrillo Pacheco (TID)

Table 74: Federation Maintenance Contact

The **infrastructure maintainer** takes responsibility for coordinating the maintenance process for a single infrastructure. This role coordinates with the federation maintainer in case a procedure requires support from remote infrastructure nodes. This role requires mapping to an identity and must obtain suitable access rights to the infrastructure maintained.

This role

- can initiate the implementation of a maintenance procedure. In particular, this role is in charge of
 - installation of validated subsystems releases by default to the pre-production environment (or in the production environment),
 - move of a sub-system from the pre-production to the production after internal coordination with the test owner.
- responds to the request to implement a maintenance procedure;
- can delegate a received request to implement a maintenance procedure;
- can escalate requests to implement a maintenance procedure to the federation.

The **infrastructure maintenance contact** is the single point of contact in case a maintenance procedure needs to request support from a particular infrastructure node. This role is equivalent to the federation maintenance contact.

This role

- can initiate the implementation of a maintenance procedure;
- can delegate a received request to implement a maintenance procedure.

Infrastructure	Contact
Berlin	xifi-support@fokus.fraunhofer.de
Brittany	support-lannion@imaginlab.fr
Spain	fi-admin@rediris.es
Trento	support-xifi@trentinonetwork.it
Waterford	jtynan@tssg.org
IMINDS	support-xifi@intec.ugent.be
ZHAW	murp@zhaw.ch
PSNC	xifi-psnc@lists.man.poznan.pl
Neuropublic	xifi-support@neuropublic.com
CESNET	xifi-support@cesnet.cz
UPRC	iwave@unipi.gr
Com4Innov	support@com4innov.com
ACREO Swedish ICT	testbed@acreo.se
WIGNER	xifi-support@wigner.mta.hu
UTH	nitlab@inf.uth.gr
BTH	xifi-helpdesk@bth.se
IntelliCloud (Crete)	xifi-support@intelligence.tuc.gr
InfoTech (Mexico)	internetdelfuturo@infotec.com.mx
University of Messina (IT)	xifi-helpdesk@unime.it
Wroclaw University of Technology (Poland)	Not yet available

Table 75: Infrastructure maintenance contact

The **test owner** is a particular role that is responsible for implementing a test case in the pre-production environment (or eventually, under responsibility of the node owner, in the production environment) of a node. In the scope of the maintenance process this role is responsible to implement a procedure that aims at verifying correctness of a sub-system in the course of proactively monitoring the infrastructure node. He interacts with the testbed maintainer (see below) to allocate resources necessary to conduct this test. The testbed is involved in the maintenance process in case a sub-system is suspect to cause problems prior to requesting a maintenance procedure involving the node or escalating to the federation. This role requires mapping to an identity and must obtain suitable access rights to the infrastructure hosting the test case.

This role

- can initiate the implementation of a maintenance procedure;
- responds to the request to implement a maintenance procedure;
- can delegate a received request to implement a maintenance procedure to a sub-system maintainer or to a component maintainer;
- can escalate requests to implement a maintenance procedure to the infrastructure node.

The **testbed maintainer** complements the infrastructure maintainer role for nodes that provide developer testbed services. Role responsibilities are the same regarding the testbed infrastructure as they are for the infrastructure node and the infrastructure maintainer. This role requires mapping to an identity and must obtain suitable access rights to the testbed maintained.

This role

- can initiate the implementation of a maintenance procedure;
- responds to the request to implement a maintenance procedure;
- can delegate a received request to implement a maintenance procedure.

The **testbed maintenance contact** complements the infrastructure maintainer contact role for nodes that provide developer testbed services. Role responsibilities are the same regarding the testbed infrastructure as they are for the infrastructure node and the infrastructure maintenance contact.

This role

- can initiate the implementation of a maintenance procedure;
- can delegate a received request to implement a maintenance procedure.

The **component owner** is responsible for a particular component and usually is identical with the component developer. In the scope of a maintenance process this role is responsible for implementing modification requests for a particular component.

This role

- responds to the request to implement a maintenance procedure.

The **component maintainer** is responsible for implementing maintenance procedures on a particular component, which may or may not involve the component owner if modifications need to be applied to that component. This role requires mapping to an identity and must obtain suitable access rights to the infrastructure hosting the component under maintenance.

This role

- can initiate the implementation of a maintenance procedure;
- can delegate a received request to implement a maintenance procedure.

The **sub-system maintainer** is responsible for implementing a maintenance procedure on a particular sub-system, which may or may not involve further component maintainers and other sub-system maintainers if modifications need to be applied to that sub-system(s). This role requires mapping to an

identity and must obtain suitable access rights to the infrastructure(s) hosting the sub-system under maintenance.

This role

- can initiate the implementation of a maintenance procedure;
- can delegate a received request to implement a maintenance procedure
- responds to the request to implement a maintenance procedure.
- coordinates component owners, component maintainers, and tests owners in order to check consistency of a particular software sub-system release. In case of the testbed, this role produces a validated sub-system.

The **sub-system maintenance contact** is the single point of contact in case a maintenance procedure needs to be applied to a particular sub-system. The sub-system maintenance contact is the first point of contact for external requests (e.g. made through the help-desk) or originating from other maintainers.

This role

- can initiate the implementation of a maintenance procedure;
- can delegate a received request to implement a maintenance procedure;
- can escalate requests to implement a maintenance procedure to infrastructure nodes and to the federation

The following figure details a realistic life-cycle of an issue report causing a maintenance action. The example detailed here assumes that a FI developer detected an issue with one of the sub-systems and submits a problem report to the federation, since the developer cannot identify particular nodes potentially responsible for resolving that issue. In a procedural interaction between federation, involved node infrastructures, testbed, sub-system and component maintainers the issue then is resolved. Details of this procedure are given next.

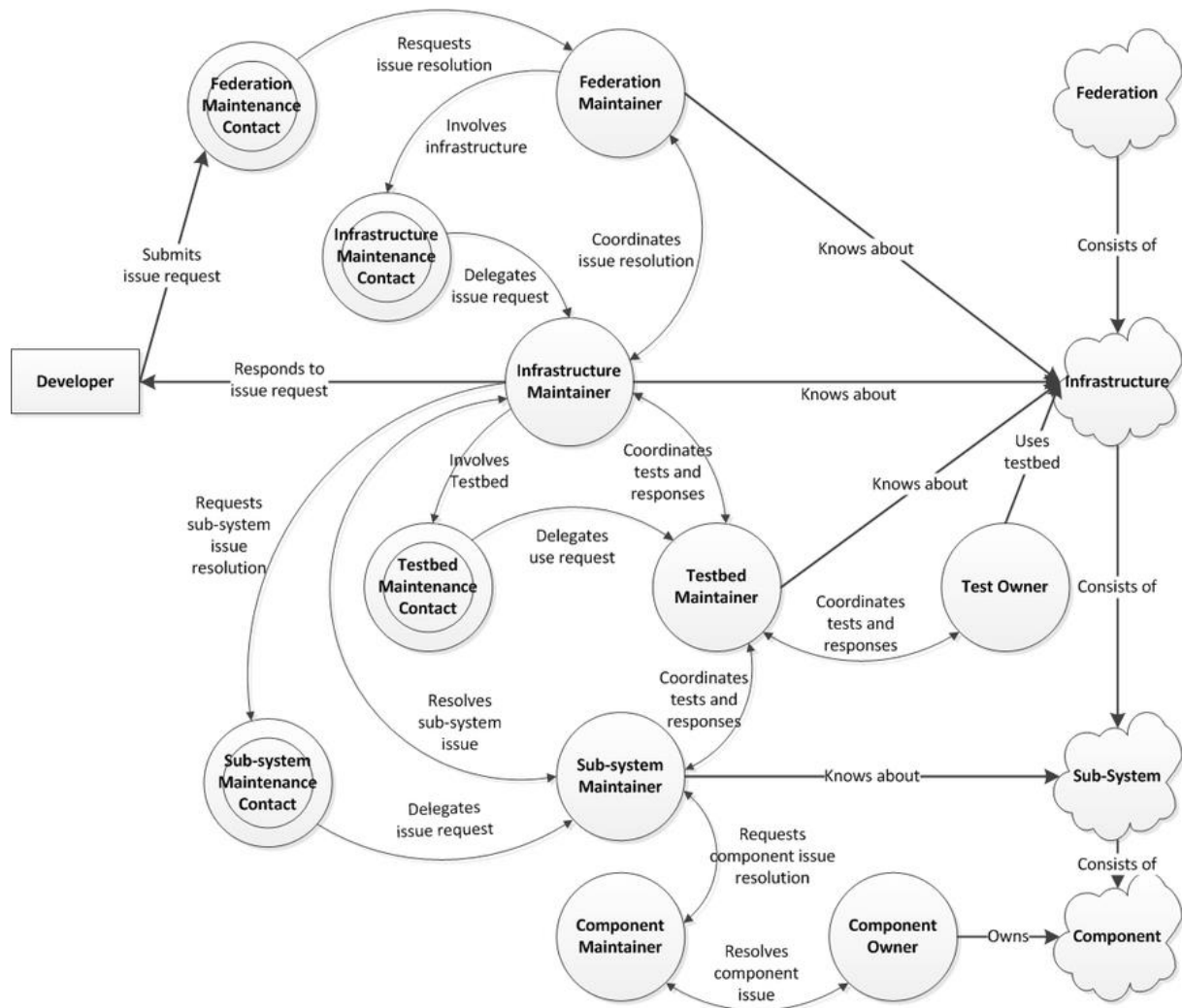


Figure 54: Sample Maintenance Stakeholder Interaction (Developer initiated issue request on a sub-system issue)

1. **Developer submits an issue request to the federation maintenance contact** -- This usually is done by creating a JIRA ticket only describing the problem and the sub-system(s) in scope.
2. **The federation maintenance contact forwards the issue request to the federation maintainer** -- The federation maintainer are one or more persons able to decide if the issue is formally complete, can be accepted and delegated to the infrastructure and sub-system maintainers. In the course of this process the issue is analysed in order to identify the correct targets (i.e. which infrastructure to involve and which sub-system(s) to address).
3. **The federation maintainer involves infrastructure(s)** -- This is usually done by creating a JIRA ticket to one or more infrastructures describing which sub-system has to be evaluated on the infrastructure addressed. The federation maintainer may take a coordinating role or may delegate the coordinating role to a particular infrastructure maintainer. The specific approach determines who (i.e. the federation maintainer or the lead infrastructure maintainer, or the ticket owner) will response to the issue request originated by the developer, and who will request involvement of further maintainers (e.g. sub-system and testbed maintainers) if necessary. Since the federation maintainer probably does not know about current responsibilities for the infrastructures to address, this involvement is done through the corresponding infrastructure maintenance contact(s).

4. **The infrastructure maintainer requests supportive actions** -- Depending on the coordinating role (in case multiple maintainers are involved) interaction with the federation maintainer may be required to contact (i.e., submit issue resolution requests to) other stakeholders. In general, it is recommendable not to involve too many stakeholders at a time for a single issue, but to favour bi-lateral interaction in order to keep administrative overhead (i.e. ticket management) at a reasonable level.
 1. **The infrastructure maintainer requests a sub-system issue resolution** -- This is usually done by submitting an issue resolution request to a sub-system maintainer. The sub-system maintenance contact here acts as a single point of contact for maintenance of one or more sub-systems. It delegates the issue request to sub-system maintainer responsible for the sub-system under consideration. The sub-system maintainer is knowledgeable regarding the interaction of sub-systems and the interaction of components of the sub-system under consideration. The sub-system maintainer therefore can decide and involve component maintainers in case the sub-system issue has been tracked down to the responsible component(s).
 2. **The infrastructure maintainer involves a testbed** -- The testbed is considered in the course of tracking down an issue and verifying the resolution. It is actively involved by the infrastructure maintainer in collaboration with the sub-system maintainer and the testbed maintainer. It is utilized by submitting requests for conducting tests to the testbed maintainer, which in turn delegates tests to test owners.
5. **The issue is considered as resolved** if all involved maintainers report and acknowledge the resolution of the issue in their particular scope. Maintainers having a coordinating role (e.g. the sub-system maintainer) need to judge upon the completeness and correctness of the resolution based on the joint reports. If there is a particular lead role (e.g. a single infrastructure maintainer as mentioned above), this stakeholder decides upon the maintenance procedure(s) to apply in order to resolve the issue federation wide. If not, the federation maintainer has to initiate a suitable maintenance procedure.
6. **A response to the issue request** is submitted to the originator of the issue request (i.e. the developer) by either the lead infrastructure maintainer or the federation maintainer detailing on the resolution applied, or on the state of the resolution if still ongoing.

It is obvious that the maintenance procedure in scope of this document is only a small part in the complex process described above. It has been described here with the purpose to clarify that it is important to know in which context a maintenance procedure is executed: it may be part of a problem resolution as described above involving many steps prior to the decision when and how to execute the maintenance procedure, or may be self-contained not involving any stakeholders except for issuing a notification to stakeholders that a maintenance procedure is executed right now.

5.3 Stakeholder Interaction through the Help-desk

5.3.1 Interaction of Maintainers and Developers

There can be cases where support requests from developers trigger the launch of a maintenance action. This could e.g. occur when a developer reports a problem which, after checking, points to an issue in a node or the federation which needs to be fixed through a maintenance task.

The requests received from developers are however not specific to maintenance. Developers simply report an issue and there is no difference in the way interactions take place between developers and helpdesk and maintainers. All such requests are received first and being checked by Level-1 Helpdesk. Only after that it will show whether Level-1 Helpdesk or a node can answer to the request or whether the request triggers a maintenance task. Therefore there is no need for a specific process for the

interaction between developers and maintenance. Instead, the regular process for providing support to FI-Developers is applied. This process is described in section 6.

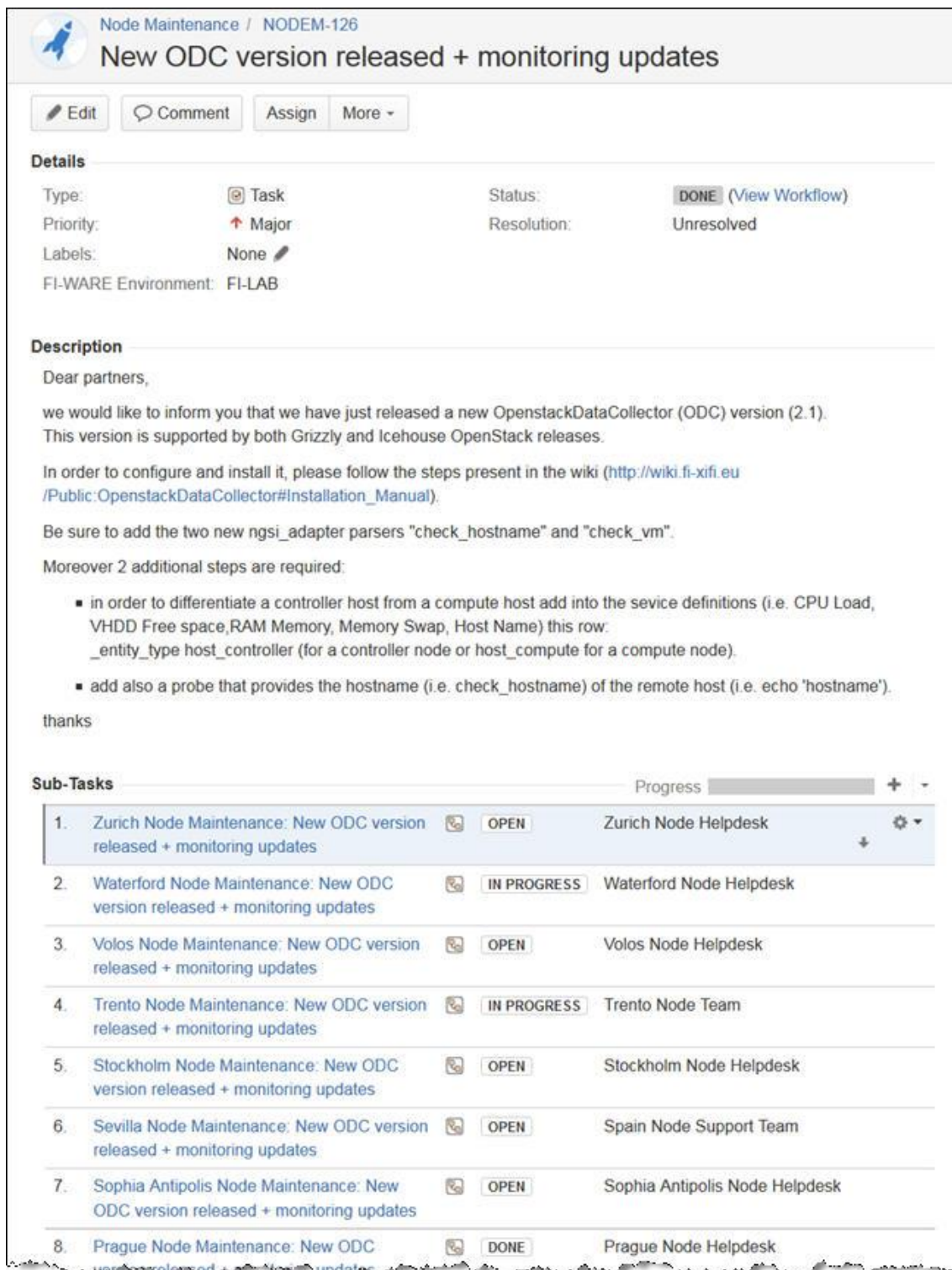
5.3.2 Interaction of Maintainers

Some tool support has been implemented in order to support the organisation and management of maintenance. The tool support targets those types of maintenance that involve all (or most of the) nodes in the federation. This could e.g. be the update of a FI-Ops or OpenStack component that needs to be made on all nodes. As it is principally the same task that needs to be done on all nodes, and because the large number of nodes requires a significant amount of monitoring of the acknowledgement, progress and completion of the maintenance, it is well suited for having it supported by a proper tool. For the sake of convenience and cost, the same tool that is also used for the helpdesk – Jira – has been employed for providing the tool support here.

When a maintenance task is planned and should be launched, the federation maintenance contact or federation maintainer needs to involve the infrastructures. This is done by creating a JIRA ticket assigned to all infrastructures required to participate in the maintenance. The ticket must describe which sub-system or component has to be maintained in the infrastructure addressed. The federation maintainer may take a coordinating role or may delegate the coordinating role to a particular infrastructure maintainer.

For realising the tool support, a special project “Node Maintenance” has been created in Jira.




When a new ticket (in project “Node maintenance”) is created, 18 sub-tasks will automatically be created – which is the current number of nodes in the federation. This means that the sub-tickets don't have to be created manually. They can all be found in the "master"-ticket, showing their current status, see Figure 55. Each subtask is assigned to a node, i.e. each node has its own subtask. The ticket creator and also any node can see the list with all the sub-tasks for every node and their status and thus has a very good overview of the progress of maintenance.



Node Maintenance / NODEM-126
New ODC version released + monitoring updates

[Edit](#) [Comment](#) [Assign](#) [More ▾](#)

Details

Type:  Task Status: **DONE** ([View Workflow](#))
 Priority:  Major Resolution: Unresolved
 Labels: None 
 FI-WARE Environment: FI-LAB

Description

Dear partners,

we would like to inform you that we have just released a new OpenstackDataCollector (ODC) version (2.1). This version is supported by both Grizzly and Icehouse OpenStack releases.

In order to configure and install it, please follow the steps present in the wiki (http://wiki.fi-xifi.eu/Public:OpenstackDataCollector#Installation_Manual).

Be sure to add the two new ngsi_adapter parsers "check_hostname" and "check_vm".

Moreover 2 additional steps are required:

- in order to differentiate a controller host from a compute host add into the sevice definitions (i.e. CPU Load, VHDD Free space, RAM Memory, Memory Swap, Host Name) this row:
 _entity_type host_controller (for a controller node or host_compute for a compute node).
- add also a probe that provides the hostname (i.e. check_hostname) of the remote host (i.e. echo 'hostname').

thanks

Sub-Tasks Progress + ▾









1.	Zurich Node Maintenance: New ODC version released + monitoring updates	 OPEN	Zurich Node Helpdesk
2.	Waterford Node Maintenance: New ODC version released + monitoring updates	 IN PROGRESS	Waterford Node Helpdesk
3.	Volos Node Maintenance: New ODC version released + monitoring updates	 OPEN	Volos Node Helpdesk
4.	Trento Node Maintenance: New ODC version released + monitoring updates	 IN PROGRESS	Trento Node Team
5.	Stockholm Node Maintenance: New ODC version released + monitoring updates	 OPEN	Stockholm Node Helpdesk
6.	Sevilla Node Maintenance: New ODC version released + monitoring updates	 OPEN	Spain Node Support Team
7.	Sophia Antipolis Node Maintenance: New ODC version released + monitoring updates	 OPEN	Sophia Antipolis Node Helpdesk
8.	Prague Node Maintenance: New ODC version released + monitoring updates	 DONE	Prague Node Helpdesk

Figure 55: Maintenance ticket in Jira

5.4 Sub-systems Subject to Maintenance

WP2 introduced the concept of independently testable sub-systems (cf. [D2.3]). It seems to be convenient to consider these sub-systems also as subject to maintenance processes aside the current definition that considers node infrastructures and software components. The following elements thus are subject to the maintenance process and its procedures.

5.4.1 Infrastructure Node

Maintenance of the 'bare metal' is under responsibility of the node owner. It is a mandatory task but is considered out of scope regarding the XIFI federation maintenance objectives. Nevertheless a number of hardware components herein are considered subject to maintenance processes due to their tight integration with software components under maintenance. For example, Openflow switches may be considered 'bare metal' but implement agents mandatory to support the PaaS components of the XIFI federation. Infrastructure node maintenance applies to:

- Computing resources (i.e. servers and associated storage sub-systems);
- Communication resources (i.e. NICs, switches, routers and cabling for various networks internal to the node);
- Software resources (i.e. operating systems, software deployment and maintenance tools as well as basic cloud services such as OpenStack, Fuel or similar).

Ideally maintenance tasks are focused on resource update, repair or replacement with minimum interaction with regards to other maintenance task considered next.

Scope	Maintenance Contact	Example	Possible Escalation	Sample Escalation Cause
Infrastructure local	none (internal)	Scheduled maintenance period	Federation-wide	Incident or other immediate unscheduled maintenance procedures - notification and coordination with federation
Infrastructure upon external request	Infrastructure	Maintenance procedure following-up an issue resolution	Multiple infrastructures or Federation-wide	Scope change of issue reported requiring coordinated actions
Multiple infrastructures peer-to-peer	Infrastructure	Maintenance of (point-to-point) communication services	Federation-wide	Need for temporary fail-over, remote success verification or support by another node required
Multiple infrastructures upon external request	Federation	Coordinated successive maintenance to minimize federation down-times	Federation-wide	Failure of scheduled maintenance procedure or excess of scheduled down-times
Federation-wide	Federation	Maintenance of a distributed sub-system	n.a.	n.a.

Table 76: Infrastructure Maintenance Escalation Levels

5.4.2 Communication Infrastructure

Communication between nodes is essential for operating and maintaining the XIFI federation. While the node internal communication infrastructure is not particular subject to the maintenance processes considered here, it is a mandatory platform for enabling communication between infrastructure nodes. For example, this consideration applies for the virtual router functions (VRFs) that allows an infrastructure node to connect to the MD-VPN and to the Internet at the same time and to route traffic between local node networks, MD-VPN and Internet. VRFs thus are infrastructure components but are under maintenance since all communication services of the federation are affected by any potential failure of these components. Maintenance of the communication infrastructure thus applies to:

- Edge switch (e.g. regarding the configuration of VLANs to access the MD-VPN, and to the OpenFlow functionality supporting the PaaS);
- L3 MD-VPN (e.g. regarding the local VRF maintenance, and the connectivity between nodes through the MD-VPN), involving a high degree of interaction between nodes, federation and external network providers in case of unscheduled maintenance (e.g. in case of fault recovery);
- L2 MD-VPN.

Scope	Maintenance Contact	Example	Possible Escalation	Sample Cause	Escalation
Node local	none (internal)	Maintenance of MD-VPN connectivity	Federation-wide	Loss of connectivity	
Node and network provider	Infrastructure	Maintenance of node's edge router and firewall	Federation-wide	Updates on a node's VRF, ACLs or transit networks	
Multiple nodes peer-to-peer	Federation	Verification of connectivity in following-up a maintenance procedure	Federation-wide	Connectivity problems for MD-VPN or public IP	
Multiple nodes and multiple network providers peer-to-peer	Federation	Maintenance of L2 connectivity	Federation-wide	Inconsistencies in peer-to-peer L2 tunnel configuration	
Federation-wide	Federation	Maintenance of DNS as a Service	n.a.	n.a.	

Table 77: Communication Infrastructure Maintenance Escalation Levels

5.4.3 Software Components

The complete list of software components, their dependencies, and related documentation is provided on the XIFI Wiki under Public:Software_Components[31] and (internally) under XIFI:Components[31]. Maintenance of software components is driven by request (e.g. an update request by the component owner) or as a consequence of sub-system failures triggering a maintenance process to resolve an issue observed (e.g. update, fail-over, roll-back, downgrade or restart). A scheduled maintenance process may apply to software components in order to ensure availability by

limiting the duration of unattended operations. This includes suspending, functional testing and resuming following a well-defined schedule, or a periodical refresh (e.g. restarting in a clean environment) to avoid accumulating issues. Software component maintenance applies to:

- All components are listed under XIFI:Components [32], including both Generic Enablers developed by XIFI and FIWARE. A process of escalating component issues from component maintainer towards component owner is part of the corresponding maintenance process.
- Third-party components required by above software components. These usually have to be scheduled along with a maintenance process that affects software components that depend on these third-party components: A maintenance process for a third-party component thus is triggered by the component owner of another software component except for security issues with such third-party components. In case of a security issue the component owner has to approve a maintenance process for a third-party component based on his knowledge of the depending software component.

Component	Co-location ^{Note1}		Component Maintainer	Component Owner ^{Note2} or Contributors	Internal Documentation ^{Note3}	Public Documentation
	Master node	All nodes				
ABNO Controller	yes	yes	XIFI	TID	published [33]	unpublished
Access Control GE	no	yes	XIFI	THALES	published [34]	Published [35]
Big Data GE	yes	yes	FIWARE	TID	---	published [36]
Cloud Portal	yes	no	XIFI	UPM-DIT	published [37]	published [38]
Context Broker GE	yes	yes	FIWARE	TID	---	published [39]
DEM Adapter	no	yes	XIFI	SYNELIXIS	published [40]	published [41]
Deployment and Configuration Adapter	yes	no	XIFI	SYNELIXIS	published [42]	published [43]
DCRM GE	yes	yes	XIFI	CREATE-NET, FIWARE	published [44]	published [45]
DNS as a Service	yes	no	XIFI	WIT, TSSG	published [46]	unpublished
Federation Manager	yes	no	XIFI	TUB	published [47]	published [48]
Federation Monitoring	yes	yes	XIFI	CREATE-NET	[49]	published [50]
Identity Management GE	yes	yes	XIFI	UPM-DIT, ENG, CREATE-NET	published [51]	published [52]
Infographics and status pages	yes	no	XIFI	CREATE-NET	published [53]	published [54]

Component	Co-location ^{Note1}		Component Maintainer	Component Owner ^{Note2} or Contributors	Internal Documentation ^{Note3}	Public Documentation
	Master node	All nodes				
Infrastructure Toolbox	yes	yes	XIFI	CREATE-NET	published [55]	published [56]
Interoperability tool	yes	yes	XIFI	IT-INNOV	published [57]	published [58]
Monitoring Dashboard	yes	no	XIFI	WIT	published [59]	published [60]
NAM Adapter	no	yes	XIFI	UPM	published [61]	published [62]
Network Provisioning Manager	yes	yes	XIFI	TID	published [63]	unpublished
NGSI Adapter	no	yes	XIFI	TID	published [64]	published [65]
NPM Adapter	no	yes	XIFI	TI	published [66]	Published [67]
OpenNaaS	yes	no	XIFI	I2CAT	published [68]	unpublished
OpenStack Data Collector	no	yes	XIFI	CREATE-NET	published [69]	Published [70]
Path Computation Element	yes	no	XIFI	TID	published [71]	unpublished
Platform as a Service Manager GE	yes	no	XIFI	TID, FIWARE	published [72]	Published [73]
Quick Online Test	yes	yes	XIFI	WIT	published [74]	unpublished
Resource Catalogue and Recommendation tools	yes	no	XIFI	ATOS, UPM-SSR	published [75]	published [76]
Software Deployment and Configuration GE	yes	yes	XIFI	TID, FIWARE	published [77]	published [78]
Security Dashboard	yes	no	XIFI	ATOS, THALES	published [79]	published [80]
Security Monitoring	yes	yes	XIFI	ATOS	published [81]	published [82]
Security Proxy	no	yes	XIFI	UPM-DIT	published [31]	published [83]
SLA Manager	yes	no	XIFI	ATOS, SYNELIXIS	published [84]	published [84]

^{Note1} Co-location refers to the requirement that a certain component must be deployed to a master node or must be present on a master node for proper functioning (Master node) or that it can or must be deployed to other nodes (All nodes). The following distinctions have been made:

Component	Co-location ^{Note1}		Component Maintainer	Component Owner ^{Note2} or Contributors	Internal Documentation ^{Note3}	Public Documentation
	Master node	All nodes				
<p>Master node: yes, All nodes: no – The component under consideration must be present on the master note. It must not be deployed to other nodes.</p> <p>Master node: no, All nodes: yes – The component under consideration can be deployed to any node. That includes the option that the component must be present on all nodes for correct operation.</p> <p>Master node: yes, All nodes: yes – The component under consideration must be deployed to a master node and to other nodes for proper operation. This includes that a component may be configured differently for master nodes and other nodes, and that it may not be deployed to all but only to some selected other nodes.</p> <p>^{Note2} A component may consist of multiple “sub-components” having distinct ownership. It may be considered as a sub-system if this collection of “sub-components” is self-contained. From the maintenance perspective, a component is associated with only one component maintainer while a sub-system is associated with multiple component maintainers under coordination by a sub-system maintainer. This distinction may be used to decide if the component under consideration is considered a component or a sub-system.</p> <p>^{Note3} This is the situation at the time of writing. Being “unpublished“ does not imply that the component documentation relevant for maintenance is not available, but that it does not have a publicly accessible link from the main portal and that it is currently accessible only for internal federation use. It may be made public later if suitable.</p>						

Table 78: Components under Maintenance

The component maintainer role must be assumed by the main contributor to this component. In practice, a single component may have many contributors. In this case, the maintainer should be jointly nominated by the contributors. It should be considered more convenient to handle a component formally like a sub-system if it does not have a clear ownership. That is, such component should be assigned to a component maintenance contact who is informed about whom the contributors to assign as a maintainer temporarily.

5.4.4 Software Sub-systems

A software sub-system constitutes as a collection of components that interact for implementing a certain objective. WP2 defined a sub-system with regards to testability aspects as a collection of components that interact for a given purpose, that are self-contained only depending on basic infrastructure services and do not depend on other sub-systems or components for implementing their objective. In consequence, a sub-system has well-defined interfaces and can be tested for interoperability, conformance and performance by defining its operational parameters, applying input parameters and observing resulting output parameters. A software sub-system thus can be subject to a maintenance process since any issue with an enclosed component will affect the sub-system only. A scheduled maintenance process can be much more efficient when applied to a sub-system rather than to an independent set of components. Since all interacting components are affected at the same time reducing the probability of inconsistent internal states, simplifying the provision of fail-over configurations and reducing down-times. The complete list of software sub-systems, their dependencies and related documentation is provided on the internal XIFI Wiki under XIFI:Subsystems. Software sub-system maintenance applies to:

Sub-system		First Level Maintenance contact ^{Note1}	Responsible Maintainer ^{Note2}	Possible next action ^{Note3}
Monitoring		Federation	Sub-system	Delegate to component maintainer or escalate to infrastructure maintainer
Security	Identity management	Federation	Sub-system	Delegate to component maintainer or escalate to infrastructure maintainer (master node)
	Security monitoring	Federation	Sub-system	Delegate to component maintainer or escalate to infrastructure maintainer
User Oriented and GUI Subsystems	Monitoring Dashboard	Sub-system	Sub-system	Delegate to component maintainer or to infrastructure maintainer, or escalate to federation maintainer
	Security Dashboard	Sub-system	Sub-system	Delegate to component maintainer or escalate to federation maintainer and infrastructure maintainer in parallel
	Infographics and status pages	Sub-system	Sub-system	Delegate to component maintainer or infrastructure maintainer, or escalate to federation maintainer
	Cloud Portal	Sub-system	Sub-system	Delegate to component maintainer, other sub-system maintainer or infrastructure maintainer, or escalate to federation maintainer
	SLA Manager	Federation	Sub-system	Delegate to component maintainer or to other sub-system maintainer, or escalate to infrastructure maintainer or federation maintainer
	Federation Manager	Sub-system	Sub-system	Delegate to component maintainer or to other sub-system maintainer, or escalate to infrastructure maintainer or federation maintainer
	Interoperability tool	Sub-system	Sub-system	Delegate to component maintainer or to other sub-system maintainer, or escalate to infrastructure maintainer or federation maintainer
	Resource Catalogue	Federation	Sub-system	Delegate to component maintainer or to infrastructure maintainer, or escalate to federation maintainer
Deployment & Operations Subsystems	Infrastructure Toolbox	Sub-system	Sub-system	Delegate to component maintainer or to other sub-system maintainer, or escalate to infrastructure maintainer
	Deployment and Configuration Adapter	Sub-system	Sub-system	Delegate to component maintainer or to other sub-system maintainer, or escalate to infrastructure maintainer
	PaaS Manager	Federation	Sub-system	Delegate to component maintainer or

Sub-system		First Level Maintenance contact ^{Note1}	Responsible Maintainer ^{Note2}	Possible next action ^{Note3}
	GE			escalate to infrastructure maintainer or federation maintainer
	SDC GE	Sub-system	Sub-system	Delegate to component maintainer or escalate to infrastructure maintainer
	Quick Online Test	Sub-system	Sub-system	Delegate to component maintainer or escalate to infrastructure maintainer
<p>^{Note1} The first level contact should be the default recipient of an issue report. In case of a distributed sub-system being subject to the issue report, this should be the Federation Maintenance Contact. If the sub-system is co-located with more than one node and the issue report is not specifying a particular node then the federation manager can delegate the issue report to one or more infrastructure maintainers or to the particular sub-system maintainer. All other issue reports may be directed to the infrastructure or sub-system maintainers.</p> <p>^{Note2} Responsibility of this maintainer is in coordinating the issue resolution either as a delegate (i.e. having ownership assigned by the federation maintainer or an infrastructure maintainer), or by receiving the request through its associated contact (i.e. infrastructure or sub-system maintenance contact).</p> <p>^{Note3} A maintainer has several options to resolve an issue. By default, the responsible maintainer analyses the issue and then delegates to an appropriate maintainer for resolving the issue. In case the responsible maintainer is not able to decide on the next step or the issue has to be handled in a wider scope, it might be needed to escalate the issue.</p>				

Table 79: Sub-systems under Maintenance

5.4.5 Procedures of the Maintenance Process

Deliverable D5.1 identified the following types of maintenance processes:

- **Scheduled maintenance**

A regular and planned, usually periodic, procedure applied to a dedicated component, sub-system or infrastructure (or to the federation in whole). Its purpose is to monitor and evaluate the risk of failure and to prevent the occurrence of incidents. It also aims to reduce the amount of disruptions of regular operations through unscheduled maintenance processes. Optimization of regular operations and service enhancement is not in scope of scheduled maintenance but part of a quality assurance process.

Example: Regular Hardware or software updates (e.g. security updates); usually scheduled per infrastructure node optionally minimizing down-times collaboratively.

- **Unscheduled maintenance**

A procedure usually initiated through an unsolicited (external or internal) event such as a foreseeable fault condition prior to its occurrence or as a protective measure to avoid potential failure. Applicable to all potential issues that cannot be assigned to scheduled maintenance due to a short deadline.

Example: Replacement of defective equipment or of misbehaving software; includes restoring back-up states; unplanned or scheduled on short notice usually responding to urgent action requirements; may incur node or federation down-times.

- **Incident handling**

Incident handling as a maintenance procedure responses to an immediate critical issue and interrupts (more or less disruptive) regular operations to resolve an ongoing failure situation or an

immediate threat. Incident handling may be preventive with a very short deadline as encountered in the course of power failures.

Example: Failure of major node, federation or communication infrastructure, potentially due to physical damage; usually involves significant down-times with barely predictable duration; requires preparation of incident handling processes including risk management strategies; may require subsequent unscheduled maintenance processes being initiated.

In addition to these, maintenance procedures can be initiated in the course of an issue resolution process initiated by a developer (clearly, this also may be an internal developer or component owner reporting an issue and must not necessarily involve developers external to XIFI). Depending on the relevance of the issue addressed, the maintenance procedure then is assigned to one of the three categories outlined above. The following figure depicts the management of the maintenance process as a business process in form of an event-driven process chain (EPC).



Figure 56: Management of maintenance procedures (most relevant cases of the management process)

The leftmost portion of the process chain shown in the figure denotes the regular case for an issue report created by a developer. In most cases it is not necessary to include a maintenance procedure as part of the issue resolution process. For the initial start-up of the XIFI federation, still gaining

experience in operations and maintenance, it is nevertheless reasonable to assume that an issue resolution will frequently cause a subsequent maintenance procedure to be performed, or to perform a maintenance procedure as the issue resolution itself. In that case the issue responsible (which is assumed here as a simplification of the various options to assign responsibilities to stakeholders) may request to perform a maintenance procedure (e.g. a software update for the federation in consequence of removing a software bug in a sub-system).

Occasionally, it may happen that no suitable maintenance procedure is available and must be defined in the course of an issue resolution (e.g. if certain test cases must be implemented and performed prior to a software update). In that case, a certain quality assurance process should be maintained that allows reviewing and approving the proposed new maintenance procedure prior to implement it in the federation.

The assignment of responsibilities in this management process should be considered as preliminary and subject to further revision. But is reasonable to assume that

- the federation maintainer will need to take the lead if a decision in this management process is required (e.g. an agreement on the maintenance plan, which is the basis for federation-wide scheduled maintenance activities);
- the federation maintainer will collaborate with infrastructure maintainer(s) and will involve sub-system maintainers in technical decisions whenever a decision affects the operation and maintenance process of the federation or part of it (e.g. when approving a new maintenance procedure that must be deployed to infrastructures subsequently).

5.4.6 Scheduled Maintenance (Single Infrastructure Node)

The following figure details the workflow of a scheduled maintenance procedure for a single node.

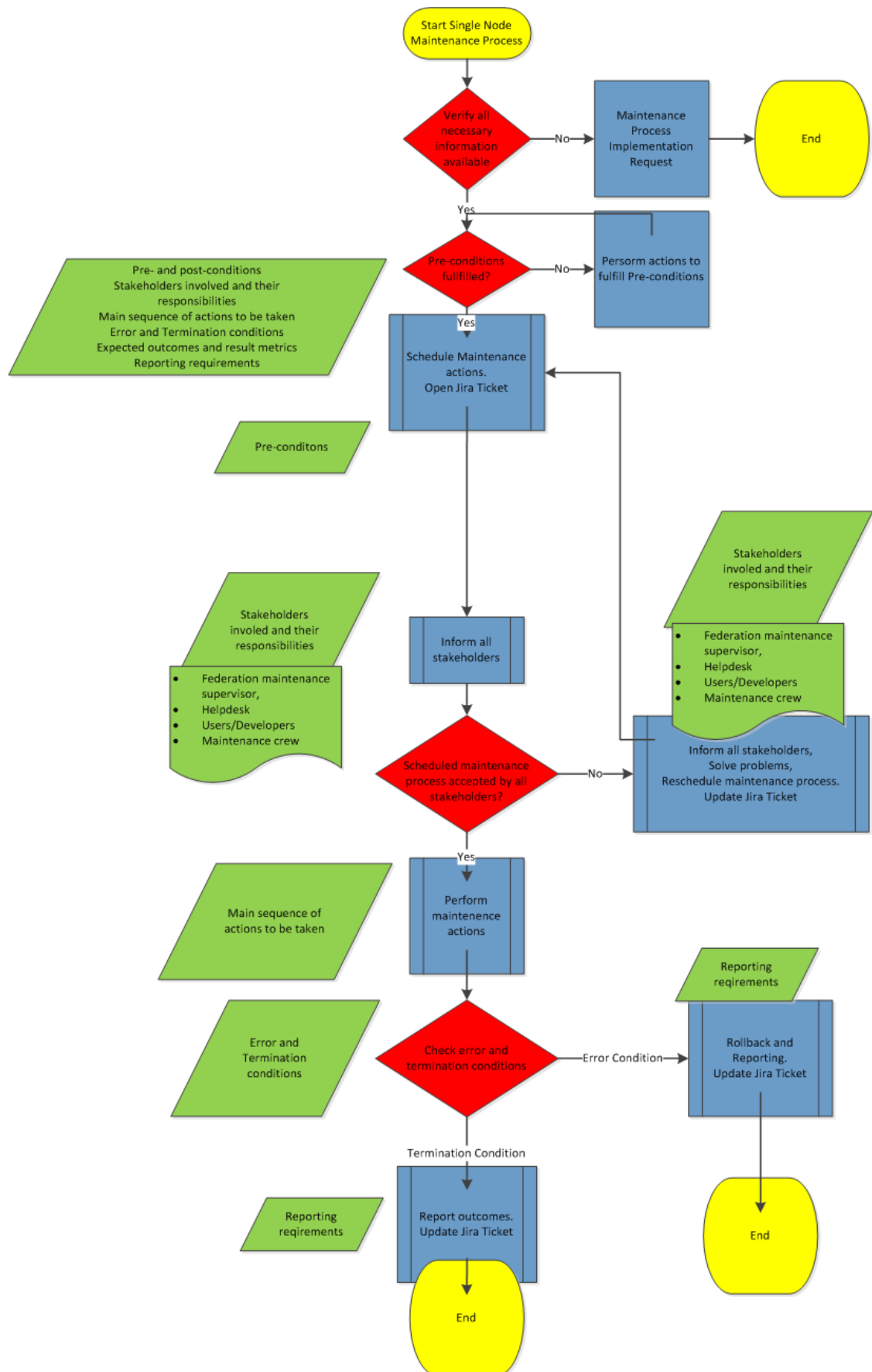


Figure 57: Outline of a scheduled maintenance procedure affecting a single infrastructure node

A maintenance procedure involving only a single infrastructure node in general is started by the infrastructure maintainer (clearly, an infrastructure maintainer might respond in that to a request of the federation maintainer or a sub-system maintainer). In the first step the infrastructure maintainer must check whether the maintenance procedure requested is defined and available, and it must be validated that all needed information for performing the maintenance procedure is at hand. The following information must be available to describe the procedure:

- Pre- and post-conditions -- to ensure that the procedure can be imitated, and that it has a clear target outcome;
- Stakeholders involved and their responsibilities -- to ensure that all stakeholders affected by the procedure can be informed about activation and success (or failure) of the procedure and that all contact points are at hand if support will be needed in the course of performing the maintenance procedure;
- A main sequence of actions to be taken -- to ensure that procedure is well determined and reproducible;
- Error and Termination conditions -- to ensure the proper successful termination of the procedure or a fail-safe handling in case of problems in performing the procedure;
- Expected outcomes and result metrics -- to allow validation of the result and to judge if the procedure succeeded, succeeded partially or failed and may require a roll-back;
- Reporting requirements (for filing and effectiveness evaluation) -- to report consistently to stakeholders involved and to document the outcome of a procedure performed for quality assurance.

Ideally, the approval process for new maintenance procedures must ensure that these conditions are met.

Reporting to stakeholders (referring to the sub-process "inform all stakeholders") is considered a dedicated maintenance procedure since there may be special requirements on the form of a report or on whom to inform in case of a failure of the procedure, which might differ from the audience addressed in case of a successful completion.

5.4.7 Scheduled Maintenance (Multiple Infrastructure Nodes)

The following figure details the workflow of a scheduled federation-wide maintenance procedure involving multiple infrastructure nodes.

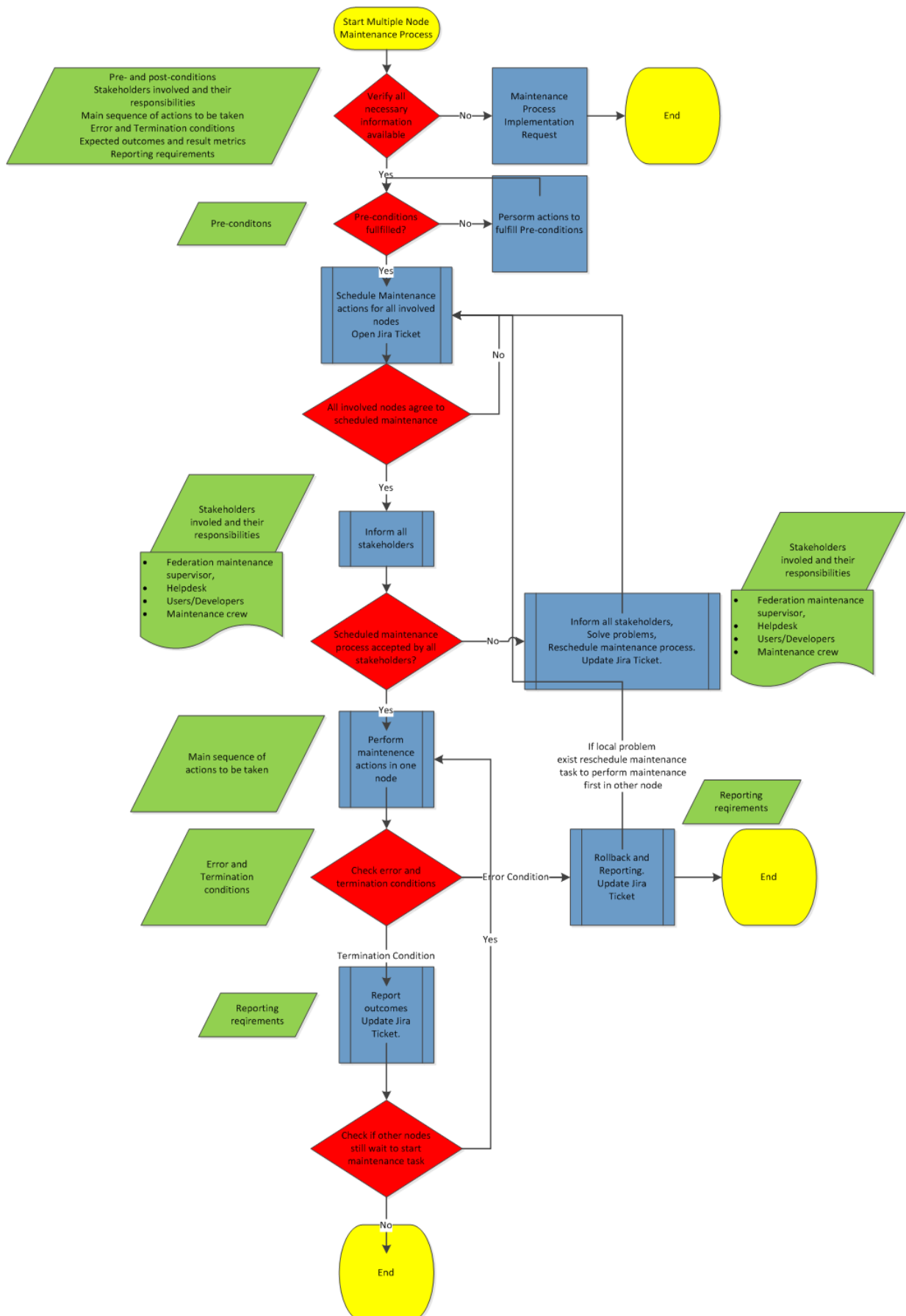


Figure 58: Outline of scheduled federation-wide maintenance procedure affecting multiple infrastructure nodes

A maintenance procedure involving multiple nodes usually is initiated by the federation maintainer (clearly, the federation maintainer may act upon request by an infrastructure maintainer or by a sub-system maintainer). In the first step the federation maintainer must check whether the maintenance procedure requested is defined and available, and it must be validated that all needed information for performing the maintenance procedure is at hand. Since the infrastructure maintainer rarely acts directly on the infrastructure nodes in course of a maintenance procedure but rather coordinates between infrastructure maintainers, validating prerequisites for performing a maintenance procedure may already be delegated to the performing infrastructure maintainers. Upon completion of all scheduled activities in the course of the maintenance procedure by all infrastructure maintainers involved, the federation maintainer again takes control of the procedure to judge upon the procedure's outcome and to inform stakeholders affected.

5.4.8 **Unscheduled Maintenance (Single Infrastructure Node)**

Unscheduled maintenance may be required in case an issue has been detected that needs consideration on short notice. It should be noted here that this is in contrast to incident handling since the latter may need immediate consideration and also might have disabled already the infrastructure node or its capacity to federate and to inform affected stakeholders. Incident handling in consequence most often applies to recovery procedures while unscheduled maintenance is still a well-defined and controlled process. A maintenance procedure should only be performed unscheduled if it addresses a subject that has or would have major impact on the function, capacity or performance of one or more infrastructure nodes. It is assumed that such can only occur at one or several nodes of the federation but not on the whole federation at a given time. The following figure details the procedure for unscheduled maintenance on a single infrastructure node.

5.4.9 **Review of the Maintenance Procedures**

In order to implement the maintenance process a number of procedures have been defined and specified to some detail. These are grouped according to a common scope allowing to implement all or part of it depending on the particular requirements of the node infrastructures. Since the federated node infrastructures are heterogeneous to some degree it is up to the discretion of a node

- to implement the full set of procedures;
- to implement a sub-set of the procedures and to complement these by proprietary solutions to achieve a comparable process capability;
- to skip or postpone implementation of some procedures potentially unnecessary, unwanted or locally assigned to the responsibility of third-parties for the node.

This flexibility is needed to adapt to the different policies on organizational and national level while maintaining the level of collaboration required for the federation.

The procedures described focus on infrastructure maintenance, software maintenance and collaboration among nodes. Therefore, they are building blocks to implement Operational Level Agreements (OLA) among nodes and in consequence determine Service Level Agreements (SLA) that can be provided by the federation to the user.

Depending on the main purpose of a procedure the description provides a detailed sequence of instructions to set-up the infrastructure for a particular target (e.g. in order to later customize a node set-up originally provided by the ITBox), provides a formal or informal description of a process chain that needs to be customized and detailed by particular actions or instruction by the node maintainer (e.g. maintenance of floating IP addresses), or provides a discussion of best practice strategies that involve third parties or external stakeholders (e.g. to resolve network connectivity problems in collaboration with a node's ISP).

A **collaboration framework** has been established and described partly already in scope of D5.3. The most prominent consists of this framework are the formalization of the interaction between nodes and between user and nodes through the Jira help-desk and ticketing system, and the formal process that handles maintenance procedures' creation, verification, approval and deployment (as well as revocation if needed). This framework now has been complemented by the shared software repository as a tool that simplifies the implementation of software maintenance processes, and the procedures to utilize this repository efficiently.

The **infrastructure maintenance procedures** are aiming to ensure that a node can maintain its role in the federation in terms of resilience, functionality and trust, and are applicable for both scheduled and unscheduled maintenance (e.g. for incident handling), if needed. Although infrastructure maintenance includes barebone maintenance (i.e. keeping up and running the physical infrastructure consisting of server hardware, operating system, storage and local network), this clearly is not in focus of infrastructure maintenance but is seen as a prerequisite to be performed privately by the node. Therefore, this is not formalized in the federation context.

The **software maintenance procedures** aim to maintain the level of service quality provided by a node to the federation. Software maintenance procedures apply to software packages (e.g. installation, removal, upgrade) of the platform (e.g. on OpenStack upgrades), to the cloud services (e.g. to the monitoring subsystem) and to software components accessible for the user through public interfaces (i.e. Generic and Specific Enablers). A number of maintenance procedures defined here are considered "software maintenance" but are actually at the boundaries towards infrastructure maintenance (e.g. Maintenance of floating IP pools) or framework (e.g. Alignment of SW maintenance procedures between FI-WARE and XIFI).

Infrastructure maintenance

Infrastructure maintenance procedures at the time of writing cover three distinct application scopes

- Node customization;
- Resources management;
- Fault management.

Node customization addresses post-deployment changes due to new resource demands loaded onto a node infrastructure, for example [85].

The procedure addresses modifications to the three main areas networking (e.g. routing, broadcast domains, OpenFlow configuration, IP address allocation ...), software (e.g. DCRM configuration, object and block storage on top of iSCSI and NFS, Nagios monitoring, ...) and hardware resources (e.g. adding compute nodes, switches and firewalls).

This customization assumes an ITBox based configurations as a starting point for all customizations discussed. Node heterogeneity or node customization potentially leads to compatibility issues that may degrade or even disable a node in the federation. In that case, a procedure was defined that provides guidelines on handling nodes that are not (or no longer) compliant with the federation (see Not compliant node [86]).

Resource Management addresses maintenance of resources pools and operational means that manage resources (e.g. resource management tools). Management of tenant quotas [88] applies to compute (e.g. cores, instances, RAM ...), block storage (e.g. volume size) and network resource quotas (e.g. router, subnets, floating IPs) in the OpenStack environment. The procedures enabling resource disengagement [87] in particular aim to provide hints how to detect unused or suspicious resources and how to remove them sanely. Unused resources often remain from temporary users and must be removed or returned to the resource pool for operational reasons. Suspicious use of resources may

indicate malicious use or an ongoing attack and must be logged for legal reasons and must be removed for security reasons. Resources under consideration in this scope are the same as considered for tenant quota management.

Finally, inter-node network connectivity maintenance [89] addresses a variety of configuration options and stakeholders involved in the maintenance procedures and is therefore to be considered as a framework of possible actions to identify potential networking problems and corrective actions. For a number of network topologies observed with the different federated node infrastructures, the network connectivity maintenance procedures provide hints for possible failure scenarios regarding symptoms that should trigger corrective actions, stakeholders that should be involved in these actions and processes that should be followed to implement these actions efficiently.

Fault management procedures address fail over scenarios (i.e. describing procedures that may cause a type of corrective response, see [90]) and links these with the OpenStack high availability (HA) capacities for control, compute and storage nodes. In addition, an analysis of requirements regarding the recovery from security-related events (see [91]) in the federation is provided. It concludes that a Computer Security Incident Response Team (CSIRT) should be established for the federation to define the type of events that may be considered as security relevant, who should be informed about security-related events, and how to respond to a security-related event. It provides some general recommendations give some initial suggestions for the questions above.

Software maintenance

Software maintenance procedures at the time of writing cover four distinct application scopes

- Software distribution;
- Resource management;
- Collaboration;
- Fault management.

It should be noted here that a number of software maintenance procedures are currently under revision to include the XIFI software repository as a concept and tool to simplify (and unify) the procedures mentioned below under software distribution, collaboration and fault management.

Software distribution procedures cover several aspect of maintenance for the node infrastructure's software basis. There have been procedures defined for deploying software to a node (see [92]), for updating deployed software [93]. These also address the creation, validation and maintenance of software packages containing sub-systems consisting of multiple components. Additionally, software compliance topics are addressed [94] and are linked with the software component testing strategies outlined in D2.3 and D2.6. Finally, software backup procedures [95] are outlined initially and are set into relation to the fail-over procedures. Finally, OpenStack release upgrade [96] is considered a software maintenance procedure of relevance for infrastructure maintenance and node collaboration as well. The detailed procedures are under development at the time of writing since the federation currently undergoes its first major upgrade and the task of migration between releases already proved to be of uttermost complexity.

Resource management is a topic in scope for both infrastructure and software maintenance assuming that all types of resources under consideration here are at some point mapped to hardware, operating system or network resources. Procedures regarding the maintenance of floating IP pools [97] are considered as software maintenance here because various requirements from operations and maintenance as well as from the jurisdictional domains apply. This requires procedures covering the management of floating IPs as scarce node resources (considering save resource disengagement procedures as well as detection of unused resources or misuse of resources), fail-over procedures (e.g

for enabling fall-back solutions in case of exhausted pools) as well as monitoring and logging requirements due to legal considerations (e.g. non-repudiation requirements).

Collaboration among nodes here is considered a software maintenance topic since it covers the alignment of software maintenance procedures across several projects such as FIWARE and XIFI [98] under the common umbrella of FIWARE-Ops and FIWARE-Lab. This includes already reported approaches (ref. D5.3) such as the user communication through the Jira help-desk (see [99]), the node maintainer ticket system implemented through Jira, and the announcement procedures for scheduled and unscheduled node maintenance events (see[100])

Fault management is considered both a software and infrastructure maintenance topic and requires distinct procedures. Here we considered the recovery from security-related events (see[101]) as complementary to the infrastructure fault management section from the previous section. It is suggested here also to establish a CSIRT further detailing particular response procedures. Since immediate reaction is required in case of suspicious behavior, there is strong interaction and consideration of all other software maintenance procedures required, and assistance through infrastructure maintenance must be sought, underlining the need for a federation CSIRT.

5.4.10 Software Repository

Repository concept

The XIFI software repository aims to provide downloadable software packages ready for deployment to the XIFI nodes. It is considered part of the continuous development workflow and of the maintenance process.

Figure 59 is outlining the overall process consisting of (from left to right) the component development and test supported by ETICS and several other individual tools, the deployment of components for packaging with a sub-system or for testing utilizing the XIFI testbeds, the integration and test of sub-systems potentially utilizing the XIFI testbeds, and deployment of sub-systems to the federated infrastructure nodes. Since infrastructures have to maintain continuous operation and rely on their individual maintenance cycles and schedules, the repository indicates to nodes the update of deployed packages (components or sub-systems) for further use.

The XIFI software repository is at the boundaries between software development, infrastructure node operations, and node maintenance. It thus is an integral part of the continuous development and integration process (WP2) as well as of the node operations and maintenance processes (WP5).

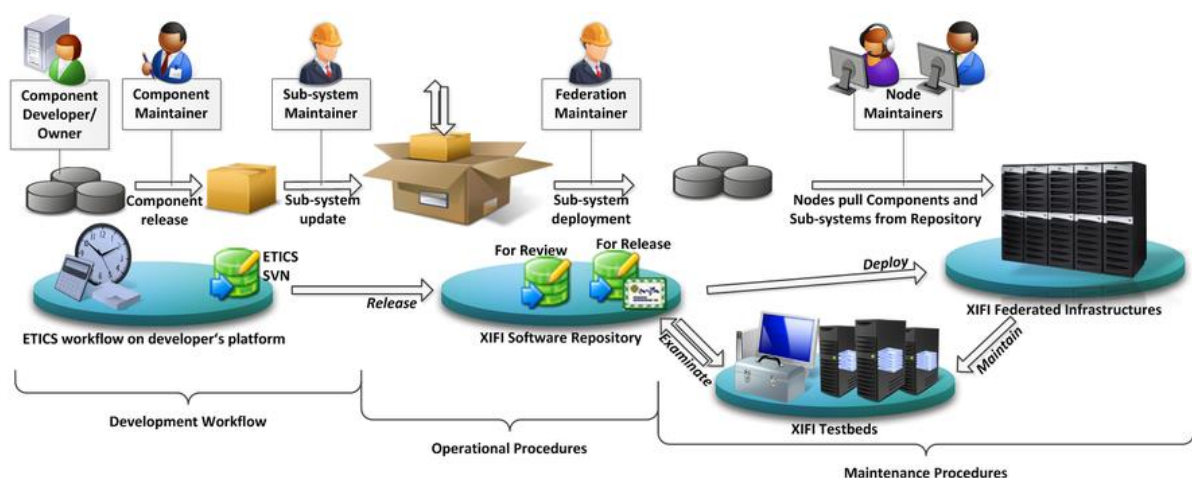


Figure 59: Continuous integration workflow - Overview

The software repository provides distinct package repositories intended for

- deployment of valid sub-system packages to operative nodes;

- download of component packages for testing in the XIFI testbeds;
- download of sub-system packages for testing in the XIFI testbeds;
- And for upload of faulty sub-systems from the operational nodes for further evaluation.

It is accessible for sub-system maintainers and node maintainers.

Instance Configuration:

The repository is a virtual machine (VM) providing a minimum number of services required to maintain the repository and to synchronize the data storage with another (secondary) host:

- SSH root access
- SSH access for node maintainers (user NodeMaint)
- SSH access for sub-system maintainers (user SubSysMaint)
- An Apache Maven 2 repository
- RSync demon via ssh (tentative)

The repository storage is provided by a virtual disk (also maintaining a current snapshot of the VM for convenience).

The VM can be copied from one node to another along with its storage volume in case a secondary repository host is required. Secondary hosts should then be configured to synchronize their local storage volumes via rsync. Please note that any communication between primary and secondary nodes should take place via the MD-VPN or must use end-to-end encryption to protect the credentials stored in the VM and in the virtual disk volume.

VM Image details:

Property	Value
Property 'base_image_ref'	677dadbd4-6e87-4522-bd8b-82e3d51b76af
Property 'image_location'	snapshot
Property 'image_state'	available
Property 'image_type'	snapshot
Property 'instance_type_ephemeral_gb'	0
Property 'instance_type_flavorid'	3
Property 'instance_type_id'	1
Property 'instance_type_memory_mb'	4096
Property 'instance_type_name'	m1.medium
Property 'instance_type_root_gb'	40
Property 'instance_type_rxtx_factor'	1
Property 'instance_type_swap'	0
Property 'instance_type_vcpu_weight'	None
Property 'instance_type_vcpus'	2
Property 'instance_uuid'	0b91399f-0315-41a1-874d-e07003ab92b5
Property 'owner_id'	000000000000000000000000000000000003015
Property 'user_id'	bernd-bochow
checksum	7d673ae642a7b03d907959939cd2136c
container_format	ovf
created_at	2014-10-22T11:41:34
deleted	False
disk_format	qcow2
id	a3085b37-6e94-4ae7-8fab-620b083c90ae
is_public	False
min_disk	0
min_ram	0
name	xifi-sw-repository_20141022
owner	000000000000000000000000000000000003015
protected	False
size	2890399744
status	active


```
| updated_at | 2014-10-22T11:51:08 |
+-----+-----+
```

Figure 60: VM image details

A virtual **disk volume** acts as the repository data storage. It has been given an initial size of 200GB but can be enlarged if needed. The disk is imported to the VM as /dev/vdb and is mounted under /mnt/volume. Snapshots of this disk are not available due to its size. It is assumed that an initial copy is required only since later changes can be applied through a regular rsync.

```
[root@host-192-168-120-14 ~]# cat /proc/partitions
    major minor  #blocks  name

252          0   41943040 vda
252          1    512000 vda1
252          2   9971712 vda2
253          0   1835008 dm-0
253          1   7077888 dm-1
252         16  209715200 vdb
```

You can find in the table below the list of ports open on the instance and its corresponding services.

XIFI software repository hosts and services						
	Host	Hosting node	Contact	Port	Protocol	Service
Primary	193.175.132.52	Berlin	xifi-support-berlin	22	TCP	ssh (valid key required)
				873	TCP	rsync (using ssh)
				8081	http	Artifactory (provides the Maven-2 central repository)

Table 80: XIFI software repository hosts and services

Repositories provided

The figure below shows how the web interface of the repository looks like.

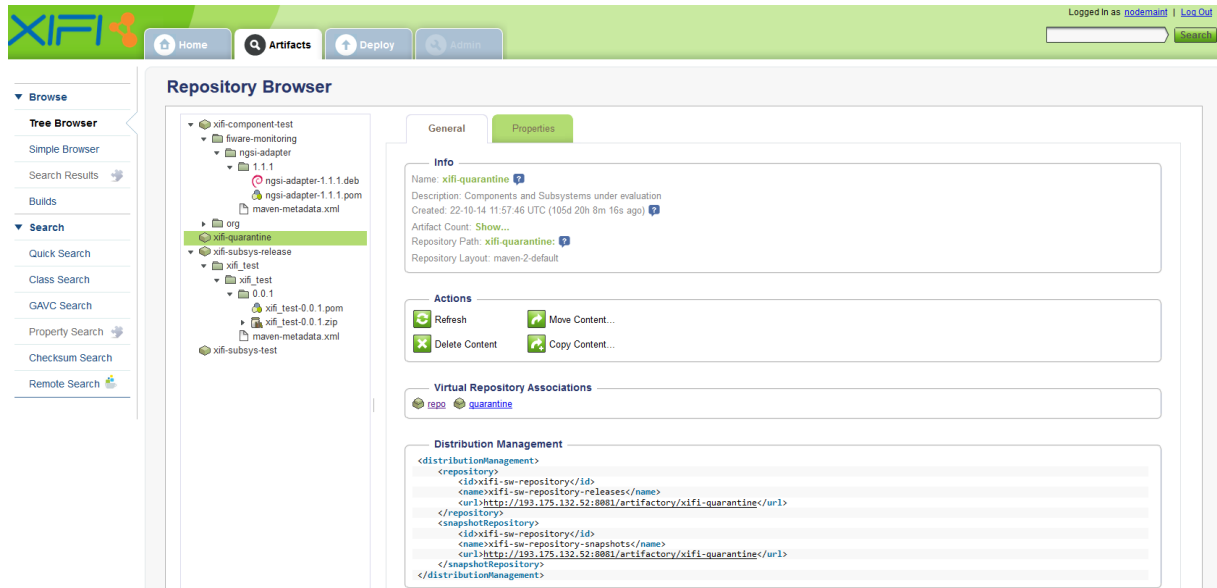


Figure 61: Artifactory Web Interface

When deploying your Maven builds, you must ensure that any `<repository>` element in your distribution settings has a corresponding `<server>` element in the settings.xml file with a valid username and password. Since the repositories are provided by artifactory, passwords are encrypted in the example below.

Sample maven-2 settings.xml configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.1.0
http://maven.apache.org/xsd/settings-1.1.0.xsd"
  xmlns="http://maven.apache.org/SETTINGS/1.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <servers>
    <server>
      <username>user</username>
      <password>SANITIZED</password>
      <id>central</id>
    </server>
    <server>
      <username>user</username>
      <password>SANITIZED</password>
      <id>snapshots</id>
    </server>
  </servers>
  <profiles>
    <profile>
      <repositories>
        <repository>
          <snapshots>
            <enabled>>false</enabled>
          </snapshots>
          <id>central</id>
          <name>release-subsys</name>
          <url>http://193.175.132.52:8081/artifactory/release-subsys</url>
        </repository>
```

```

    <repository>
      <snapshots />
      <id>snapshots</id>
      <name>test-subsys</name>
      <url>http://193.175.132.52:8081/artifactory/test-subsys</url>
    </repository>
  </repositories>
  <pluginRepositories>
    <pluginRepository>
      <snapshots>
        <enabled>>false</enabled>
      </snapshots>
      <id>central</id>
      <name>test-component</name>
      <url>http://193.175.132.52:8081/artifactory/test-component</url>
    </pluginRepository>
    <pluginRepository>
      <snapshots />
      <id>snapshots</id>
      <name>quarantine</name>
      <url>http://193.175.132.52:8081/artifactory/quarantine</url>
    </pluginRepository>
  </pluginRepositories>
  <id>artifactory</id>
</profile>
</profiles>
<activeProfiles>
  <activeProfile>artifactory</activeProfile>
</activeProfiles>
</settings>

```

Repository : xifi-subsys-release

Scope: XIFI Subsystems ready for deployment to the federated nodes.

Purpose: Only Subsystem maintainers should upload to this repository. Node maintainers may pull Subsystem packages from this repository to deploy to their node infrastructures.

Sample Maven-2 settings to deploy to this repository:

```

<distributionManagement>
  <repository>
    <id>xifi-sw-repository</id>
    <name>xifi-sw-repository-releases</name>
    <url>http://193.175.132.52:8081/artifactory/xifi-subsys-release</url>
  </repository>
</distributionManagement>

```

Suggested Repository Layout:

```

xifi-component-test
xifi-quarantine
xifi-subsys-release
  -- group-id
  -- subsystem-id

```

```
-- revision
    <subsystem-id>-<revision>.pom
    <subsystem-id>-<revision>.zip
    maven-metadata.xml
xifi-subsys-test
```

Group-id:**Maven:** <groupId>group-id</groupId>**Groups:** XIFI-SubSystem | XIFI-Component**Subsystem-id:****Maven:** <artifactId>subsystem-id</artifactId>**Subsystems:** Package name for the subsystem (excluding revision number).

Recommended package naming: Monitoring | IdentityManagement | SecurityMonitoring | MonitoringDashboard | SecurityDashboard | InfographicsStatusPages | CloudPortal | SLAManager | FederationManager | InteroperabilityTool | ResourceCatalogue | InfrastructureToolbox | DeploymentConfiguration | PaaSManager | SDC

Revision:**Maven:** <version>0.0.1</version>**Versions numbers:** Should have at least three digits in the format <release number>.<major version number>.<minor version number>**subsystemid-revision.pom**

The Maven POM file. Sample POM file:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd"
    xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <modelVersion>4.0.0</modelVersion>
  <groupId>XIFI-SubSystem</groupId>
  <artifactId>FederationManager</artifactId>
  <version>1.0.0</version>
  <packaging>jar</packaging>
  <description>Federation Manager POM file</description>
</project>
```

subsystemid-revision.zip

The package file containing all sub-systems files. Third-party components may be bundled with the sub-system package or may be referenced through Maven resolution links.

Packaging formats: ZIP | JAR | WAR | RPM | (other common packaging formats under discussion)

maven-metadata.xml

Maven artifact descriptor. Sample file:

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata>
  <groupId>XIFI-SubSystem</groupId>
```

```
<artifactId>FederationManager</artifactId>
<version>0.0.1</version>
<versioning>
  <latest>1.0.0</latest>
  <release>1.0.0</release>
  <versions>
    <version>1.0.0</version>
  </versions>
  <lastUpdated>20141024110838</lastUpdated>
</versioning>
</metadata>
```

Repository: xifi-subsys-test

Scope: XIFI Subsystems ready for testing in the testbed environment.

Purpose: Subsystem maintainers should upload to this repository prior to request testbed maintainers or test owners to conduct tests on this Subsystem. Test owners or testbed maintainers pull Subsystem packages from this repository to deploy to the testbed infrastructure and may upload modified Subsystem packages.

Sample Maven-2 settings to deploy to this repository:

```
<distributionManagement>
  <repository>
    <id>xifi-sw-repository</id>
    <name>xifi-sw-repository-releases</name>
    <url>http://193.175.132.52:8081/artifactory/xifi-subsys-test</url>
  </repository>
  <snapshotRepository>
    <id>xifi-sw-repository</id>
    <name>xifi-sw-repository-snapshots</name>
    <url>http://193.175.132.52:8081/artifactory/xifi-subsys-test</url>
  </snapshotRepository>
</distributionManagement>
```

Repository: xifi-component-test

Scope: XIFI Components ready for testing in the testbed environment.

Purpose: Component maintainers should upload to this repository prior to request testbed maintainers or test owners to conduct tests on this Component. Test owners or testbed maintainers pull Components from this repository to deploy to the testbed infrastructure and may upload modified Components.

Sample Maven-2 settings to deploy to this repository:

```
<distributionManagement>
  <repository>
    <id>xifi-sw-repository</id>
    <name>xifi-sw-repository-releases</name>
    <url>http://193.175.132.52:8081/artifactory/xifi-component-test</url>
  </repository>
  <snapshotRepository>
    <id>xifi-sw-repository</id>
```

```
<name>xifi-sw-repository-snapshots</name>
<url>http://193.175.132.52:8081/artifactory/xifi-component-test</url>
</snapshotRepository>
</distributionManagement>
```

Repository: xifi-quarantine

Scope: Components and Subsystems under evaluation

Purpose: Node maintainers may use this repository to save snapshots of components or Subsystems that failed in the node infrastructures for further study by the Subsystem or component maintainer.

Sample Maven-2 settings to deploy to this repository:

```
<distributionManagement>
  <repository>
    <id>xifi-sw-repository</id>
    <name>xifi-sw-repository-releases</name>
    <url>http://193.175.132.52:8081/artifactory/xifi-quarantine</url>
  </repository>
  <snapshotRepository>
    <id>xifi-sw-repository</id>
    <name>xifi-sw-repository-snapshots</name>
    <url>http://193.175.132.52:8081/artifactory/xifi-quarantine</url>
  </snapshotRepository>
</distributionManagement>
```



Figure 62: Outline of an unscheduled maintenance procedure affecting a single infrastructure node

6 SUPPORT TO FI-DEVELOPERS

Section 6 defines the process of providing support related to the federation of nodes to FI-developers as provided by XIFI.

FIWARE has defined a number of email addresses that can be used by anybody outside (or even inside) the FI-PPP in order to request information or to obtain support. An overview of the topics and corresponding support email addresses is available on the FIWARE web³. They range from general questions about FIWARE to support requests regarding GEs, FIWARE Ops or FIWARE Lab as far as even requests regarding speakers or towards the FIWARE press office. This section here describes how the support process for FI developers that use FIWARE-Lab is realised – in the specific scope of the federated nodes that jointly form the infrastructural basis of FIWARE-Lab. This section does not cover e.g. any support regarding GEs (which can be deployed on any of the FIWARE Lab nodes) as this is handled in FI-Core and is not in the scope of XIFI.

It should be noted that, while the above section 5 on the maintenance procedures had a focus on roles, their tasks and interactions; the description of support to FI-developers in this section – as the reader will see – is structured according to escalation levels, i.e. Level-0 /-1 / -2 / -3 support. Of course, there are also roles defined for FI-developer support, but the overarching framework is built by the support escalation levels in which the various roles are involved that are necessary to implement this process. This is a small but noticeable methodological difference that should be mentioned.

6.1 Introduction to Support Levels

In this introduction we introduce the generic concepts related to helpdesk levels from 0 to 3. These definitions will be used and refined later in this section.

- Level 0 support – Automated or self-service solutions that users can access themselves without the aid of the Help Desk. These include automated password resets, Web sites for requesting ITIL (Information Technology Infrastructure Library) support and knowledge base lookup. Level 0 support is performed without the aid of an Help Desk individual.
- Level 1 support – a group of technicians filtering Help Desk requests and providing basic support and troubleshooting, such as password resets, giving break/fix instructions, ticket routing and escalation to Level 2 and Level 3 support. A Level 1 technician gathers and analyses information about the user's issue and determines the best way to resolve their problem. Level 1 may also provide support for identified Level 2 and Level 3 issues where configuration solutions have already been documented.
- Level 2 support is provided by a group of more specialized technicians. Level 2 generally handles break/fix, configuration issues, troubleshooting, software installations, and hardware repair. They handle escalated issues that Level 1 support is not capable of handling. Level 2 will sometimes escalate to Level 3, depending on the issue (see next paragraph for the escalation rules). Depending on the Help Desk organization, a level 2 technician may either 1) be limited to only solving known issues and escalate new issues to level 3; or 2) be authorized to research and implement fixes for new issues and only escalate to Level 3, if it is out of their skill set or ability to solve.
- Level 3 support: Generally, Level 3 support is a group of specialized technician performing troubleshooting, configuration, database administration, and repair for server, network, infrastructure, Data Centre, email, file shares, and other infrastructure issues. Besides always having the ability to deploy solutions to new problems, a Level 3 technician usually has the

³ <http://www.fiware.org/contact-us/>

most specialised expertise in a company and is the go-to person for solving difficult specific issues. In the context of FIWARE Lab, Level 3 support is provided by GE owners and XIFI FIWARE Ops owners.

6.2 Support Levels Applied in FI-WARE Lab Support

The basic roles that take part in FIWARE Lab support for FI-developers have already been defined in section 3.2. This section defines in more detail the Level 1 / Level 2 and Level 3 support process and flows.

- Level 0 support is provided in StackOverflow as decided in coordination with FIWARE.
- Level 1 support is provided by a helpdesk team. The team is in charge of filtering incoming requests, managing all requests, issues and problems coming from FI-Developers. Issues that can be solved by Level 1 are directly answered. An issue is passed from Level 1 to Level 2 or Level 3 helpdesk if it is related to:
 - 1 SW/HW or configuration problems related to a specific node that L1 helpdesk cannot solve: Level 1 passes the issue to the Level 2 team of the respective node.
 - 2 Software component problem related to a GE / GEi: Level 1 passes the issue to the respective Level 3 support.
 - 3 Requests that are not related to FIWARE-Lab, i.e. which were incorrectly addressed towards FIWARE-Lab helpdesk, will be ported to the correct helpdesk in charge of the topic.
- Level 2 support consists of the Node Help Desks that are run by node specific specialists in each of the nodes of the federation. The node helpdesk is in charge of the support of developer requests specific to a node. L2 helpdesk experts communicate directly with the issuer of the request, i.e. replies are sent directly and not via L1 helpdesk. However, the used tool allows L1 helpdesk to be informed and notified about the progress of the issue. There is a Level 2 helpdesk team at each node. This team takes care of all node-related issues of any complexity.
- Level 3 support has been defined in XIFI as support provided for Software Components, i.e. for the Generic Enablers developed and offered by FI-WARE / FI-Core partners. The support is provided by the GE owners for their specific GE(s).

It should be noted that Level 2 and Level 3 are not subsequent support levels in FIWARE Lab, where escalation occurs from one to the other, as these can both be invoked directly from Level 1.

In the following paragraphs some scenarios are presented to exemplify the related interactions between FI-Developers, FIWARE Lab Helpdesk Level 1 / -2 and -3 support.

6.3 Helpdesk Process Flows and Interaction with FI developers

This section describes the process and information flows in the helpdesk and the interaction with FI developers. An overview of the process flow and the correct interactions is depicted in Figure 63.

Helpdesk activities are supported by the **JIRA ticketing tool**. As a question of commodity, the same tool used by FI-WARE was decided to be used also in XIFI. Its flexibility and the possibility to add a large panel of add-ons permits to limit the restriction we could have in the definition of the issue reporting process. XIFI utilizes JIRA for both the interaction between FI-developers and helpdesk, as well as for interaction between all internal stakeholders in the helpdesk process.

When an FI-developer **submits a new support request**, either by sending an email or by filling a respective collector form, a ticket is automatically created in the Jira tool and Level-1 helpdesk is informed. For an overview of the Level-1 helpdesk team please see section 6.5. **Level 1 helpdesk** is then doing a number of checks as a first step:

- Is the issue in scope of FIWARE-Lab support, i.e. was the request sent to the correct helpdesk? If not, then the ticket is ported to the correct helpdesk, i.e. being moved out of the responsibility of FIWARE-Lab helpdesk. For example, queries of general nature that are not FIWARE Lab related will be dealt with by FI-WARE / FI-Core and a different Jira project was created for exactly that.
- Has the user or developer provided all (technical) information related to the issue that is required to provide support (e.g. to what XIFI node is the issue related)? If information is missing, Level-1 helpdesk will contact the FI-Developer to request the missing information.

Once it is clear that the request is relevant for FIWARE-Lab helpdesk and all required information is provided, Level-1 helpdesk checks if it can solve the issue themselves and answer directly.

- Is the question about general information that Level-1 helpdesk has available or could easily find out?
- Is the issue related to a known problem, e.g. already described and answered in the FAQs or already addressed in an existing Jira ticket?
- Does the question possibly require helpdesk to contact another expert in XIFI (e.g. the federation office) but the respective person is not a Jira user which would make it impossible to assign the Jira ticket to this person?

In all above cases Level-1 helpdesk will try to acquire the needed information in order to prepare the answer and then to send it to the FI-developer.

In case Level-1 helpdesk cannot answer the question it will assign the ticket to the respective node, i.e. **Level-2 helpdesk**. If the matter is clearly related to a specific node then it may however even be advisable not to try to answer by Level-1 helpdesk but to assign the ticket immediately to the respective node, as this ensures that the node gets aware of issues that developers have when using the node, which can help nodes to improve the quality of their services and node's operation.

If the FI-developer does not request support in email format but by filling an **issue collector form**, the FI-developer can indicate if he thinks that the issue is related to a certain node e.g. because he is running his experiments on that node. In this case the ticket is directly assigned to the node help desk (Level 2) of the respective node. The node helpdesk might have to reassign the ticket to the help desk of another node in case the developer indicated the "wrong" node, e.g. if a problem related to the Trento node got assigned to Berlin Level 2 support, Berlin must reassign the ticket to Trento. The node helpdesk might also have to reassign to Level 1 helpdesk if it is not clear what node is in charge, or if the issue is not related to a single node at all.

- We will offer support in English language in level 1 helpdesk. In Level 2 node helpdesk support could also be offered in a local language if this is of benefit to the local user

In case Level-1 or -2 helpdesk conclude that a request is rather related to a GE / GEi component (i.e. **Level-3 helpdesk** as per our definition) then the ticket could be either ported to the respective GE project in Jira, or it could be kept in FIWARE-Lab helpdesk and simply assigned to the expert in charge of providing support for the respective GE. The latter is in particular useful if the issue has also certain relevance for FIWARE-Lab.

In all cases, the FI-developer will receive a notification when the issue has been resolved, optionally with some supplementary information about the issue and its solution.

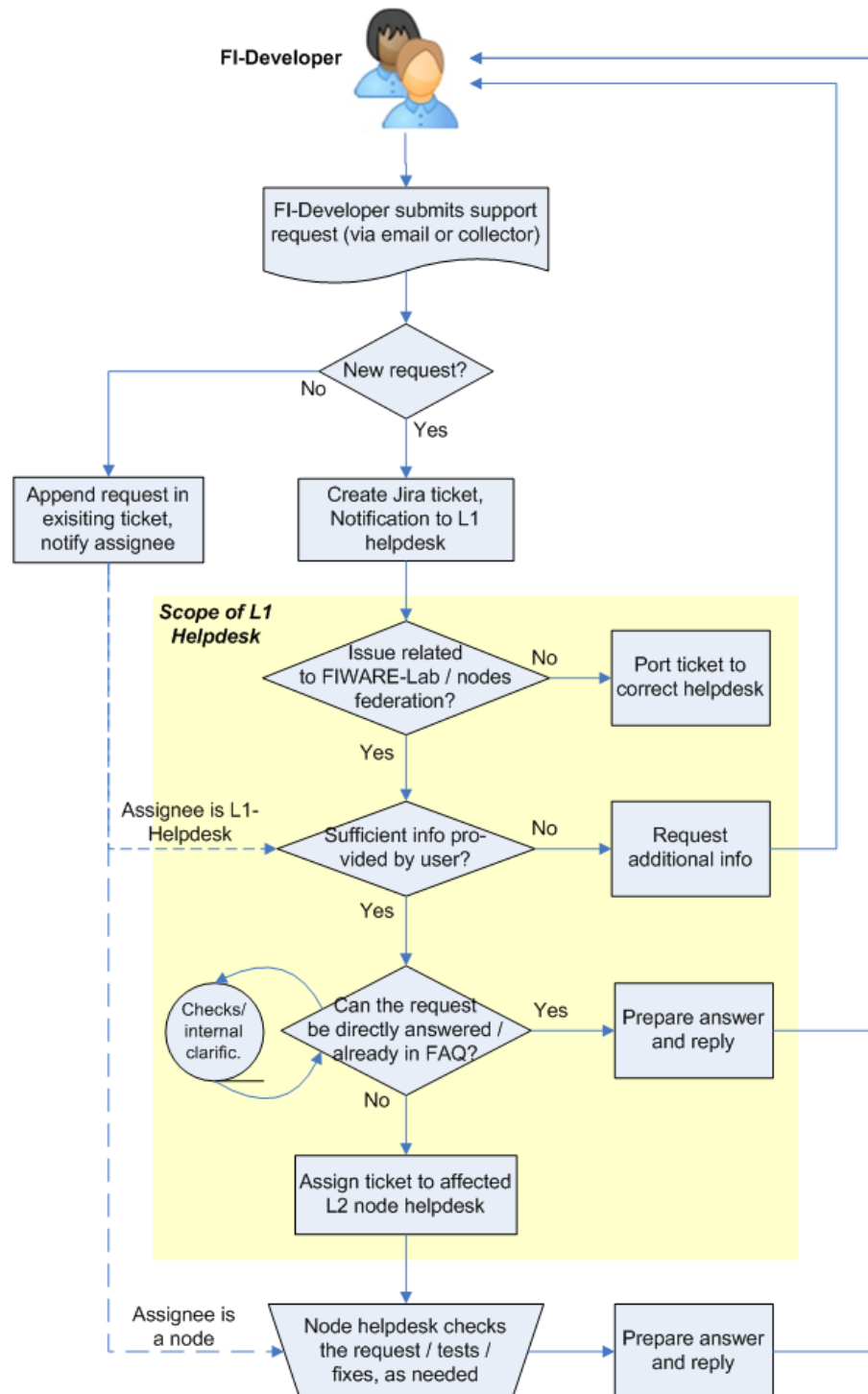


Figure 63: FIWARE Lab Level 1 helpdesk support process

6.4 JIRA Ticketing Process

As already introduced, helpdesk activities are supported by the JIRA ticketing tool.

Users, i.e. FI-developers have two options (channels) to submit a support request. The preferred form (from the perspective of helpdesk) is the JIRA collector. A JIRA collector is a form in which the user can fill in all required information in a structured form, which is directly submitted into JIRA. By requesting specific information to be entered in the form enables the assignment of the created ticket directly to the corresponding support team best suited to address the issue, i.e. to the respective node

region. The second option, alternatively to the collector, is to send an email to a support email address (fiware-lab-help@lists.fi-ware.org). Emails sent to this address are converted into a JIRA ticket by use of a dedicated JIRA plugin. For practical and license cost reasons developer do not get an account in Jira. Jira will automatically transform requests sent by email or submitted via a collector into a ticket.

Places from where support requests can be submitted

JIRA collectors are placed at the FIWARE Lab nodes status page⁴. The email contact option is announced in two main places: On the FIWARE Lab “Help & Info” page⁵, and on the FIWARE “Help/Contact” page⁶ (see Figure 64).

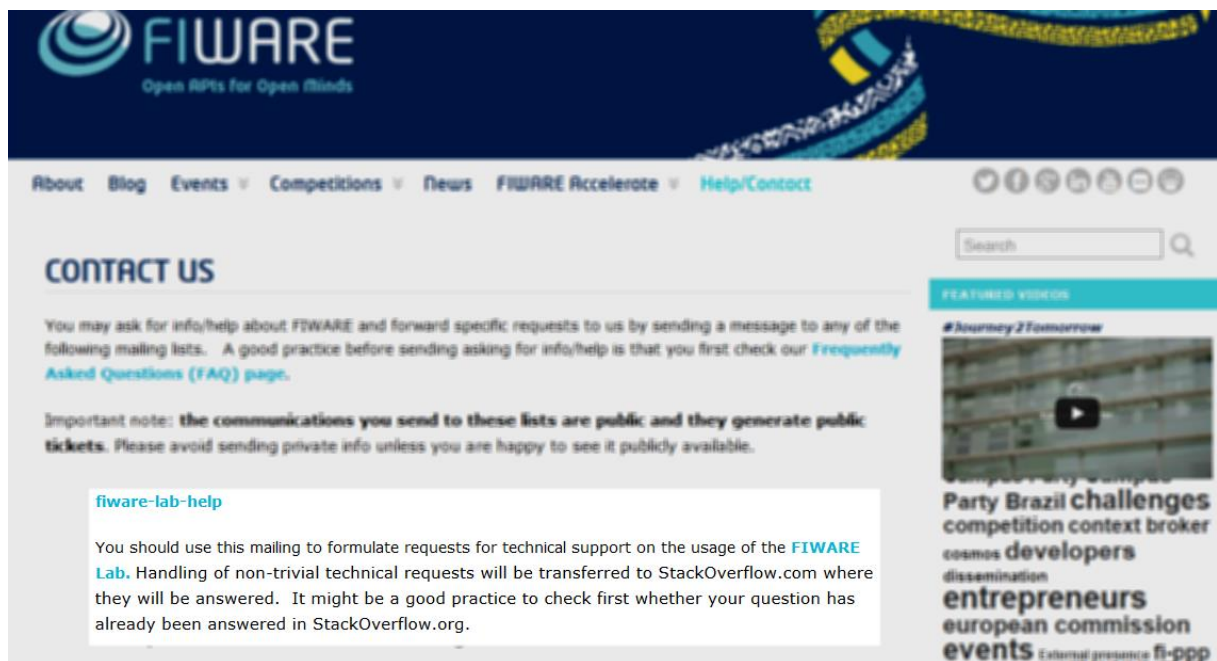


Figure 64: Pointer to Helpdesk on FIWARE Lab “Help & Info” page⁶

New tickets

When a developer submits a support request a new ticket is created:

- This ticket will be created and its status will be set to OPEN (=new)
- The issuing developer will be the Reporter (“FW External User”)
- Initially the ticket is “unassigned”, and the helpdesk person on duty will, among other things, need to make the correct assignment. However, if a concrete node was indicated in the collector form then the ticket is immediately assigned to the respective node helpdesk team.
- The initial priority is set to “Major” (as the machine routine cannot decide upon that)
- The title of the ticket will be the title of the email / of the collector form
- The description of the ticket is filled with the body of the mail / the respective content from the collector form.

There can be cases where a ticket requires activities of more than one infrastructure owner. In such case the activities can be either in *parallel* where new and separate tickets will be created for each involved

⁴ <http://status.lab.fiware.org/>

⁵ <http://help.lab.fiware.org/>

⁶ <http://www.fiware.org/contact-us/>

stakeholder (i.e. the ticket is “cloned”), and the ticket is solved when all the sub-activities are done. Alternatively, this could be handled by *sequential activities*. Then the last activity will solve the ticket. Typically parallel processing will be preferred as this saves time.

In general, ownership of a ticket reflects both responsibilities and activities. For example, if L1-helpdesk hands over responsibility to the helpdesk of a node, the ownership of the corresponding ticket reflects who (currently) is responsible for performing the support task.

Jira has been configured in a way that makes all tickets publicly available in Read-only mode⁷.

Status of a ticket

The status of each ticket informs of its current processing status. A new ticket gets the status OPEN. Every person in the helpdesk that is involved in processing a ticket needs to manually change the ticket status according to the progress of the ticket. An overview of the various statuses and transition that a ticket can have is shown in Figure 65.

- OPEN: A new ticket. Even if it has already been assigned by L1-Helpdesk to the actual assignee it remains OPEN until the actual progressing has started.
- IN PROGRESS: The actual addressing of the support request has started, e.g. by asking the developer for further details or by doing some preparatory check that are needed before a reply is sent to the developer. Also re-assignments to other persons can take place in this state.
- IMPEDED: The ticket is on hold; its further progress is impeded e.g. because information from the developer is missing.
- ANSWERED: A (first) answer has been sent to the developer that should answer the request. A request to the developer for further details is not considered as an answer in the sense here.
- CLOSED: If the helpdesk has completed the support the status can be set to CLOSED. This can be reached either after the issuer of the ticket has given his approval to the resolution, or simply after some period of time if the ticket issuer did not come back again, assuming his tacit approval. If the developer unexpectedly comes back again then the ticket could be re-opened going to status IN-PROGRESS. There can be two types of CLOSED tickets:
 - A “CLOSED and resolved ticket”, when a solution has been proposed and has been approved;
 - A “CLOSED but unresolved ticket”, corresponding to the case that no resolution has been provided.

Jira does not enable distinguishing between these two types.

⁷ <http://jira.fi-ware.org/>

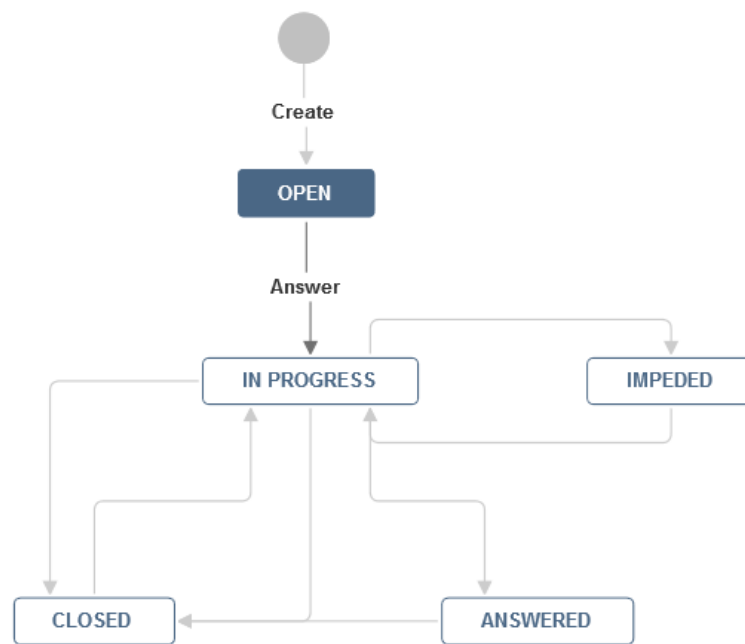


Figure 65: Ticket status tree

Priority of tickets

A priority can be assigned to each Jira ticket. When a ticket is created from an external support request the default priority it gets is “Major”, and the person processing the ticket decides whether this is appropriate or needs to be adjusted. In practice, within the federation helpdesk the priority feature is not heavily used as tickets are immediately processed anyway. Nevertheless, if it appears that a support request is critical or rather trivial then the priority is set accordingly. The overall helpdesk statistics show that 96% of all tickets are “Major”, 2% are “Minor”, and 1% is labelled “Blocker”.



Figure 66: Priority of Jira tickets

Notifications

Notifications are generated along with each support process instance. All notifications are created by Jira. A notification is issued each time a ticket is modified, e.g. when a comment is added, when the ticket is assigned to another person or when the ticket status is changed (e.g. ticket is closed).

Notifications are sent to all persons that are “watchers” of a ticket. Any person that has made changes to a ticket automatically becomes a watcher. This means in practice that anybody who has been involved in the processing of a ticket is notified of its further progress. In addition, any user in the Jira system can add himself manually to become watcher of a ticket or later unsubscribe (“unwatch”) again if no longer interested in the ticket.

The JIRA system is the only channel utilized for notifications between the stakeholders involved in the helpdesk. It is however not the only channel overall because external developers, i.e. those that request support, cannot be reached via this path. The reason is that external developers don’t have a Jira

account. Therefore, communication with external developers is done via email.

6.5 Responsibilities

Responsibilities have been fixed for all roles participating in FI-developer support. The following tables summarise the responsibilities of person involved:

Responsibility Assignment Matrix	
General coordination	Federico M. Facca (CREATE-NET) Miguel Carrillo Pacheco (TID)
Node coordination	Contact details are listed in section 3.1 of this document.
GEi	GE / GEi owners are identified in a list maintained by FI-WARE / FI-Core project, see the list at [23]
FIWARE Ops Support	Contacts are defined and available on the internal XIFI Wiki.
Reporting (JIRA statistics)	Miguel Carrillo (TID) and Uwe Herzog (EURES)
JIRA support (Systems Level)	TID Bitergium (FI-WARE partner, for support on Linux level)
JIRA support (Application Admin Level)	Florian Rommel (EURES), Manuel Escriche (TID)
Liaison with FIWARE Lab Portal	Fernando López Aguilar (TID)

Table 81: Overall responsibility assignment

Full name	Organisation
Miguel Carrillo	TID
Manuel Escriche Vicente	TID
Uwe Herzog	EURES
Florian Rommel	EURES
Marco Cipriani	TI
Daniele Santoro	CREATE-NET
Aristi Galani	UPRC
Sándor Laki	Wigner

Table 82: Level 1 Helpdesk team

6.6 JIRA Administration

The JIRA platform is hosted in one of the servers of the FIWARE Lab infrastructure, currently in the Spain node. FIWARE HELP-DESK is a JIRA public project (tracker) devoted to collecting incoming requests from FIWARE users. These requests are received on twelve email lists offered at <http://www.fiware.org/contact-us/> page. Each email list focuses on a specific topic.

- **fiware-general-help**, it is used to forward requests for non-technical, general info/help about FIWARE (e.g., How does FIWARE work? How can I join? What are FIWARE goals and value proposition?, ...).
- **fiware-acceleration-help**, it is used to forward requests on the FIWARE Acceleration

Programme.

- **fiware-press-req**, it is used to contact the FIWARE Press Office to get further info and assistance.
- **fiware-speakers-req**, it is used to organize an event where you wish that someone provides a speech on FIWARE.
- **fiware-feedback**, it is used to get your feedback about FIWARE, the FIWARE website, the FIWARE Catalogue or the FIWARE University.
- **fiware-tech-help**, it is used to formulate technical questions on the FIWARE platform, including requests for technical support or enhancements on reference implementations of FIWARE GEs (GÉris). It will provide a first-level technical support regarding FIWARE technologies. Handling of non-trivial technical requests will be transferred to StackOverflow.com where they will be answered.
- **fiware-lab-help**, it is used to formulate requests for technical support on the usage of the FIWARE Lab. Handling of non-trivial technical requests will be transferred to StackOverflow.com where they will be answered.
- **fiware-ops-help**, it is used to formulate requests for technical support on the usage of FIWARE Ops tools. Handling of non-trivial technical requests will be transferred to StackOverflow.com where they will be answered.
- **fiware-collaboration-req**, it is used to set up a liaison with FIWARE or wish to use FIWARE in other project.
- **fiware-smart-cities-req**, it is used for cities in order to transform them into a Smart City and connect to the FIWARE Lab and meet entrepreneurs to work with you.
- **fiware-open-data-req**, it is used if any Open Data initiative or project wish leverage on FIWARE and usage of the FIWARE Lab.
- **fiware-mundus-req**, it is used in order to provide expansion of the FIWARE Lab across multiple countries all over the world.

JIRA has configured a number of email handlers, which allow transforming emails into JIRA issues. They also allow to chain subsequent related emails as comments to the same issue. When a request arrives to the help-desk, it is assigned by the first support level, to whichever the support node it belongs to. When the request is assigned, JIRA send a notification to the group of people in that node.

JIRA has configured a number of email handlers, which allow transforming emails into JIRA issues. They also allow to chain subsequent related emails as comments to the same issue. When a request arrives to the help-desk, it is assigned by the first support level, to whichever the support node it belongs to. When the request is assigned, JIRA send a notification to the group of people in that node.



Figure 67: Issue creation

Once that each team receive the issue, they reacts by answering the request, and a dialogue is established until it's considered over. The following image shows us detail of the different issues that was created in the tracker regarding the different email used in the process.

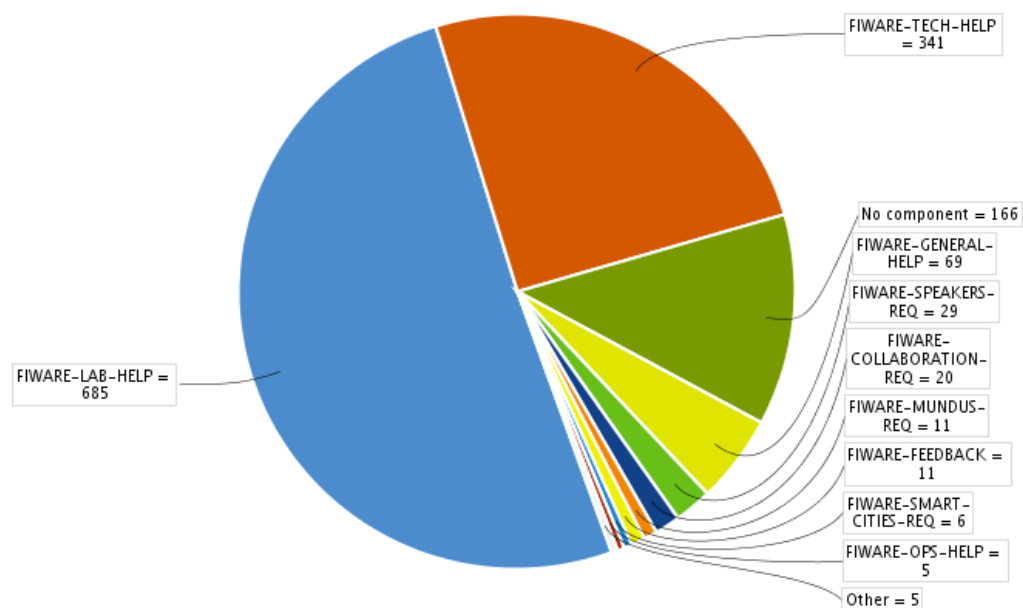


Figure 68: Different email lists in HELP-DESK

Each of the issue created in the HELP-DESK project can be monitoring in order to see the evolution in time of the resolution of it. JIRA provides several views on issues evolution; the example bellow shows last 30 days in the helpdesk.

Issues: 30 Day Summary

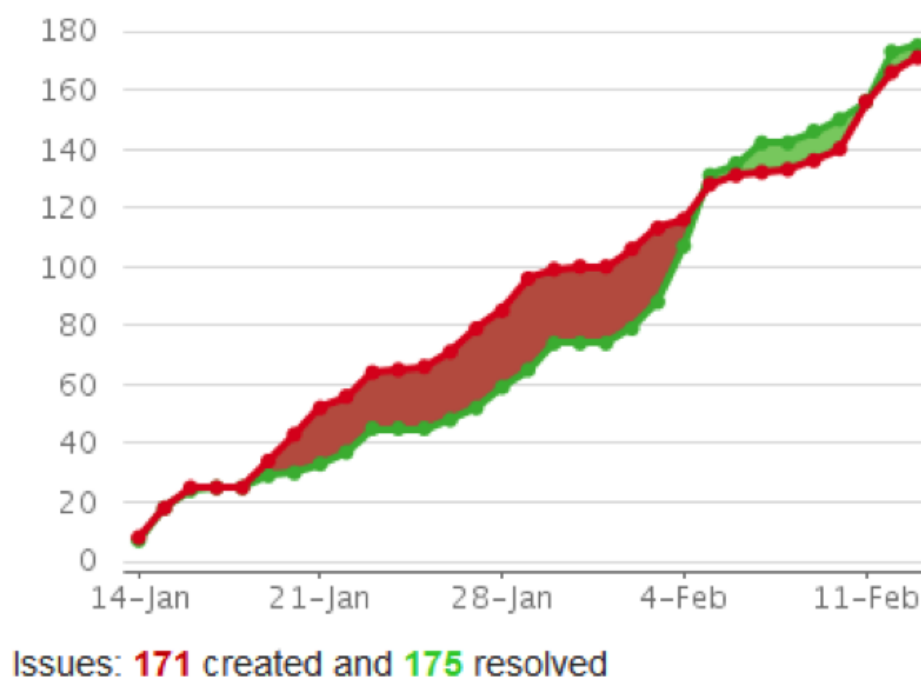


Figure 69: Issues created in the last 30 days

Additionally, each of the issue has a activity steam. The activity stream displays all incoming issues and their answers, which give us an overview of the evolution in time in the resolution of that specific issue. An example of that activity stream can be found in the following image:

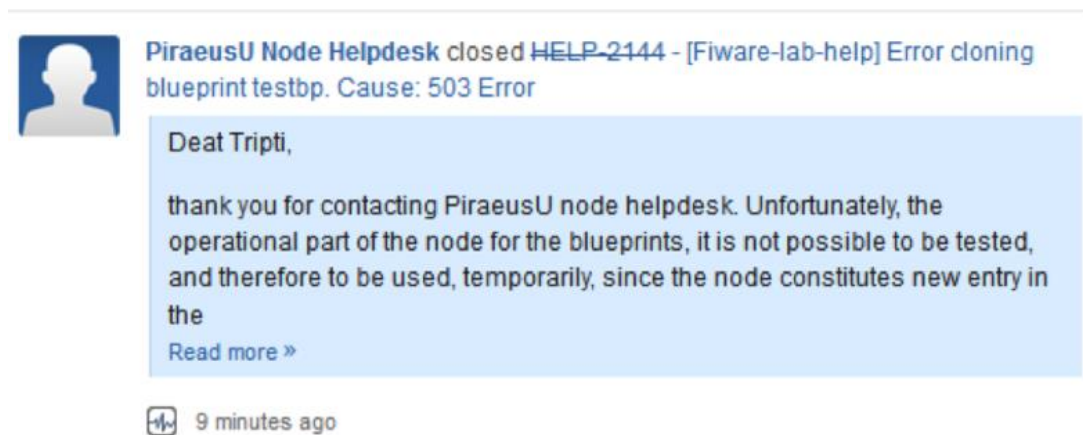


Figure 70: Activity stream

Last but not least, any tracker in Jira has to be defined a specific workflow in order to know the way in which each issue evolves during time. This workflow for the issues is displayed bellow. An issue is created with the Open status, afterwards the assignee move it to in progress. If anything blocks it, it becomes impeded until it's released and can be answered, and finally closed. There's a shortcut for issues which are dismissed. Issues can also be re-open in case the user keeps asking on the same topic.

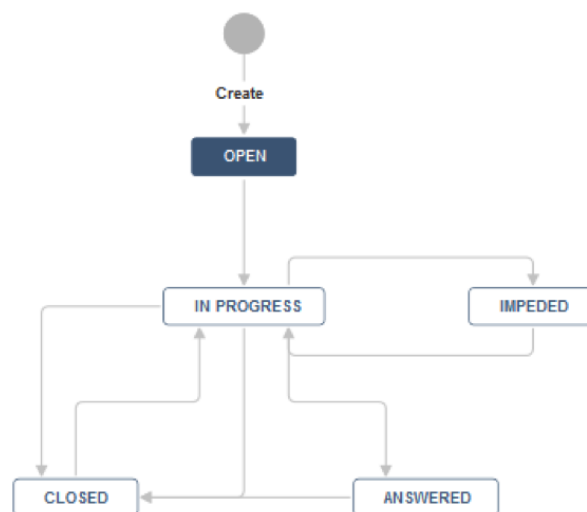


Figure 71: Workflow of the HELP-DESK issues

6.7 Reporting (JIRA statistics)

Jira has been setup by FI-WARE and has been in use for quite some time. However, Jira has started to be used for managing external support requests from FI-developers towards FIWARE-Lab only in September 2014. More precisely, on 24-Sep-2014 Jira was initially filled with 183 tickets sourced from support request emails received since February 2014. From that point in time all new support requests fed immediately into Jira, creating a ticket. As of 4 February 2015, **847 requests** have been received from developers sent to the fiware-lab-help mailing list and creating a ticket in Jira. However, not all requests are indeed relevant for FIWARE-Lab Helpdesk, mostly because sometimes developers

do not send their request to the correct mailing list. For example, GE related issues should be sent to the fiware-tech-help mailing list but are quite frequently also sent to fiware-lab-help. Therefore, there is a field in each ticket record where L1-Helpdesk should indicate if the ticket is indeed in the scope of FIWARE-Lab Helpdesk and if so set the correct parameter (component = FIWARE-LAB-HELP). **683 tickets** (out of the 847 in total) have been assigned to this category so far.

Figure 72 shows how many tickets have been created since September 2014 in each month. On average these have been between 5 and 14 tickets per working day.

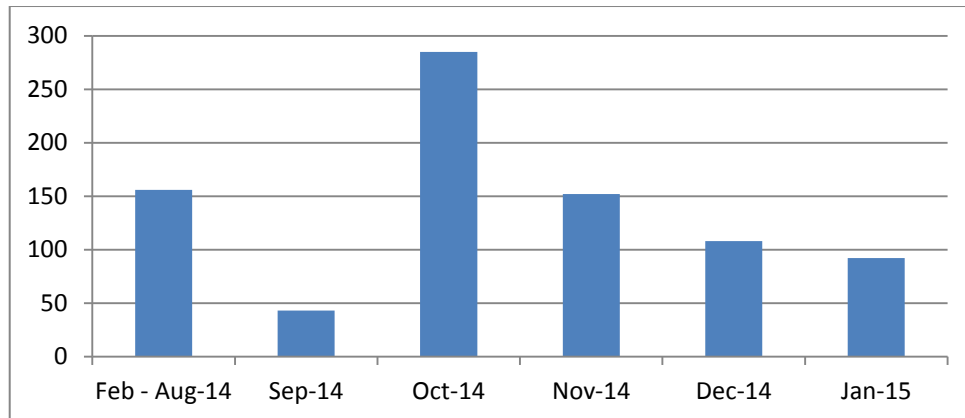


Figure 72: Number of support requests received / Jira tickets created

An analysis of the 683 tickets was done in order to find out towards whom they were addressed, and who got assigned to answer the request and/ or to resolve the issue. As explained before, L1-Helpdesk is tasked to check all tickets and answers them directly if possible, and only issues that cannot be answered are assigned to the respective federation node or elsewhere. Below table gives an overview of the number and status of all tickets received so far and assigned to L1-Helpdesk or any of the federation nodes, a snapshot as of 4 February 2015:

	Open	In progress	Answered	Dismissed	Closed/Done	Total
L1-Helpdesk	4	2		2	107	115
Berlin	2		6		6	14
Brittany					5	5
Budapest	3	1			9	13
Crete						0
Ghent					1	1
Karlskrona						0
Messina						0
PiraeusN					1	1
PiraeusU				3	7	10
Poznan					2	2
Prague					4	4
SophiaAntipolis	1	1				2
Spain					10	10
Stockholm/Hudi	5	1			1	7
Trento	2	2			10	14
Volos						0
Waterford						0
Zurich	2	1				3
	19	8	6	5	163	201

Table 83: Overview of status of all tickets assigned to L1-Helpdesk and L2-nodes-helpdesks

Above table shows that **201 tickets** were addressing issues related to the federation of nodes, of which 115 were directly handled by L1-Helpdesk. 86 tickets were assigned to the L2-Helpdesk of the nodes.

In terms of **timeliness** of responding to developer requests, L1-Helpdesk is operating Monday – Friday from 9:00-17:00. There is a specific person from among the L1-Helpdesk team on duty on every day. The person on duty should ensure that at the end of the day processing of all new tickets has started, and that the external developer was sent at least an interim notification in cases where a full answer could not be given immediately on the support request.

In order to get an idea of the **effort that is required** for processing the above tickets, it should be noted that in most cases there are two or more people involved in dealing with a ticket. For example, every ticket that was assigned to a node went through the hands of L1-Helpdesk before. Also, often it takes one or two interim “hops” until the actual expert to answer the request is identified. Moreover, there are often several emails going between an external developer and helpdesk or node on each issue. These are all recorded in one single ticket, so the level of interaction is higher than what the sole number of tickets does suggest.

In addition to above table, all tickets received that were not in scope of FIWARE-Lab, i.e. the nodes federation, had to be assigned to the respective expert in charge, most often a GE owner, and these tickets are not included in the overview table above. Volume-wise these must have been in the range of the difference between the 683 and 201, i.e. more than 450 tickets. It should be noted, however, that it was not L1-Helpdesk alone that dispatched all the 450 tickets. Basically every expert in FIWARE who has some responsibility in providing user support has a Jira account can access the tickets created in fiware-lab-help. GE owners for example regularly scan incoming tickets also in fiware-lab-help and assign them to themselves immediately if their GE is affected, which means that such tickets do not require involvement of L1-Helpdesk.

A conclusion that could be drawn from this analysis is that so far the level of workload for L1-Helpdesk and the nodes for supporting developers is manageable. As Figure 72 shows, the number of tickets is even decreasing since its peak in October 2014, although it is expected that it will increase again once the Phase 3 SMEs will fully start activities. The statistics also indicates that the need for support in terms of the federation of nodes is much lower compared to the needed support for GEs, which is not even fully captured by above figures given that there are further channels through which GE support can be requested.

6.8 FAQ and Beginners Guide

After initial discussion in D5.3 [7], it was decided in coordination with FIWARE that Stackoverflow should be used as the tool where questions should be asked and where answers would be given (see info at <http://www.fiware.org/contact-us/>).

7 REPORT ON ACTIVITIES PERFORMED DURING THE EXTENSION PERIOD (M24-M30)

The federation operation by XIFI, to better cover the time laps between the end of XIFI and the FIWARE Open Call, was extended as agreed with the European Commission of additional 6 months, i.e. till 30th September 2015.

During this period, XIFI continued to support the management of the FIWARE Lab and enacted a transition toward the corresponding activities within FI-Core project. In general, as documented in the next sections, during the period most of nodes improved in their reliability and usability (except for few nodes that clearly underperformed almost during the period) thanks to a number of actions, such as:

- Introduction on usage limitation that limited abuse and overload of resources;
- Upgrade to more stable releases of OpenStack (IceHouse or Juno);
- Introduction of Karma points as positive feedback incentive towards FIWARE Lab Nodes;
- Supported the users' migration from discontinued nodes (i.e., Berlin, Stockholm, Waterford) on 30th September to a new node.

The following sections include statistics on performed activity in the period **from 1st April 2015 to 30th September 2015**, with focus on:

- **Level 1 Helpdesk statistics**, with more than 300 request served (cf. Section 7.1);
- **Community Account management**, with more than 500 requests served (cf. Section 7.2);
- **Node performance** as measured by Karma points in the covered period (cf. Section 7.3).

7.1 Level 1 Helpdesk support

This section provides a short report on the Level 1 Helpdesk statistics during the extension period. The main task of the Level 1 Helpdesk is to run a first analysis on tickets issued by FIWARE Lab users, provide a resolution to the user issues, if possible, if not redirect the issue to the appropriate Level 2 support mechanism (i.e. Generic Enablers owners or FIWARE Lab nodes).

During this period, the Level 1 Helpdesk has been run in collaboration with FI-Core. In particular Engineering, as leader of the FIWARE Lab Operation activities in FI-Core, started to collaborate to the management of the helpdesk.

Table 84, reports the Level 1 Helpdesk schedule during the period from 1st April 2015 to 30th September 2015.

	Monday	Tuesday	Wednesday	Thursday	Friday
6-10 Apr 2015	TID (FI-Core)	Create-Net	TID (FI-Core)	TI	Wigner
13-17 Apr 2015	TID (FI-Core)	Create-Net	TI	TI	Wigner
20-24 Apr 2015	TID (FI-Core)	Create-Net	TI	UPRC	Wigner
27 Apr - 01 May	TID (FI-Core)	Create-Net	TID (FI-Core)	UPRC	Wigner
04-08 May 2015	Create-Net	Create-Net	Create-Net & TI	TI	Wigner
11-15 May 2015	ENG (FI-Core)	UPRC	TI	Create-Net	Wigner
18-22 May 2015	ENG (FI-Core)	UPRC	TI	Create-Net	Wigner
25-29 May 2015	ENG (FI-Core)	Create-Net	TI	Create-Net & TI	Wigner
01-05 Jun 2015	ENG (FI-Core)	ENG (FI-Core)	Create-Net	TI	Wigner
08-12 Jun 2015	UPRC	Create-Net	TI	TI	Wigner
15-19 Jun 2015	ENG (FI-Core)	Create-Net	TI	FI-Core	Wigner
22-26 Jun 2015	ENG (FI-Core)	Create-Net	TI	FI-Core	Wigner
29 Jun - 03 Jul 2015	ENG (FI-Core)	Create-Net	ENG (FI-Core)	ENG (FI-Core)	Wigner
06-10 Jul 2015	UPRC	Create-Net	TI	FI-Core	ENG (FI-Core)
13-17 Jul 2015	UPRC	ENG (FI-Core)	TI	Wigner	ENG (FI-Core)
20-24 Jul 2015	TI	ENG (FI-Core)	TI	Create-Net	Wigner
27-31 Jul 2015	TI	Create-Net	ENG (FI-Core)	Wigner	ENG (FI-Core)
03-07 Aug 2015	ENG (FI-Core)	ENG (FI-Core)	ENG (FI-Core)	Wigner	Wigner
10-14 Aug 2015	Create-Net	Create-Net	Create-Net	Create-Net	Create-Net
17-21 Aug 2015	Wigner	Create-Net	Create-Net	TI	TI
24-28 Aug 2015	Create-Net	UPRC	TI	UPRC	ENG (FI-Core)
31 Aug - 04 Sep 2015	ENG (FI-Core)	Create-Net	TI	Create-Net	Wigner
07 Sep - 11 Sep 2015	Create-Net	Create-Net	TI	UPRC	ENG (FI-Core)
14 Sep - 18 Sep 2015	Create-Net	Create-Net	TI	ENG (FI-Core)	Wigner
21 Sep - 25 Sep 2015	Create-Net	Create-Net	TI	UPRC	Wigner
28 Sep - 02 Oct 2015	Create-Net	Wigner	TI	ENG (FI-Core)	Create-Net

Table 84: Level 1 Helpdesk schedule

During the covered period (1st April 2015 to 30th September 2015) more than 300 tickets have been managed, with an average of 2,3 ticket per working day. The average decreased compared to the previous period, and this may be interpreted as an improvement of the behaviour of the FIWARE Lab and of the completeness of the documentation. Table 85 provides detailed information on the number and status of all tickets received during the extension phase as for 30th September 2015.

	Open	In progress	Answered	Dismissed	Closed/Done	Total
L1-Helpdesk					180	180
Berlin					9	9
Brittany		2	1		31	34
Budapest			1		14	15
Crete						0
Gent			1		1	2
Karlskrona					3	3
PiraeusN					5	5
PiraeusU		1			9	10
Poznan			2		4	6
Prague					3	3
SophiaAntipolis					1	1
Spain					5	5
Stockholm/Hudiksvall		1				1
Trento					28	28
Volos					8	8
Waterford			1			1
Zurich					11	11
	0	4	6	0	312	322

Table 85: Number and status of all tickets received during the extension phase

Above table shows that **312** tickets were addressing issues related to the federation of nodes, of which **180** were directly handled by **Level 1 Helpdesk** and **142** tickets were assigned to the **Level 2 Helpdesk** of the different FIWARE Lab nodes.

It should be noted that the total number of requests created during this period, compared to the period from 1st October 2014 to 31st March 2015, decreased of around **10%** and the average of Closed requests is better (**96%** versus 81%)

A conclusion that was already proposed at M24 and that was confirmed during the 6 months extension is that the level of workload for Level 1 Helpdesk and the nodes for supporting developers is manageable. For a detailed report on the organisation of Helpdesk and lesson learned, the reader should refer to Section 6.

7.2 Community Account management

This section provides an overview of the community account management activities performed during the covered period (1st April 2015 to 30th September 2015) as by new policies introduced in February and discussed in Section 4.6. XIFI supported the overall management of the process in collaboration with FI-Core and the FIWARE coaches.

In particular, users while registering provides information on their preferred node and their relationship with FIWARE (e.g. a startup part of an accelerator). Accordingly, account requests are

validated by FIWARE coaches or by the FIWARE Lab Managers (in case the requester is not part of an accelerator programme) and then enacted by FIWARE Lab nodes operators.

As showed in Table 86, during this period more than 500 community account requests were processed, being Spain, Brittany and Trento the three most requested and active nodes.

	Done/Closed	In progress
Berlin	35	0
Brittany	115	1
Budapest	25	1
Crete	3	0
Gent	11	0
Karlskrona	12	0
PiraeusN	21	0
PiraeusU	14	0
Poznan	23	2
Prague	8	1
SophiaAntipolis	3	0
Spain	166	1
Stockholm/Hudiksvall	9	1
Trento	69	0
Volos	6	0
Waterford	4	3
Zurich	19	0
Total	543	10

Table 86: FIWARE Community Account upgrades as of 30th September 2015

As of 1st October 2015, the following nodes will be discontinued:

- Berlin
- Stockholm
- Waterford

To support Community Account migration, the following process was implemented in the month of September:

- Step 1. Inform FIWARE Lab users (with two reminders). The message sent included request for information on migration needs, to allow the operators to support the process.
- Step 2. Affected nodes (Berlin / Stockholm / Waterford) got in contact with users that replied to the message and, according to the user's reply:
 - *Need support:* coordinated the migration with the user and the new node. Added the user to the new node in IDM and informed him that he has been granted access to the

- new node. Asked the new node admins to apply quotas as the ones he had in the discontinued node.
 - *No need of support*: added the user to the new node in IDM and informed him that he has been granted access to the new node. Asked the node admins to apply quotas as the ones he had in the current node.
- Step 3. Notify users that didn't reply, that if they don't take actions before 30/09/2015 their virtual machine will be removed. (Providing instructions on how to download images in case of need).

7.3 Node performances (Karma points)

During the month of April, following a common practise in FI-Core to measure performance of development activities, the FIWARE Lab Management team introduced the concept of FIWARE Lab Karma points. The purpose of the points is to provide an accurate and objective evaluation of node performances and provide a positive feedback mechanism to encourage better performances by FIWARE Lab nodes. The Karma point computation is based around three main criteria:

- sanity check performed by FI-Health tools (e.g. number of successful functional test versus failed ones),
- node compliancy with FIWARE Lab requirements (e.g. complete support of monitoring stack),
- resources availability and usage (e.g. number of resources availables and usage ration).

This section gives the evolution of historical data based on the Karma point for each node from **M24** to **M30**. In general, it is possible to observe that the introduction of the Karma points improved the overall behaviour of the Lab and – except for some disruptive incidents –, the overall Karma over the time stabilized toward the top of the scoring board (except for few nodes that clearly underperformed almost constantly during the period: SophiAntipolis, Stockholm and Waterford).

The following paragraphs present the Karma points evolution for each node. The paragraphs includes as well additional information provided by the node operators, such as maintenance periods, upgrade activities (e.g. OpenStack migration, etc) that may impact the availability of the node and, as a consequence the Karma point.

7.3.1 Berlin

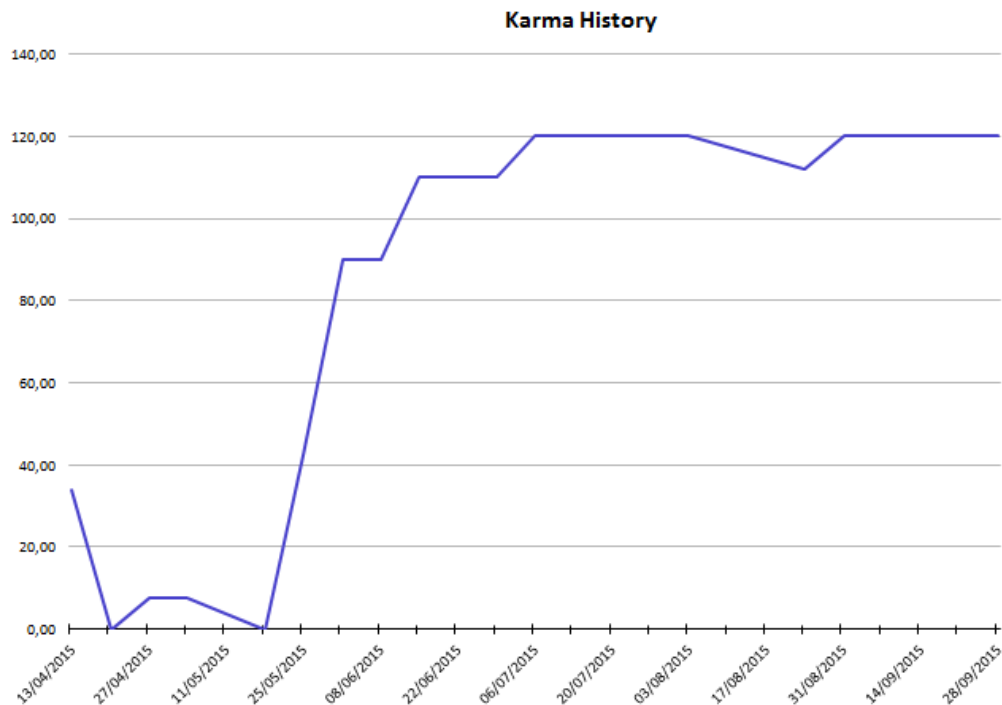


Figure 73: Karma history – Berlin Node

The Berlin node was one of the initial five nodes that had deployed the OpenStack Grizzly release. During the project it was decided that additional nodes has to deploy at least Icehouse and existing nodes need to upgrade their infrastructure accordingly. The Berlin node decided to do an onside upgrade to the OpenStack Juno release, which means that the Grizzly node was still operational while additional servers where installed and configured for the new Juno release.

The installation of the additional nodes started beginning of April. There were some configuration changes between the OpenStack releases which caused a delay to get the Juno release fully operational. During the end of April until the beginning of May the existing instances have been migrated from the “old” node to the “new” Berlin node. At the end of May the Berlin node increased the IP pool of public IP addresses from a /27 to a /26. Additionally, the node was providing historical monitoring data so that the karma score was increasing again at beginning of June.

Since the beginning of June the Berlin node is constantly operational reliable and therefore got the score of 120 karma points from there on.

7.3.2 Brittany

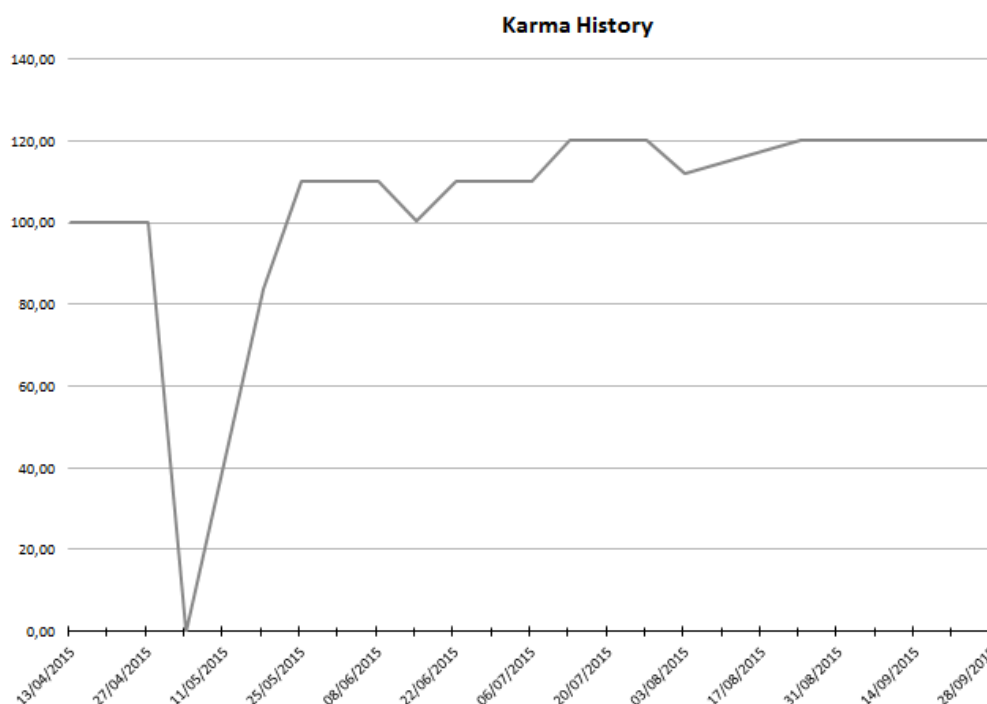


Figure 74: Karma history – Brittany Node

The Brittany node performed quite constantly during the period, with the exception of the first week of May, during which the migration of the node from Grizzly to Juno was performed.

7.3.3 Budapest

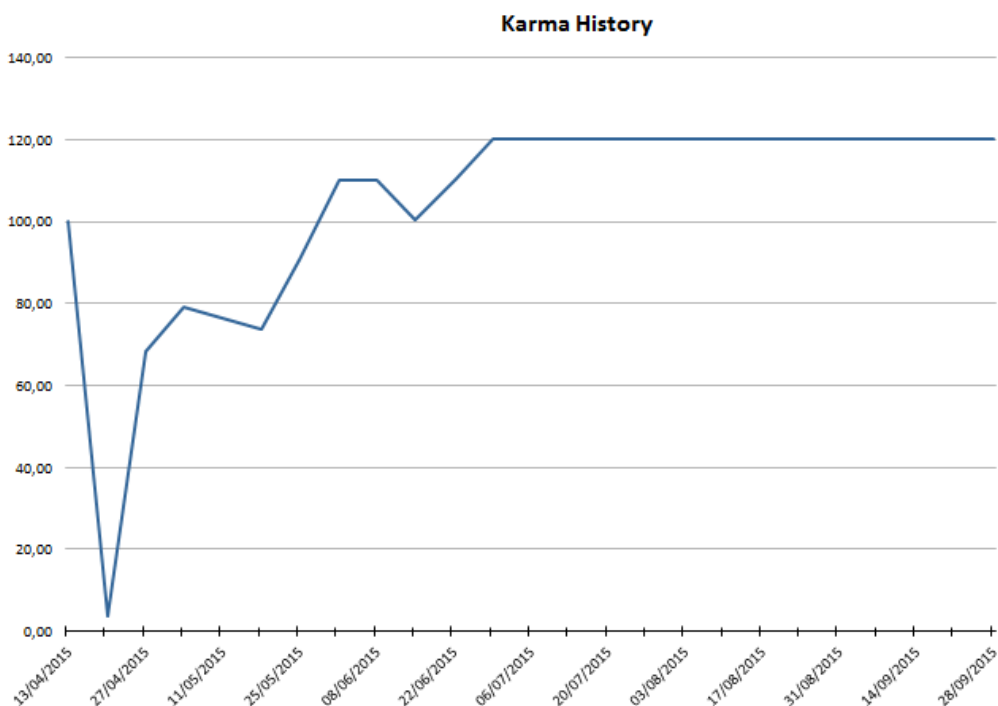


Figure 75: Karma history – Budapest Node

Budapest node joined XIFI consortium through the first open call in April 2014 and deployed an OpenStack Grizzly release. During the project it was decided that nodes have to deploy at least Icehouse and existing nodes need to upgrade their infrastructure accordingly.

In April 2015, we migrated to the IceHouse release, causing some temporal outages and failed tests (see the figure above). After the migration, the stability of Budapest node has increased a lot and the node reached the highest karma point in a short notice. The number of users is increasing continuously without major issues reported.

Budapest node is one of the first FIWARE Lab sites that offers Object Storage services to Community Users, and fulfil all the 26 tests of weekly sanity checks. The node was extended by additional computational capacity (+68 cores), resulting in 200 CPU cores dedicated to FIWARE Lab users. The preparation of OpenStack Kilo environment was started in August 2015.

7.3.4 Crete

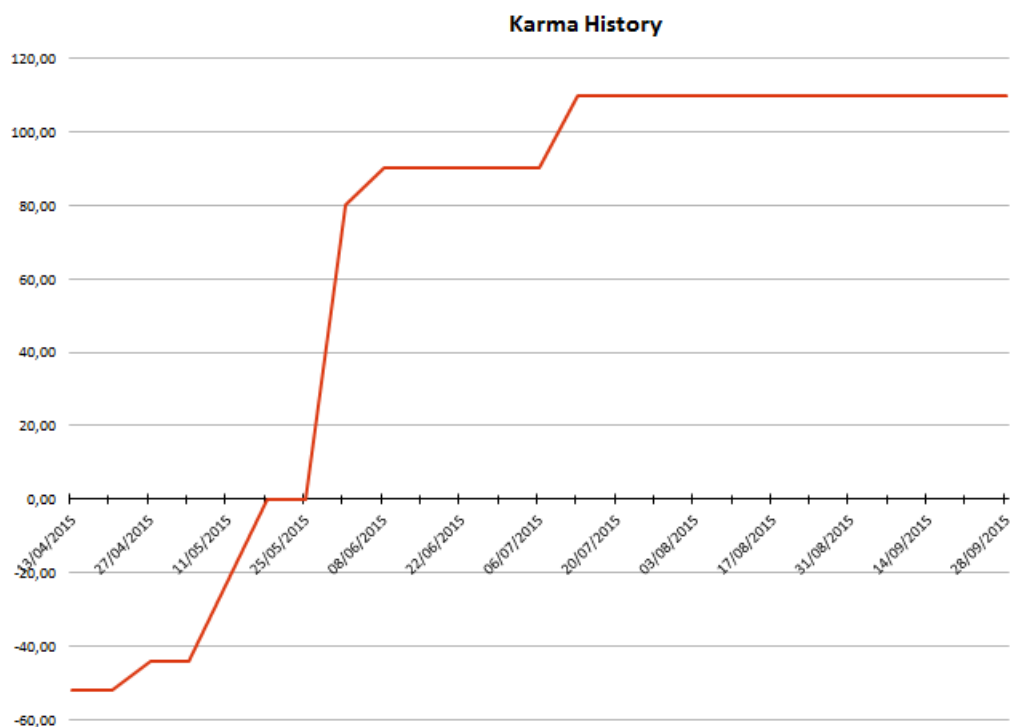


Figure 76: Karma history – Crete Node

The Crete node joined XIFI Federation on the 29th January, 2015, deploying a High Availability platform running on Openstack Grizzly over Ubuntu 12.04 LTS and KVM as hypervisor. Since then the node started facing a lot of instability problems on certain Openstack agents caused mainly by the High Availability services and also network problems caused by security holes, allowing users from third parties to attack the local node. This situation brought out low scores in the Karma calculations for the node.

On 29th May, 2015, the Crete node migrated from Grizzly to Juno environment (based on a single controller) and gradually incorporated all monitoring components of the XIFI project (including historical data).

Since then the Crete node is quite stable, fully operational and reliable as depicted in the above Karma History diagram.

7.3.5 Gent

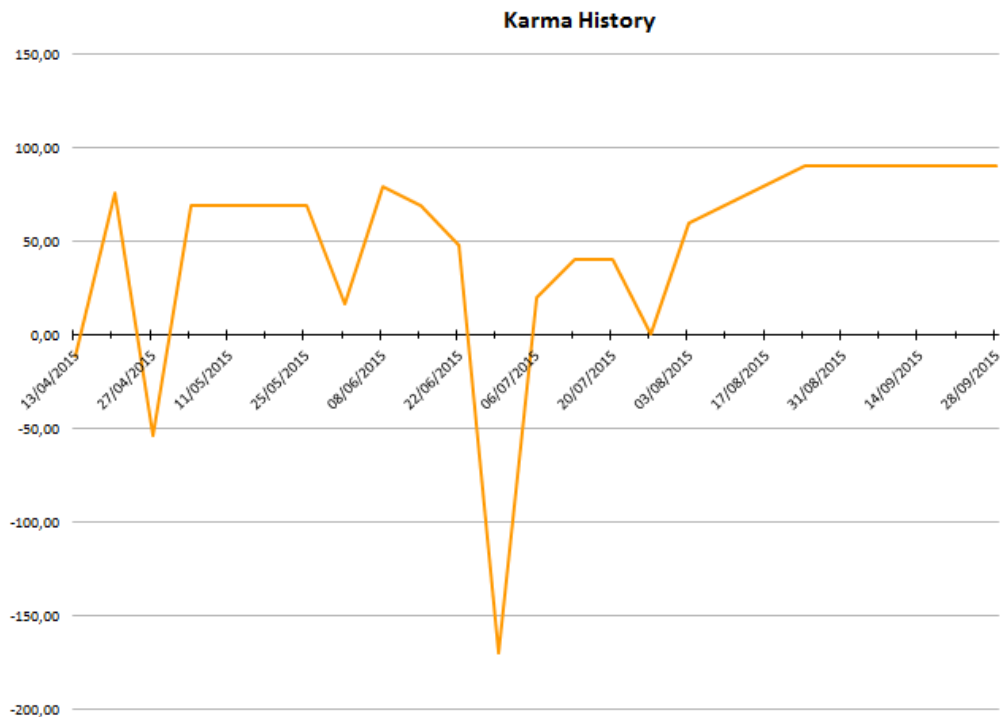


Figure 77: Karma history – Gent Node

The Gent node has been focused providing a stable platform by performing the right adjustments using scripts to ensure a pro-active monitoring in case of components failure while meeting user requirements and special demands as the use of personalized images with the platform.

Since the beginning after the launch of the platform online, users exhausted the availability of public floating IPs. For that reason, at the end of April and upgrade of the public floating IP pool was made. In fact, we also dedicated the public IPs reserved for the sanity tests to service more users, causing degraded performances on the karma points.

Last week of June, the datacenter suffered a massive failure on the cooling system causing the entire shutdown of all the servers and network for security reasons. The services were down for several days, that explains the bottom peak of the karma table, until the cooling system was fixed and restored. After that, operations were restarted normally without data losses.

7.3.6 Karlskrona

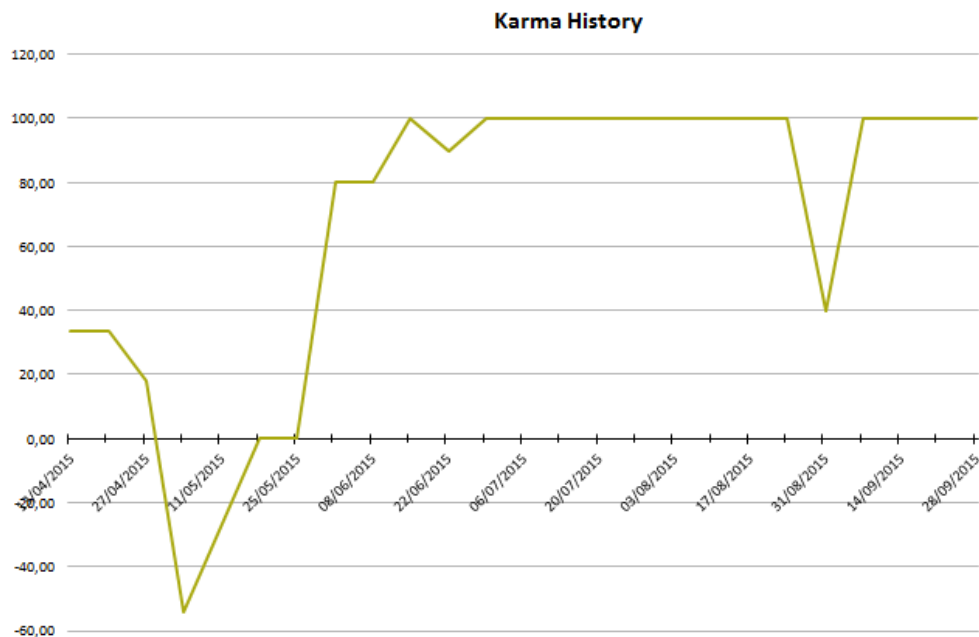


Figure 78: Karma history – Karlskrona Node

The Karlskrona node joined the XIFI federation through the first open call in April 2014 and deployed an OpenStack Grizzly release later that year. During the project it was decided that nodes have to deploy at least Icehouse and existing nodes need to upgrade their infrastructure accordingly. Our IceHouse migration begun in the end of April.

Unfortunately, at the same time the federation decided to move to the new IdM (Keystone). The migration to the new IdM proved to be more difficult than anticipated by the FIWARE engineers. In particular, there were specific configuration changes required for the new IdM that were not communicated or described properly. This affected many FIWARE Lab nodes. Since the Karlskrona node was in migration to IceHouse, the Grizzly deployment was disabled at the time. Consequently, the site was not operational at the time as can be seen in large Karma dip in the graph. We resolved the IdM issues around middle of May along with other FIWARE Lab nodes and the IceHouse deployment became operational.

The Karma dip in the beginning of September was caused by exhaustion of the pool of public IP addresses. We have cleaned up resources and returned unused IP addresses to the pool. Also, we are considering various alternatives to diminish the occurrence probability of these events in the future.

7.3.7 Piraeus

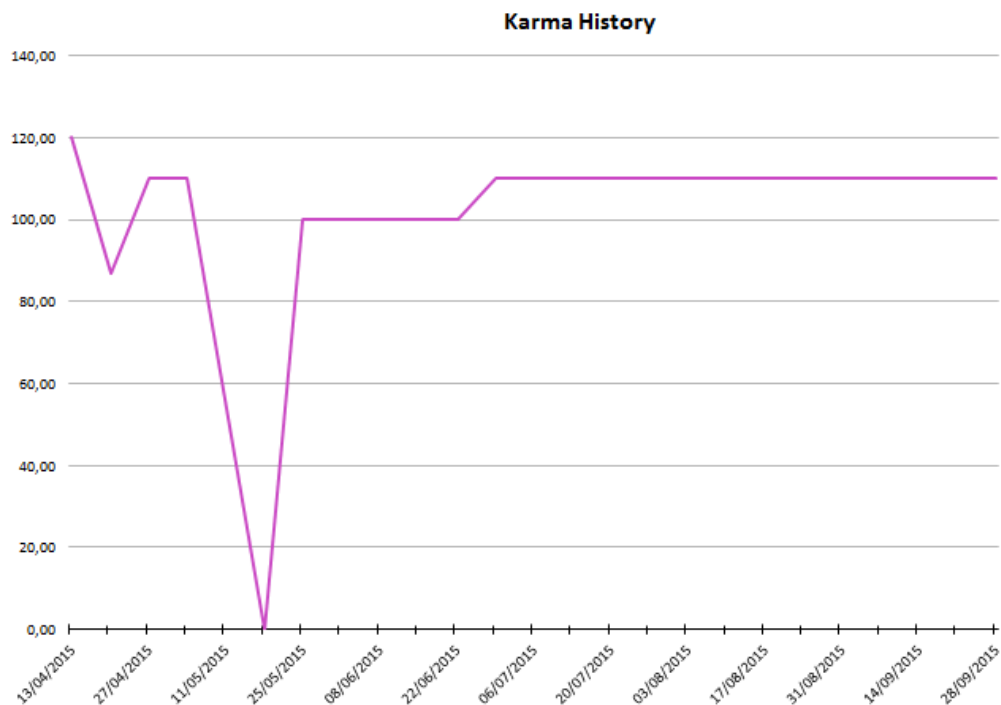


Figure 79: Karma history – Piraeus Node

PiraeusN node went through three phases of deployment. First deployed using Grizzly and then we were the first node to deploy Icehouse. We faced an unrecoverable error with the Openstack Icehouse High Availability deployment we had in place until May. This error resulted in a full rebuild of the node this time using Openstack Juno. The only low score of our node reflects the calculation made on Monday after the weekend we spent rebuilding everything.

After that our node is stable and the increase in the score reflects the number of VM's active in the node.

7.3.8 Piraeus

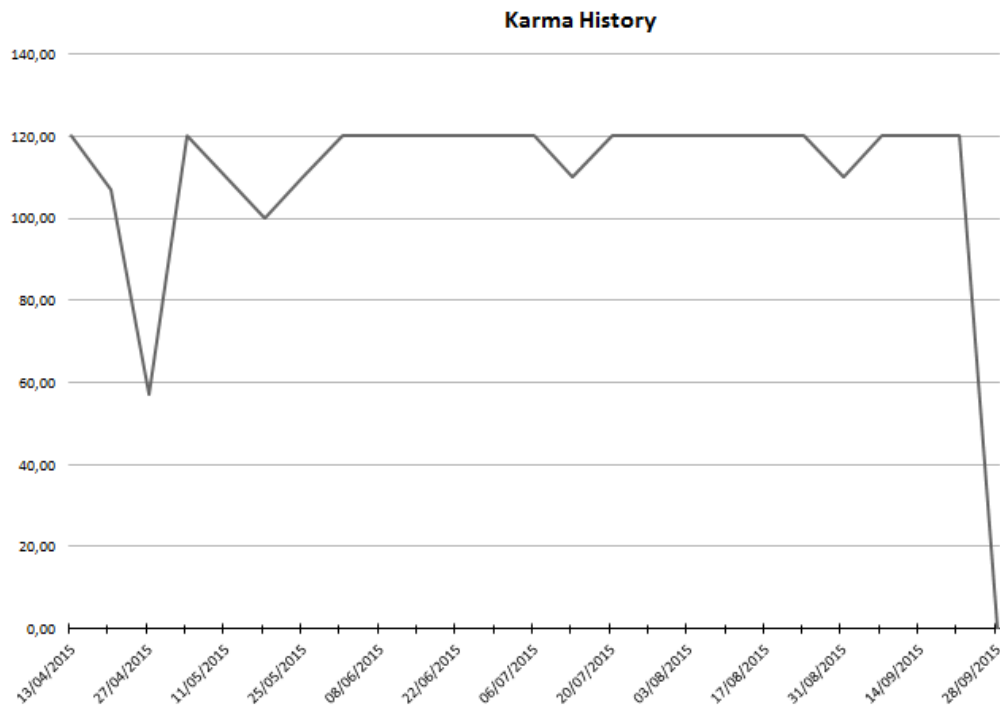


Figure 80: Karma history – Piraeus Node

The node has three low scores, the first one due to connectivity issues of sanity check tool that resulted in false report results for all nodes, and the other two due to a temporal problem with the power supply of the physical infrastructure, that caused a temporal unavailability of the network connectivity for the physical nodes. The last week of september there was an issue - that did not depend on the node- with the images uploaded on the node. The problem was about the usage privileges that for some reason due to the images upload from the automated FIWARE rsync process were changed. This issue was solved by 29/09.

Except the aforementioned cases, PiraeusU node is a stable, fully operational node, which, in order to increase its availability, increased the size of the IP pool of public IP addresses offered to users with 64 more public IPs v4.

7.3.9 Poznan

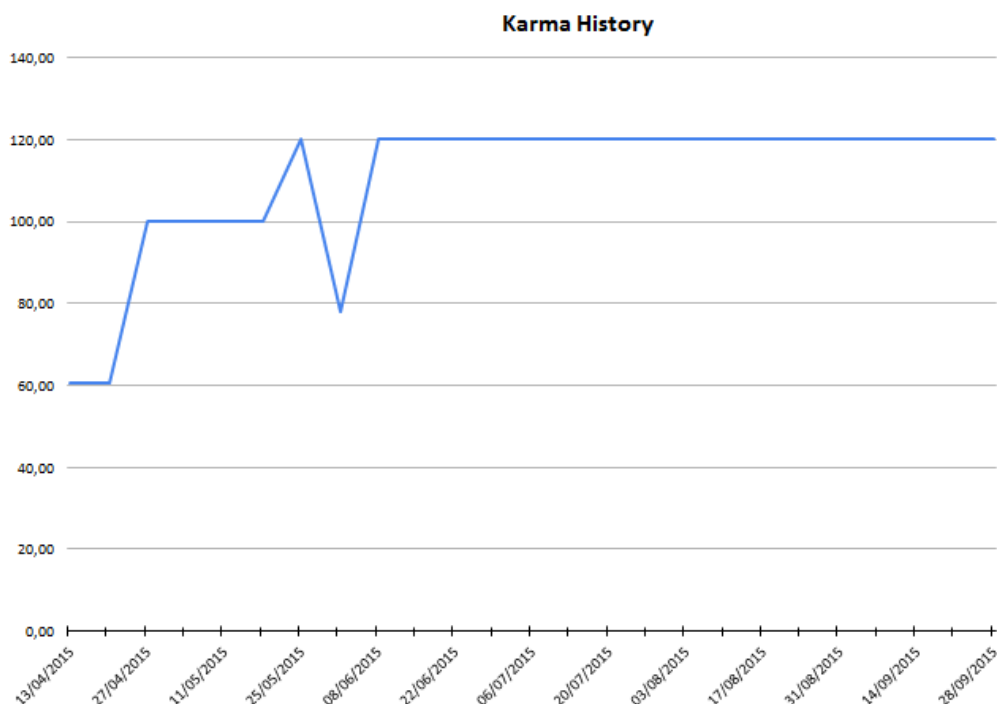


Figure 81: Karma history – Poznan Node

The Poznan node upgraded its Openstack installation to IceHouse in the end of 2014. This release of OpenStack was recommended by the consortium.

The only low value scored by the Poznan node in the Karma Points Calculation was on the very first measurement in March 2015, stemming from monitoring issues and 3 errors in the Sanity Check tests. In the next week the Poznan node solved almost all monitoring issues and the score increased to 60 points.

By the end of April all errors in Sanity Check were removed and the Karma score reached 100 points. In May, Historical Data support was added, resulting in the maximum Karma score of 120 points which has remained at this level up to the day of writing this paragraph (beginning of September 2015).

The Poznan node team is continuously working to provide the broadest functionality and best user experience possible.

7.3.10 Prague

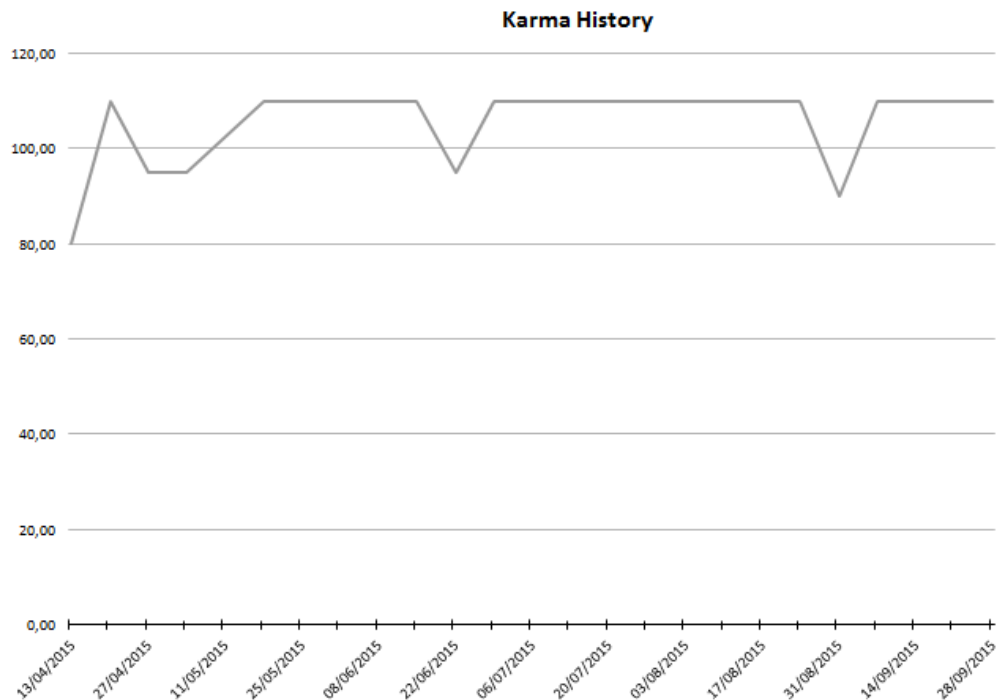


Figure 82: Karma history – Prague Node

The Prague node joined the XIFI project via the open call process on April 1st, 2014. The OpenStack deployment was performed on a Red Hat Enterprise Linux clone (the Scientific Linux) using OpenStack Grizzly release from the RDO distribution and Puppet scripts from the upstream OpenStack StackForge repositories.

The Prague node was the first node from the open call process to join FIWARE Lab, and its operation remained stable throughout the extension period as well. The hardware capacity (the number of CPU cores and available RAM) was extended in early 2015, and the servers gained a local SSD storage. A similar upgrade was performed to the network where the internal connections were changed from dual-gigabit to dual-10 Gb Ethernet with redundant connections. An L3 switch was deployed in early 2015 to facilitate this upgrade which also enabled the Prague node to offer native IPv6 connectivity to all user VMs by default. This is an important contribution to the global migration from the legacy IPv4 protocol. The IPv4 addressing remains an option to help with this process.

The Prague node has been running stable with no known disruptions to the user's VMs. The gaps in the "karma score" were caused by factors outside of the Prague node; our extended capacity was interpreted as "underutilized" by the monitoring system due to a lower-than-expected number of running virtual machines in mid-2015.

7.3.11 SophiaAntipolis

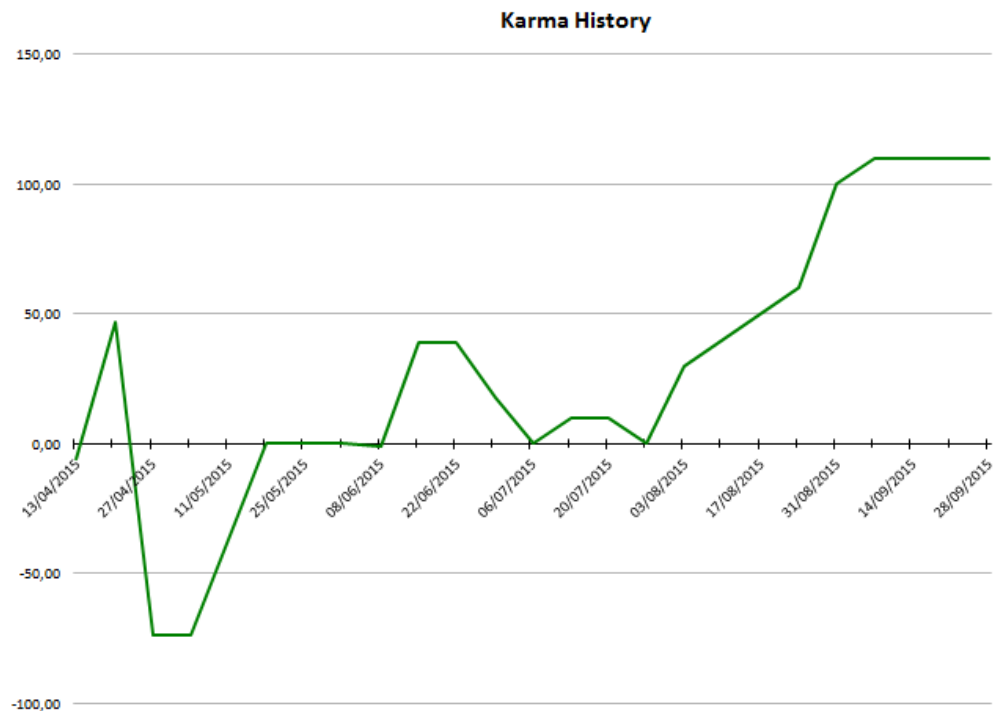


Figure 83: Karma history – Sophia Antipolis Node

Sophia-Antipolis node has been upgraded to Icehouse version to provide a stable platform. We make adjustments every day to improve the stability, reactivity of components and repair dynamics in the case of an error. We are very invested in user support also.

Low karma values are due from the migration to icehouse. Also, we had many errors that occurred during deployment. We also worked on the stability of the node during the period from May to June. We noticed some sanity check tests were not done. The issue was linked by a wrong configuration on sanity check scripts. We made the request for correction. From this point, the value of karma has been significantly improved. We strive to maintain it in such status.

7.3.12 Spain

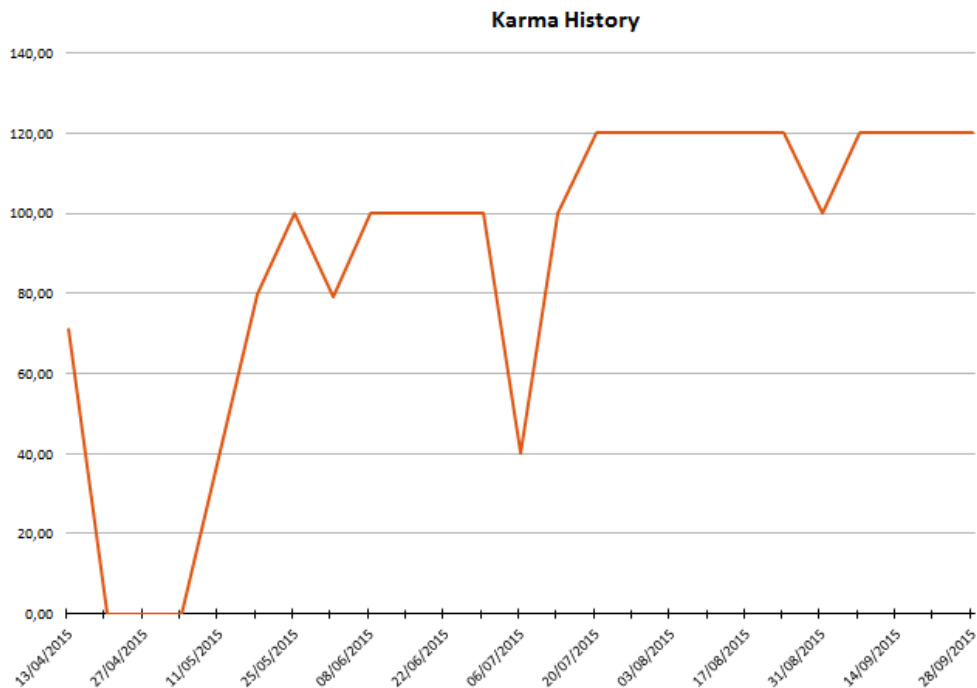


Figure 84: Karma history – Spain Node

The Spain node was the initial node that had been deployed with OpenStack Essex release. The high demand of resources and the continuous increase of FIWARE Lab users goes to a situation in which the oversubscription of resources produce errors in the users that have to be manage every day with a lot of effort in order to keep working the node. The situation was resolved with the progressive resources increase thanks to the installation of new Data Centers in Málaga, Sevilla and Las Palmas de Gran Canaria. Besides it, the introduction of new user policies gave freed a number of abused resources increasing the amount of available resources. At the same time, the freed resources gave us the possibility to migrate to Juno OpenStack release. Spain was the first FIWARE Lab node to have installed the new version of OpenStack, which gave the consortium the necessary knowledge to use this new version. This migration included the progressive migration of the users from the previous version to the new one in order to free the previous infrastructure used with the Essex version.

The fact that Spain was the first in the installation of Juno, allowed us to identify several problems with this version that was reported to the consortium in order to resolve future problems on it. Example of it was the identification of problem with the parallelisation of the neutron services that do not work with a significant number of users how it is the case in the Spain node. This gave us lots of problems at the beginning of May, this issue was identified by the community like a error in the neutron service and we had to install a centralised version of neutron agents. The OpenStack community reports that in the following versions of OpenStack the issue should be resolved.

Since the end of May the Spain node is constantly operational and reliable, therefore got the score of 100 karma points from there on. The reason why we do not obtain 120 points was due to the especial configuration of our network interfaces that produces errors in the monitoring tool that was not resolved until end of July. But apart from it, the other problems are that the monitoring components are using oldest version of GE that should be updated but it was reported that an incompatibility problems was detected and it is not updated at the moment.

7.3.13 Stockholm

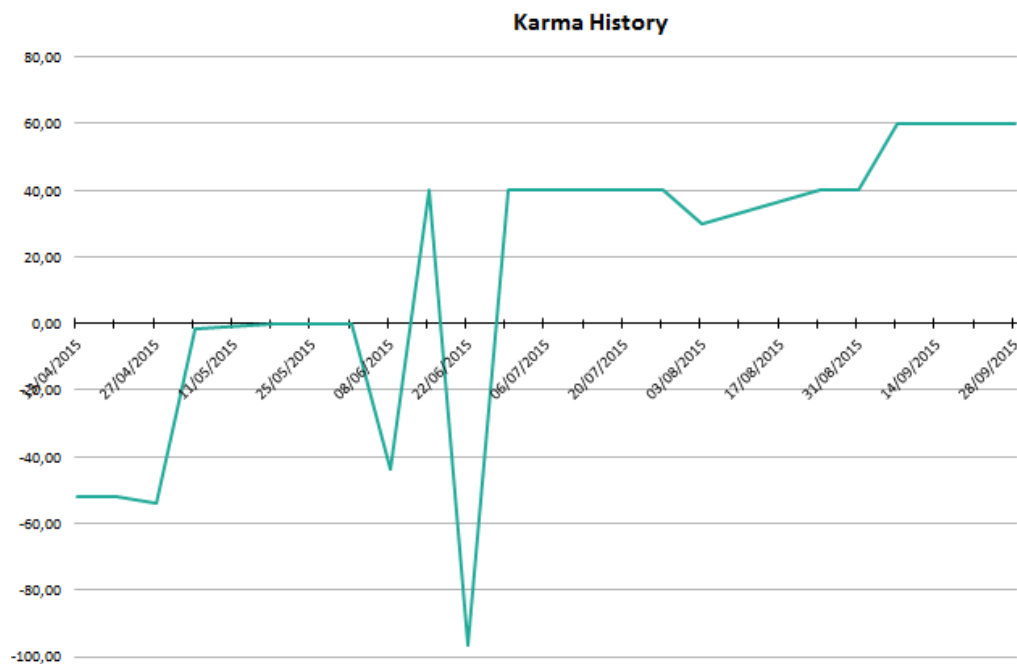


Figure 85: Karma history – Stockholm Node

The Stockholm node needed more hardware to be able to be upgraded. Unfortunately this delayed the upgrade to Icehouse, when we needed to get more financing to be able to set up two parallel nodes (Stockholm1 and Stockholm2). The Grizzly node (Stockholm1) was still up in the beginning of April but was in a really bad shape. A full investigation to track the root cause and bring the the Grizzly system up again was considered too much work to do, instead our intention was to go for an upgrade to icehouse. The probable root cause that Stockholm1 succumbed was a heavy overload usage by some tenants that violated the terms and conditions within FIWARE Lab.

After closing down the violating tenants the Stockholm1 never fully recovered, not even after a total restart of Openstack Grizzly. From end of April the work focused on bringing up our new node Stockholm2 based on Icehouse as soon as possible and leave the Grizzly installations. When the new hardware arrived in May, Icehouse was installed with the help of ITbox (Icehouse version). Extensive stability tests were followed before final federation and activation of the new Icehouse based Stockholm2 in the end of June. A new install of monitoring was needed and was in place to show green status on the Infographic in beginning of September. Because of the hardware delay and struggle with the installation (both fine tuning of Icehouse after the ITBox installation together with monitoring) the Stockholm2 node suffered on karma points. Anyway, our Stockholm2 node has been stable since the end of June and have showed good result in FIWARE Lab "SANITY CHECK STATUS".

7.3.14 Trento

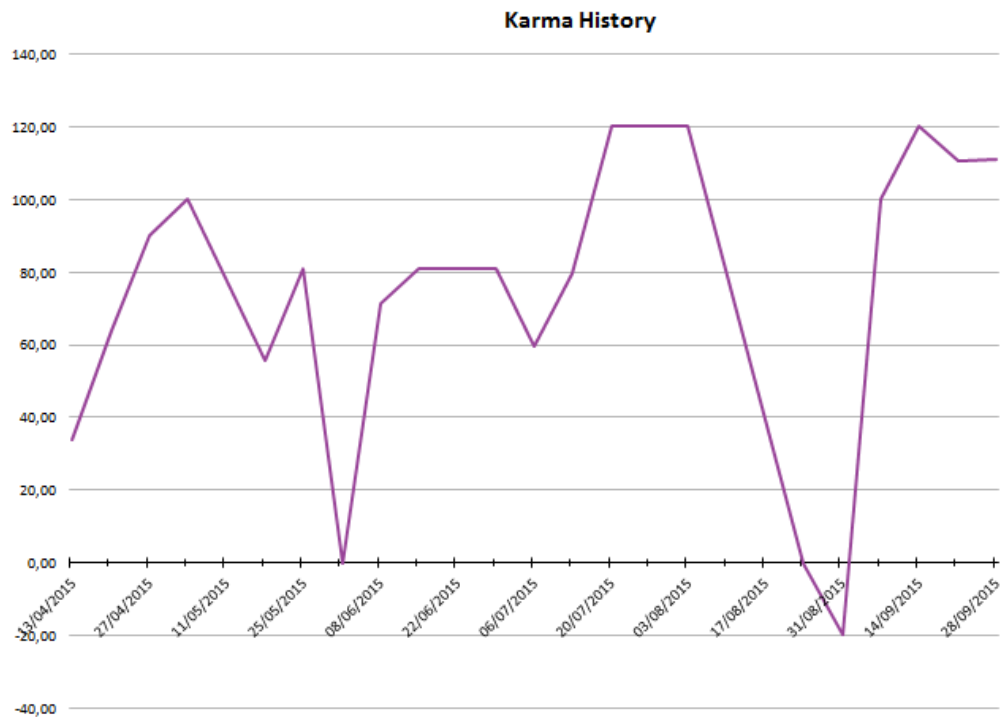


Figure 86: Karma history – Trento Node

From March, 9th Trento Node is hosting Openstack Icehouse version, deployed using Fuel release 5.1.1.

The new Icehouse production node has been deployed in HA-mode, using CephRBD for images and volumes with a replication factor equal to 3.

The installation of the additional nodes continued throughout end of March 2015 and beginning April 2015, situation that did not allow us to get maximum points available on resources vs. actual instances.

Changes on the configurations (historical data installation and configuration of infographics) continued during the end of March 2015 and beginning of April 2015.

Starting from 20th of April 2015 all the configurations have been ultimated and the node became stable also in the Karma points.

On the Karma calculation of 01/06/2015, the status "burdening FIWARE Lab" was due to an attempt of a Neutron QoS plugin for the Demo in Bruxelles.

On the Karma calculation of 24/08/2015, the status "burdening FIWARE Lab" was due to a maintenance week.

On the Karma calculation of 31/08/2015, the status "burdening FIWARE Lab" was due to the missing of the synchronization of the Fiware Lab images and all the 11 sanity check tests failed on retrieving the CentOS init image.

7.3.15 Volos

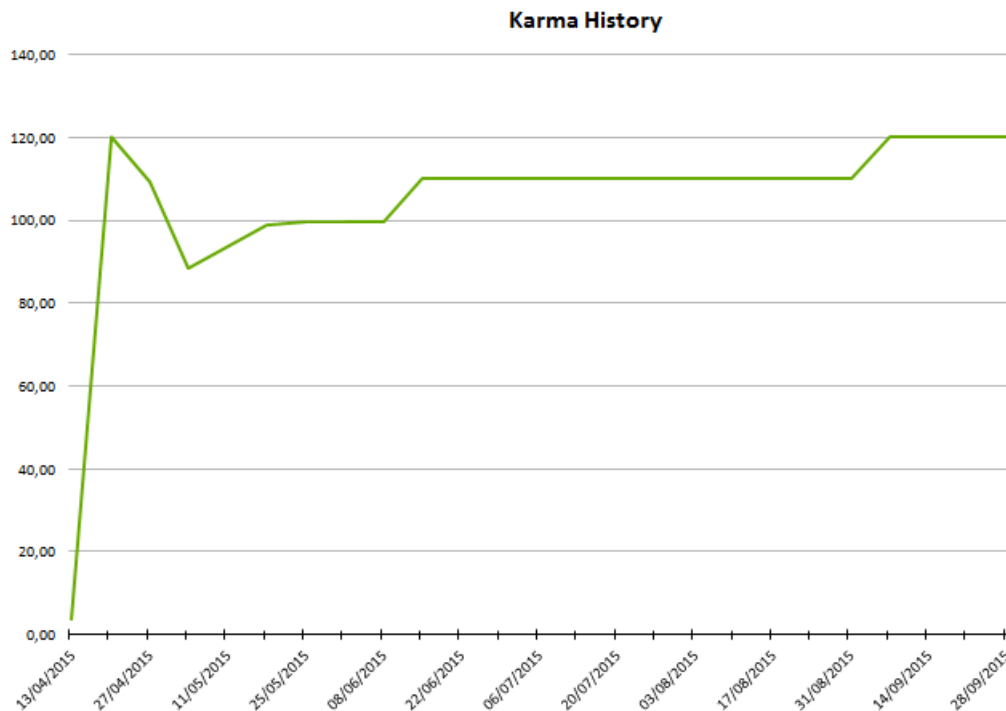


Figure 87: Karma history – Volos Node

Volos node reached its stability after the Icehouse version of Openstack was deployed for the second time. It was then that the solution of Ceph was abandoned and the usage of the more stable NFS protocol was chosen for Openstack's storage components.

After that point the measurements diversities were minor and were caused primarily due to lack of public ips, which led to errors during the sanity check testing. After cleaning the idle virtual machines and freeing the corresponding ips, accompanied with the project's adaptation of stricter registration mechanisms, the problem disappeared.

Since early May an increasing trend is depicted on the graph due to the increase of the node's resources usage via the hosting of more virtual machines from FIWARE Lab users.

Volo's node team is continuously working towards the goal of maintaining the above levels of QoS and stability.

7.3.16 Waterford

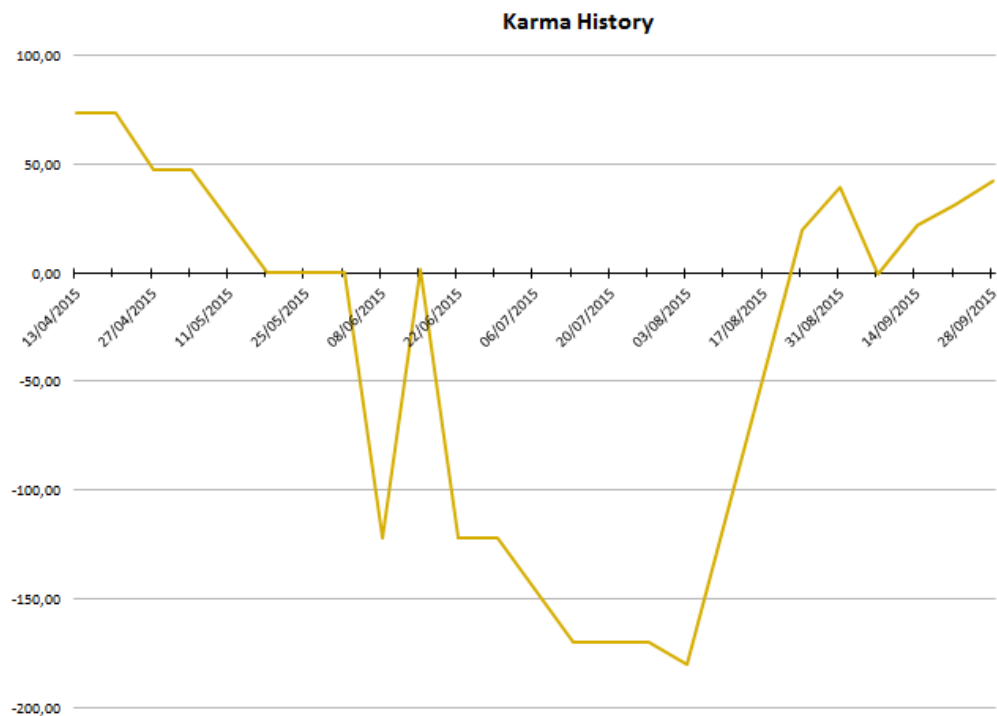


Figure 88: Karma history – Waterford Node

The Waterford node was operating well using Grizzly up to the upgrade of the IdM in May 2015. As most of the setup had been completed manually with a number of patches added over time, when there was an authentication issue with Grizzly, it was difficult to recover. In addition, the team changed significantly due to the extension of the project.

It was decided to reinstall the cluster adding hardware from the decommissioned testing environment on OpenStack Juno. This was completed by the start of August. The new install had an object storage service but it wouldn't authenticate with the centralised Keystone so the decision was made after 2 week of debugging to remove the service. This should bring the overall service stability up to 21/21. Karma scores remain low due to missing configuration of the context broker based tools (infographic).

As the central component owners have returned from annual leave and recommended fixes, this should be resolved in short order. As this was not service affecting, other maintenance issue take priority such as a brief DNS issue on 2015-09-07.

7.3.17 Zurich

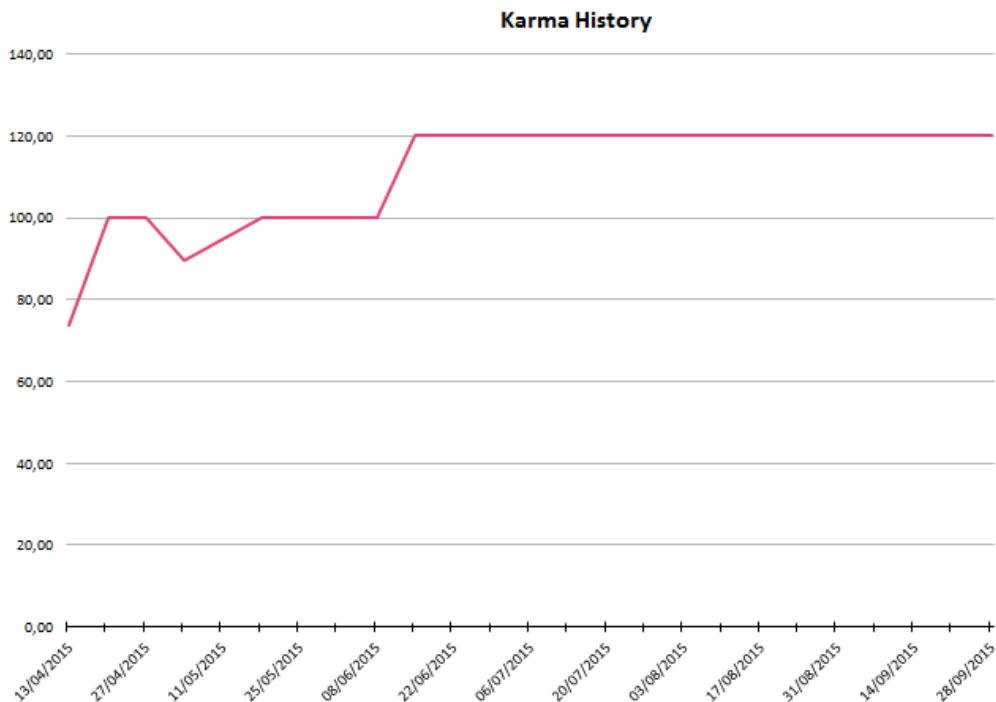


Figure 89: Karma history – Zurich Node

The Zurich node has exhibited stability since the deployment of the Icehouse release in early 2015; this manifested in consistently high karma points throughout the latter stages of the project.

As with all operations, effort was required to ensure this level of performance. As well as using the standard monitoring and notification tools (nagios) to quickly identify problems, we wrote scripts to identify malicious use of the system which may occur due to, for example, Virtual Machines being compromised. Specifically, we wrote scripts that identified large amounts of continuous traffic to or from VMs and provided notifications if found; we then investigated whether this was normal or potentially malicious behaviour. We also performed an upgrade to secure the system against Venom which involved quite some manual work associated with moving VMs to evacuate single hosts such that the upgrade could be performed.

The approach taken by the Zurich node was somewhat conservative regarding providing increased capabilities; the focus was on ensuring reliable operations. For this reason, the node was not upgraded to Juno/Kilo, nor was a storage service deployed. These are planned for the future.

8 CONCLUSIONS

This document provides an update on XIFI nodes operation and maintenance procedures. It has reported about the nodes operation and support that was provided until month 24, but more important are the description of operation and maintenance agreements and procedures. Moreover, this document provides now also a description in a good level of detail on how support is being realised for FI-Developers but also for federation partners, i.e. node owners.

While this is a significant progress since D5.1 was released at M6, the described protocols and procedures are stable now. They have evolved since the beginning of the project fed by experience gained from adding further (heterogeneous) nodes and from operating and maintaining a growing federation getting more complex.

Thus, the definitions and specifications contained in this document will be transferred to FI-Core project and others future projects to serve as the standard and binding reference.

REFERENCES

- [1] XIFI Deliverable D1.1.1: XIFI Core Concepts, Requirements and Architecture Draft
- [2] XIFI Deliverable D1.2: Analysis of UC, Infrastructures and Enablers v1
- [3] XIFI Deliverable D1.3: Federated Platform Architecture v1
- [4] XIFI Deliverable D1.3: Federated Platform Architecture v1, section 4.4.2.2 “Future Opportunities: SLAs and OLAs”
- [5] XIFI Deliverable D5.1: Procedures and Protocols for XIFI federation
- [6] XIFI Deliverable D5.2 - XIFI Core Backbone
- [7] XIFI Deliverable D5.3: XIFI node operation, maintenance, assistance and procedures updates
- [8] XIFI Deliverable D5.6: XIFI federation extension and support
- [9] XIFI Deliverable D6.2: XIFI Showcases Demonstrators v1
- [10] XIFI Deliverable D9.2b: XIFI Office – Description and Establishment
- [11] XIFI Wiki: <http://wiki.fi-xifi.eu/>
- [12] "Pegasus WMS: Enabling Large Scale Workflows on National Cyberinfrastructure" Karan Vahi, EwaDeelman, Gideon Juve, Mats Rynge, Rajiv Mayani, Rafael Ferreira da Silva. XSEDE 2014, Atlanta, Georgia. July 2014.
- [13] Wikipedia contributors, "MyExperiment," Wikipedia, The Free Encyclopedia, <http://en.wikipedia.org/w/index.php?title=MyExperiment&oldid=595344437> (accessed July 23, 2014).
- [14] Varas, C.; Hirsch, T. Self Protection through Collaboration Using D-CAF: A Distributed Context-Aware Firewall, appears in: Emerging Security Information, Systems and Technologies, 2009. SECURWARE '09. Third International Conference on, Issue Date: 18-23 June 2009
- [15] Sally Floyd maintained the RED resource at <http://icir.org/floyd/red.html>
- [16] Checklist SLA OLA according to ITIL 2011 Service Design, http://wiki.en.it-processmaps.com/index.php/Checklist_SLA_OLA
- [17] XIFI stakeholder definitions, <https://www.fi-xifi.eu/about-xifi/stakeholders.html>
- [18] The 1st International Workshop on Trust in Cloud Computing (IWTCC2014), London, UK, December 8-11, 2014, on-line at <http://computing.derby.ac.uk/IWTCC2014/>
- [19] Stackoverflow, question and answer site, <http://stackoverflow.com>
- [20] Ask OpenStack: Q&A site of the OpenStack community: <https://ask.openstack.org/en/questions/>
- [21] Glance added property NID description <http://docs.openstack.org/image-guide/content/image-metadata.html>
- [22] Guestfish: http://docs.openstack.org/image-guide/content/ch_modifying_images.html
- [23] GE / GEi owners list (maintained by FI-WARE project): <https://docs.google.com/spreadsheet/ccc?key=0AqLSWp0KXaaDdEpLT0RmXzhzbXEtSTVvSE5wX3oyWXc#gid=0>
- [24] FitSM-0:2013: Overview and vocabulary, on-line at <http://www.fedsm.eu/fitsm/0>
- [25] Pipes and Filters Pattern, Microsoft Corporation, on-line at <http://msdn.microsoft.com/en-us/library/dn568100.aspx>
- [26] Wikipedia contributors, "Operational definition," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Operational_definition&oldid=644847075 (accessed

February 15, 2015).

- [27] Laurent Ciavaglia, Samir Ghamri-Doudane, Mikhail Smirnov, Panagiotis Demestichas, Vera-Alexandra Stavroulaki, Aimilia Bantouna and Berna Sayrac *Unifying Management of Future Networks With Trust*, Bell Labs Technical Journal, vol. 17(3), pp 193–212, Article first published online: 27 DEC 2012 | DOI: 10.1002/bltj.21568
- [28] XIFI Federation Monitoring APIs : <http://docs.federationmonitoring.apiary.io/>
- [29] XIFI Federation Monitoring : http://wiki.fi-xifi.eu/Public:Federation_Monitoring
- [30] Public list of software components: http://wiki.fi-xifi.eu/Public:Software_Components
- [31] Private list of software components: <http://wiki.fi-xifi.eu/Xifi:Components>
- [32] ABNO Controller: http://wiki.fi-xifi.eu/Creating_Xifi:Wp3:Components:ABNOController
- [33] Access Control GE: http://wiki.fi-xifi.eu/Xifi:Wp2:Access_Control_GE
- [34] Public Access Control GE: http://wiki.fi-xifi.eu/Public:Access_Control_GE
- [35] BigData Analysis - Comos: <http://catalogue.fi-ware.org/enablers/bigdata-analysis-cosmos/documentation>
- [36] Cloud Portal: http://wiki.fi-xifi.eu/Xifi:Wp4:Cloud_Portal
- [37] Public Cloud Portal: http://wiki.fi-xifi.eu/Public:Cloud_Portal
- [38] Orion Context Broker: <http://catalogue.fi-ware.org/enablers/configuration-manager-orion-context-broker/documentation>
- [39] DEM: <http://wiki.fi-xifi.eu/Xifi:Wp3:Components:DEM>
- [40] Public DEM: <http://wiki.fi-xifi.eu/Public:DEM>
- [41] DCA: <http://wiki.fi-xifi.eu/Xifi:Wp3:Components:DCA>
- [42] Public DCA: http://wiki.fi-xifi.eu/Public:Deployment_and_Configuration_Adapter
- [43] DCRM GE: http://wiki.fi-xifi.eu/Xifi:Wp3:DCRM_GE
- [44] Public DCRM GE: <http://catalogue.fi-ware.org/enablers/iaas-resource-management-ge-fiware-implementation/documentation>
- [45] DNS as a Service: http://wiki.fi-xifi.eu/Xifi:Wp3:_DNS_as_a_Service
- [46] Federation Manager: http://wiki.fi-xifi.eu/Xifi:Wp2:Federation_Manager
- [47] Public Federation Manager: http://wiki.fi-xifi.eu/Public:Federation_Manager
- [48] Federation Monitoring: http://wiki.fi-xifi.eu/Xifi:Wp2:Federation_Monitoring
- [49] Public Federation Monitoring: http://wiki.fi-xifi.eu/Public:Federation_Monitoring
- [50] Identity Management: http://wiki.fi-xifi.eu/Xifi:Wp2:Identity_Management_GE
- [51] Public identity Management: http://wiki.fi-xifi.eu/Public:Federated_Identity_Management
- [52] Infographics and Status Pages: http://wiki.fi-xifi.eu/Xifi:Wp4:Infographics_and_Status_Pages
- [53] Public Infographics and Status Pages: http://wiki.fi-xifi.eu/Public:Infographics_and_Status_Pages
- [54] Infrastructure Toolbox: <http://wiki.fi-xifi.eu/Xifi:Wp3:Components:InfrastructureToolbox>
- [55] Public Infrastructure Toolbox: <http://wiki.fi-xifi.eu/Public:InfrastructureToolbox>
- [56] Interoperability Tool: http://wiki.fi-xifi.eu/Xifi:Wp4:Interoperability_Tool
- [57] Public Interoperability Tool: http://wiki.fi-xifi.eu/Public:Interoperability_Tool

- [58] Monitoring Dashboard: http://wiki.fi-xifi.eu/Xifi:Wp4:Monitoring_Dashboard
- [59] Public Monitoring Dashboard: http://wiki.fi-xifi.eu/Public:Monitoring_Dashboard
- [60] NAM: <http://wiki.fi-xifi.eu/Xifi:Wp3:Components:NAM>
- [61] Public NAM: <http://wiki.fi-xifi.eu/Public:NAM>
- [62] Network Provision Manager: <http://wiki.fi-xifi.eu/Xifi:Wp3:Components:NetworkProvisioningManager>
- [63] NGSI Adapter: <http://wiki.fi-xifi.eu/Xifi:Wp3:Components:NGSIAdapter>
- [64] Public NGSI Adapter: <http://wiki.fi-xifi.eu/Public:NGSIAdapter>
- [65] NPM Adapter: <http://wiki.fi-xifi.eu/Xifi:Wp3:Components:NPM>
- [66] Public NPM Adapter: <http://wiki.fi-xifi.eu/Public:NPM>
- [67] Open NaaS: <http://wiki.fi-xifi.eu/Xifi:Wp3:Components:OpenNaaS>
- [68] Open Stack Collector: <http://wiki.fi-xifi.eu/Xifi:Wp3:Components:OpenstackDataCollector>
- [69] Public Open Stack Collector: <http://wiki.fi-xifi.eu/Public:OpenstackDataCollector>
- [70] Path Computation Element: <http://wiki.fi-xifi.eu/Xifi:Wp3:Components:PathComputationElement>
- [71] Platform as a Service Manager: http://wiki.fi-xifi.eu/Xifi:Wp2:PaaS_GE
- [72] Platform as a Service Manager- Pegasus: <http://catalogue.fi-ware.org/enablers/paas-manager-pegasus/documentation>
- [73] Quick Online Test: http://wiki.fi-xifi.eu/Xifi:Wp4:Quick_Online_Test
- [74] Resource Catalogue and Recommendation tools: http://wiki.fi-xifi.eu/Xifi:Wp4:Resource_Catalogue%26Recommender
- [75] Public Resource Catalogue and Recommendation tools http://wiki.fi-xifi.eu/Public:Resource_Catalogue%26Recommender
- [76] Software Deployment and Configuration GE: http://wiki.fi-xifi.eu/Xifi:Wp2:SDC_GE
- [77] Public Software Deployment and Configuration GE: <http://catalogue.fi-ware.org/enablers/software-deployment-configuration-sagitta/documentation>
- [78] Security Dashboard: http://wiki.fi-xifi.eu/Xifi:Wp4:Security_Dashboard
- [79] Public Security Dashboard: http://wiki.fi-xifi.eu/Public:Security_Dashboard
- [80] Security Monitoring: http://wiki.fi-xifi.eu/Xifi:Wp4:Security_Monitoring_GE
- [81] Public Security Monitoring: http://wiki.fi-xifi.eu/Public:Security_Monitoring_GE
- [82] Security Proxy: http://wiki.fi-xifi.eu/Public:Security_Proxy
- [83] SLA Manager: http://wiki.fi-xifi.eu/Xifi:Wp4:SLA_Manager
- [84] Public SLA Manager: http://wiki.fi-xifi.eu/Public:SLA_Manager
- [85] Adopting local configurations into a federation setup [http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task1.1: Adopting local configurations into a federation setup. .E2.86.92 WIT](http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task1.1:Adopting_local_configurations_into_a_federation_setup..E2.86.92_WIT)
- [86] Not compliant node [http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task1.2: What to do with a node that is not compliant with the XIFI federation at the moment. .E2.86.92 RED.ES](http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task1.2:What_to_do_with_a_node_that_is_not_compliant_with_the_XIFI_federation_at_the_moment..E2.86.92_RED.ES)
- [87] Resource disengagement <http://wiki.fi->

- xifi.eu/Xifi:Wp5:t5.4#Task1.3: Resource disengagement .28e.g. lifetime of VMs.2C Floating IP.29 .E2.86.92 TN .2F TI
- [88] Maintenance of tenant quotas <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task1.6>: Maintenance of tenant quotas .E2.86.92 TN
- [89] Inter node network connectivity maintenance <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task1.5>: Inter node network connectivity maintenance .E2.86.92 DANTTE
- [90] Failover scenario definition <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task1.4>: Failover scenario definition .28e.g. compute node crashes.2C node not reachable29 .E2.86.92 TN .2F TI
- [91] Recovery from security events <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task1.7>: Recovery from security-related events .28complements T52-2.8 regarding infrastructure measures.29
- [92] Software distribution <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task2.2>: SW distribution .28how softwares are delivered from other WPs.29 .E2.86.92 TN .2F ILB
- [93] Software update <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task2.3>: SW updates .E2.86.92 TN), or for removing deployed software (see <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task2.4>: SW removal .E2.86.92 RED.ES
- [94] Software compliance <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task2.5>: SW compliance .E2.86.92 WIT
- [95] Software backup <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task2.6>: SW backup .28config files.3B DB etc..29 .E2.86.92 TN
- [96] Openstack release upgrade <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task2.10>: OpenStack release upgrade .E2.86.92 TID
- [97] Maintenance of floating IP <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task2.9>: Maintenance of floating IPs .E2.86.92 Fraunhofer
- [98] Alignment of software procedures <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task2.7>: Alignment of SW maintenance procedures between FIWARE and XIFI .28applies to former tasks.29 .E2.86.92 TID
- [99] Report of problems <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task4.1>: Report of problems .28Communication between Helpdesk and infrastructure owner.29 .E2.86.92 ILB
- [100] Maintenance notification <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task4.2>: How and who to notify about scheduled and unscheduled maintenance .28.29 .E2.86.92 TN .2F TI
- [101] Recovery from security <http://wiki.fi-xifi.eu/Xifi:Wp5:t5.4#Task2.8>: Recovery from security-related events .28complements T52-1.7 regarding software maintenance measures.29

Appendix A Operational Level Agreements

A.1 Operational Level Agreements

When joining the federation a node implicitly agrees on – or already has implemented as a prerequisite for joining – a number of rules, for example, to install conformant cloud management, monitoring and access control services. Hence, the joining node enters into a set of operational agreements between node and federation already in an early state of federation. When agreeing to implement common operations and maintenance procedures of the federation, for example by joining the help-desk, a more detailed set of operational agreements settles into place and performance indicators begin to apply. As further outlined by subsequent sections, operational level agreements target a) the implementation of procedures for mutual collaboration of nodes in the federation and b) the capacity-building for the federation being able to satisfy service level agreements towards the federation user.

A viable federation of IT resources and services⁸ maintains a number of SLA's that serve as means for quantitative evaluation of service invocations by the users (developers in case of XIFI). However to maintain these SLAs a federation must deploy Operational Level Agreements (OLA). The main purpose to deploy and to maintain OLAs is to assure the SLA's targets.

If a SLA is an agreement between service customers (users) and service providers, then an OLA is an agreement between different groups (roles) of customers on how they should support various aspects of SLA delivery.

The Best Current Practices of OLA are known from e.g. ITIL (see e.g. the OLA checklist – a template to define an OLA at http://wiki.en.it-processmaps.com/index.php/Checklist_SLA_OLA), FitSM (see e.g. the definition of the seven key roles in the organisation of IT service management at http://www.fedsm.eu/sites/default/files/FitSM-3-2013_1.2.pdf), etc. (another example is EuroGrid).

In XIFI the importance of OLA understanding, design and deployment is perhaps higher than in a usual case because of the following factors: 1) heterogeneity of resources, services, and providers, 2) distributed nature of the federation, 3) multiple and mostly recurrent dependencies (on GE, SE, but also on nodes and communication features).

Both SLA and OLA should be seen as evolving frameworks following certain maturity (capability) levels with First Line Support (FLS) being, probably the most common starting level.

The most common second level of SLA and OLA capabilities should be such an extension of these frameworks, which allows operational definition of workflow sequences of actions. These must be useful for customers, agreed by providers, registered in a common federation repository, and tested at sufficient level to recommend these workflows to all members of a federation.

A.2 OLA Level 3 Rationale

The first question that should be addressed is why OLA should be extended with workflows (WF). There are several benefits that directly follow from the WF orientation:

1. All activities described as WF's appear as processes, in which the usage of a set of resources and/or services is linked to a WF – an entity that can be uniquely identified by a WF ID and that makes such particular set also unique despite the fact that the same

⁸ In OLA all resources contribute to externally visible services, hence in workflow language we could talk only about services.

resources and services can be utilised by multiple WF's; the process orientation directly helps to create services;

2. Workflow orientation facilitates sustainability of a federation, because it automates repetitive tasks, attracts new users and keeps the old users by allowing them to modify existing workflows and to inherit successful designs;
3. Workflows facilitate trust in both directions: (a) users trust the infrastructure more and more as long as the results are achieved with less effort, (b) infrastructure providers tend to trust those users that register their workflows, re-use them and share with other eligible users;
4. Managed workflows are generally helping to improve the quality of overall system management – eliminate waste and noise, spare resources and energy, achieve optimization, etc.

The above list of workflow benefits can be easily extended following a large body of evidence from industry, but not only. Special attention should be paid to scientific workflows – the area, in which the XIFI project “facilitates the uptake, deployment and federation of several instances of such a common platform to pave the way for a unified European marketplace that is crucial for enabling commercial exploitation of FI resources. This is achieved via FIWARE Ops, a collection of tools that ease the deployment, set-up and operation of FIWARE instances on infrastructures.”

Concentrating on scientific workflows we should examine some best current practices (BCP) known from various academic fields. The first BCP to mention is the Pegasus [12] software package developed and maintained by the ISI (Information Science Institute at the University of Southern California) and used by a large number of mainly American Universities and research centres such as NASA. From the usage experience the developers were able to provide the following extended definition of a scientific workflow: “A scientific workflow describes the dependencies between the tasks and in most cases the workflow is described as a directed acyclic graph (DAG), where the nodes are tasks and the edges denote the task dependencies. A defining property for a scientific workflow is that it manages data flow. The tasks in a scientific workflow can be everything from short serial tasks to very large parallel tasks (MPI for example) surrounded by a large number of small, serial tasks used for pre- and post-processing.” Pegasus is a multi-platform system since it provides a mapping from an abstract workflow to a final executable one as demonstrated in Figure 90, taken from [12].

Pegasus has a number of features that contribute to its usability and effectiveness as described on its on-line resource:

- Portability/ Reuse – User created workflows can easily be run in different environments without alteration. Pegasus currently runs workflows on top of Condor, Grid infrastructures such as Open Science Grid and TeraGrid, Amazon EC2, Nimbus, and many campus clusters. The same workflow can run on a single system or across a heterogeneous set of resources.
- Performance – The Pegasus mapper can reorder, group, and prioritize tasks in order to increase the overall workflow performance.
- Scalability – Pegasus can easily scale both the size of the workflow, and the resources that the workflow is distributed over. Pegasus runs workflows ranging from just a few computational tasks up to 1 million. The number of resources involved in executing a workflow can scale as needed without any impediments to performance.
- Provenance – By default, all jobs in Pegasus are launched via the kickstart process that captures runtime provenance of the job and helps in debugging. The provenance data is collected in a database, and the data can be summaries with tools such as pegasus-statistics, pegasus-plots, or directly with SQL queries.
- Data Management – Pegasus handles replica selection, data transfers and output registrations in data catalogues. These tasks are added to a workflow as auxiliary jobs by the Pegasus planner.

- Reliability – Jobs and data transfers are automatically retried in case of failures. Debugging tools such as pegasus-analyser helps the user to debug the workflow in case of non-recoverable failures.
- Error Recovery – When errors occur, Pegasus tries to recover when possible by retrying tasks, by retrying the entire workflow, by providing workflow-level check pointing, by re-mapping portions of the workflow, by trying alternative data sources for staging data, and, when all else fails, by providing a rescue workflow containing a description of only the work that remains to be done. It cleans up storage as the workflow is executed so that data-intensive workflows have enough space to execute on storage-constrained resource. Pegasus keeps track of what has been done (provenance) including the locations of data used and produced, and which software was used with which parameters.

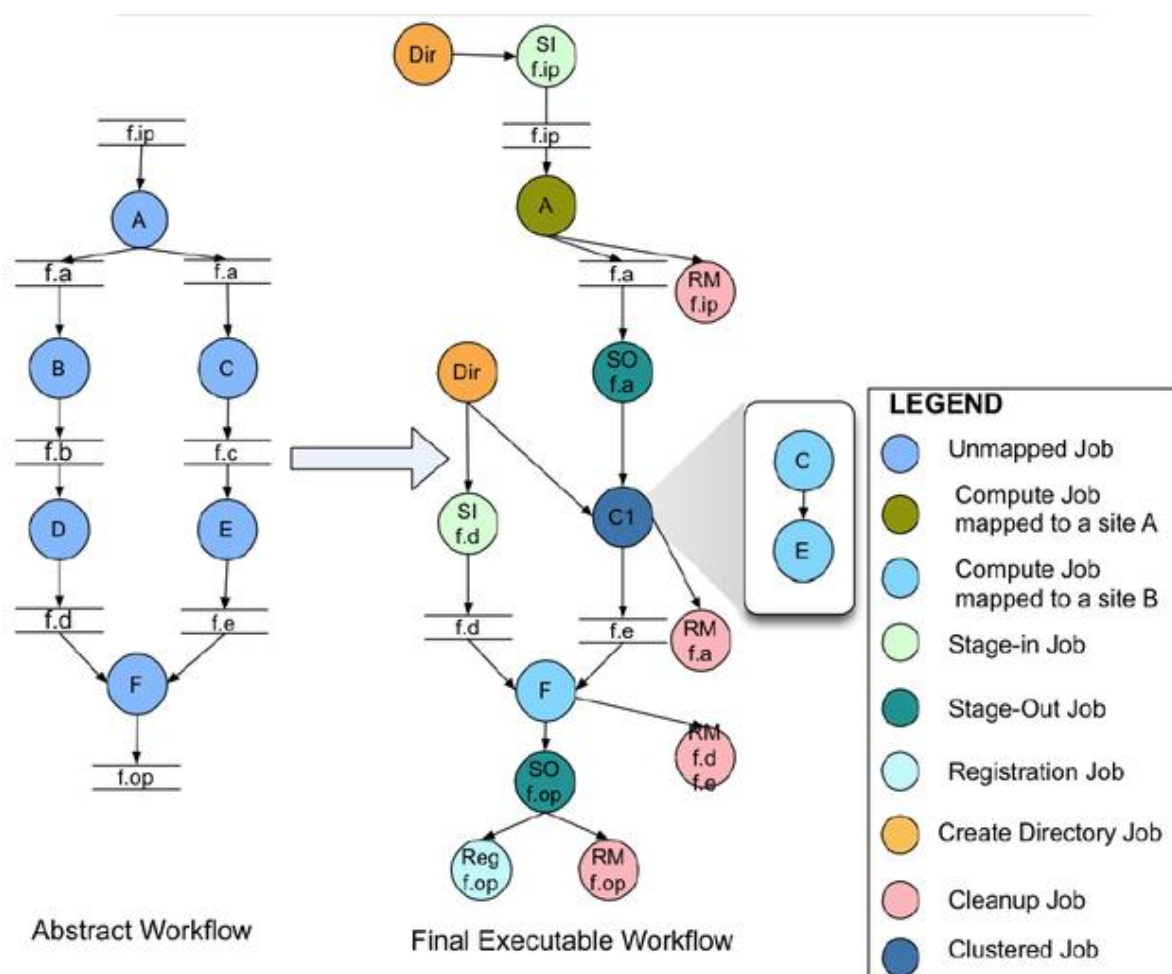


Figure 90: Pegasus WF mapping

The second BCP to mention is a workflow platform at myexperiment.org [13], a joint effort of the universities of Southampton, Manchester and Oxford in the UK, led by David De Roure and Carole Goble. The myExperiment is currently supported by three European Commission 7th Framework Programme (FP7) projects: BioVeL (Grant no. 283359), SCAPE (Grant no. 270137), and the Wf4Ever Project (Grant no. 270192) as well as the e-Research South and myGrid EPSRC Platform grants.

This platform deserves a particular attention of XIFI partners not only because it is usage based and is growing via the user generated workflows but also as reported [13] by one of the founders it has fairly early implemented the two critical features for such platforms. These features are, scalability assurance, and the right handling of IPR, since researchers who share their work are very much sensitive to the three components of IPR handling, namely Credit, Attribution and Licensing.

In summary, workflows appear to be a natural extension of First Line Support systems that can be seen as an initial OLA for a federation of infrastructure providers, because it is equally helpful in managing activities at both sides of a federation – at user side and at the side of infrastructure providers. However, since within the XIFI project the demand for OLA comes mainly from the need to maintain responsibilities of multiple stakeholders of the project we need to look at workflows exactly from that viewpoint (this is attempted in the security benefits section), but before we need to understand technical features of workflow execution that are essential for responsibility maintenance – this is attempted in the next section.

A.3 Workflows for Cross-layer Optimisation

The need for a cross-layer optimisation was well understood long ago: an example is Random Early Detection (RED)[15]- an algorithm which operates deeply in a datagram network yet it is capable of optimising the performance at transport level. The need for such algorithms is hard to underestimate in any environment that uses multiplexing; hence a federation of cloud computing infrastructures appears as a natural area for their deployment.

The problem solved by a RED stems from the fact that each IP module, when congested has a license to kill any IP datagram. Most mechanisms try to avoid congestion, and when it happens not really care about which datagram to drop. Contrary to that RED on congestion breaks unwanted synchronization between TCP connections sharing the buffer. Unfortunately, no RED flavours were developed that when selecting a datagram to drop take into account application-level utility of this datagram. Most probably, this was not developed simply because RED was cross optimising only between datagram and transport levels, hence all IP datagrams are considered equal.

When application level concerns need to be addressed the firewall or a load balancing technology is the right answer: here datagrams are dropped as explicitly prescribed by the firewall rules derived from business goals and from application policies. Unfortunately, it is practically impossible to prescribe anomalies; hence the conventional firewall technology is not as helpful as needed. However, in 2009 Hirsch and Varas have proposed D-CAF (Distributed Context Aware Firewall)[14]. Their approach in short can be summarised as follows:

- Monitor flows in an aggregate of interest and build rich metric of an aggregate and infer from the metric relative importance of each flow - this is “network opinion” on flows, for scalability could keep the opinion only for most important flows;
- Applications (or rather application level platforms) monitor on-going workflows and valueate [see below] related datagram flows; valuations are communicated to the network access router, aggregated and propagated downstream to all respective decision points (like RED, firewalls, etc.) - these are aggregated application level opinions on flows (valuations are basically utility policies);
- Decision process (e.g. in a firewall but actually in any point where requests for resources are multiplexed) takes into account both opinions.

This double evaluation – of workflows at platform (or, infrastructure) level and of traffic flows at datagram level – is depicted schematically at Figure 91, which basically exhibits a multi-source (utility) policy exchange.

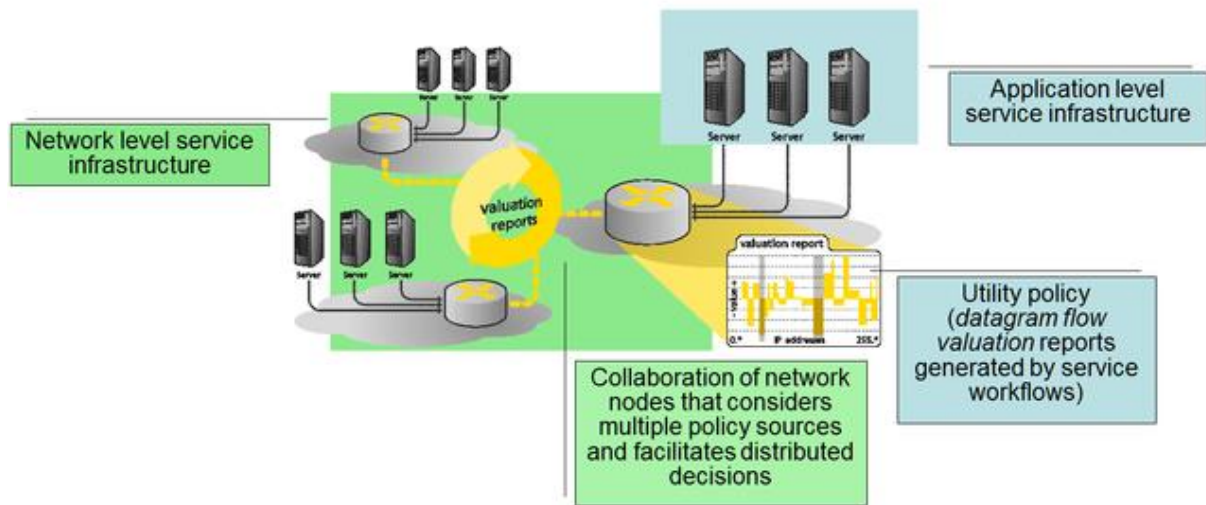


Figure 91: Cross-layer optimisation in D-CAF

This technology appears to be important for XIFI because it helps to achieve traffic engineering and traffic-based anomaly detection. One question still remains – how to generate utility policies⁹?

A simple (provided that all executed workflows are known and registered in advance) answer could be as follows:

- Each correct (normal, expected) workflow behaviour implemented by a resource request flow (datagram flow, compute request, storage request) results in incrementing of this flow valuation;
- Each negative, unexpected workflow behaviour implemented by a resource request flow (datagram flow, compute request, storage request) results in decrementing of this flow valuation.

Let us note that the above described mechanism is most suitable for a reputation service.

Additionally to reputation service based on utility generation and consequent prioritization of workflows XIFI infrastructure providers might wish to implement a strict access control to all or some (perhaps, most critical) resources. This mechanism can be termed Workflow-based Access Control (WAC) and is described in the next section.

A.4 Security Benefits

Workflow-based Access Control (WAC) is significantly different from a simple access control list (ACL) and from a role-based access control (RBAC) and is not aimed to replace them. On contrary, the power of WAC is to be revealed in a joint deployment with other mechanisms. In particular the synergy between RBAC and WAC appears to be very promising. RBAC has many benefits. First, it is known to have about 2 magnitudes smaller complexity than ACL, thus it scales in principle for large

⁹ We must be able to dynamically generate these policies since they cannot be known in advance due to a complex multiplexing nature of federated cloud infrastructure.

federations. Second, being based on a structured set of roles¹⁰ it automates permission propagation schemes as shown below in Figure 92. However RBAC is also known to have conflicts mainly due to multiple permissions inheritance and to multiple dynamic roles assignments also shown below in **Error! Reference source not found.** The latter seems to be a common case in almost any federated infrastructure with a typical use case being of a Principle Investigator delegating certain work to her student, etc. The WAC offers a cure to this problem by including the role assignment into a workflow description, thus the two sessions shown in Figure 92 will be considered as two different workflows.

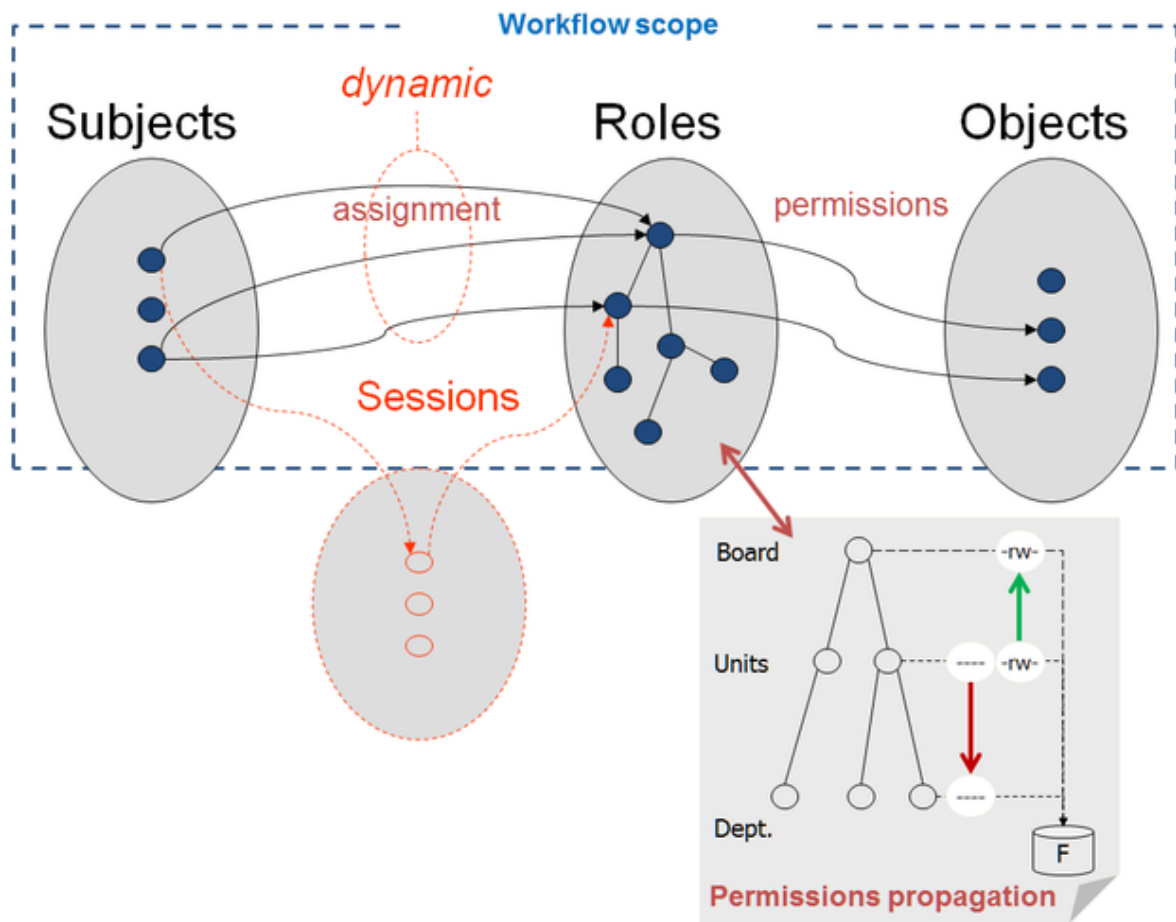


Figure 92: The scope of WAC

This feature of WAC allows not only to avoid conflicts typical to RBAC but also to detect anomalies by monitoring workflow execution. This will require a federation-wide workflow repository and its proper maintenance.

The workflows and infrastructure protection by WAC can be seen as an extension of policy. Workflow invocation is equivalent to dynamic creation of a policy domain spanning all the resources targeted by a workflow, the authorization policy established during the role assignment phase is then propagated along the workflow. Note, however that what is actually propagated is a subject's SSO (or, perhaps better to term it a SSO container), while authorization policies being instantiated along the invoked workflow are that set by infrastructure providers.

¹⁰ Reflecting the structure of an organisational hierarchy, or reflecting the structure of relations between stakeholders, otherwise.

This brings powerful flexibility: authorization policy belongs to a workflow but is being set by an infrastructure provider respective or irrespective particular workflow.

Speaking in a policy language we would term a workflow itself being made of obligation policies, predicated by SSO containers for authorization policies, while authorization policies as always are being set on objects. Implementation-wise Policy Enforcement Points (PEP) can use pointers to the instances of authorization policies, and when a workflow is being executed these pointers will replace SSO containers.

Like telephone networks were designed with the **Trust By Wire** principle in mind, the main principle we want to investigate here is the **Trust By Workflow**, meaning that infrastructure nodes that cooperate under multiple workflows can eventually elaborate significant trust based on successful history of common work.

A.5 How to Trust by Workflow

Structuring the field

Trust is a key requirement for the successful uptake and operation of Utility and Cloud Computing. Service consumers must have belief that their data is safe and protected and service providers must provide sufficient assurances to satisfy these beliefs. However many challenges need to be overcome before this is possible. A myriad of trust relationships can arise between the different players in the UCC service delivery chain and these needs to be understood and described as do the consequential assurance approaches that may be required. Complex, and varying, service delivery environments within federated clouds may require very different and context dependent mechanisms to provide the assurances needed. Thus there is a strong need to explore approaches to the definition and establishment of trust relationships in such dynamic and evolving service environments and to investigate mechanisms, methodologies and technologies to enable the provision of assurances that are needed to fulfil users trust beliefs. Thus, [18] defines the relevant areas of research from which we copy below those relevant for the XIFI operation and for the design and implementation of subsequent OLA and workflows:

- Trust models.
- Cryptographic techniques for privacy preservation
- Identity, Authentication and key management
- Formal verification for cloud architecture
- Practical cryptographic protocols for cloud security
- Intrusion Detection Technologies for cloud contexts
- SDN Security
- Provenance and digital forensics
- Monitoring systems in the cloud
- Remote attestation mechanisms in clouds
- Trusted computing technology and clouds
- Cloud Integrity and Audit
- Multi-tenancy and trust in cloud computing

In a single project it is not possible to address systematically all the above topics though all of them are relevant. Deliberately and consistently with the goals of this section we consider a workflow – first, as an abstract object, and, finally, as executable object and as a process of its execution – as an integral unit of trust, to which all of the above topics might refer to. Again, in consistency with the approach perceived in this section we analyse how the federation roles (section 4.1) exhibited by the stakeholders that are relevant for different steps of a workflow life-cycle. We explicitly refer to a

workflow as to a single unit of trust for all stakeholders inside a federation (those marked in bold in section 4.1) and outside of a federation (these are users /developers). As Figure 93 demonstrates (in a form of a concept map) the relations of the stakeholders to a workflow are not symmetric.

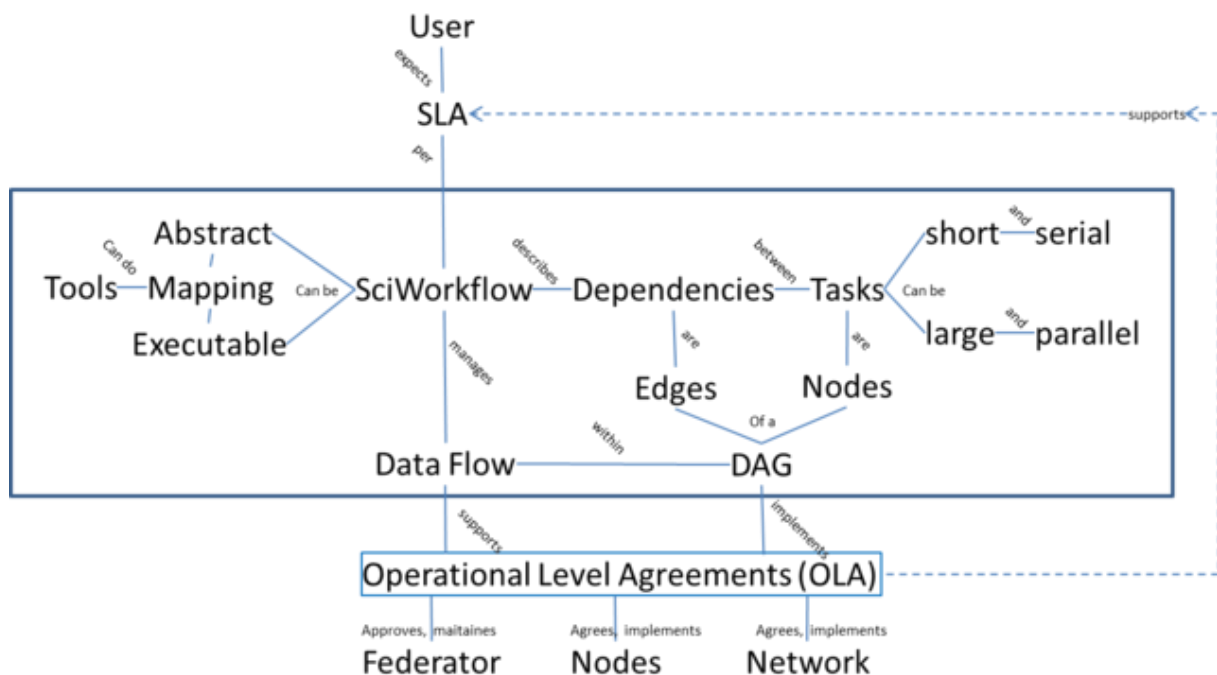


Figure 93: Workflow as a Unit of Trust in OLA

For a User, OLA is not visible at all however expecting certain SLA a User is nevertheless experiencing the quality of OLA, while all roles that are inside a federation are directly responsible for agreeing and implementing OLA (infrastructure owners and operators termed “Nodes” as well as Network provider / operator), as well as for OLA approval and maintenance (Federator).

Achieving operational trust by workflow

We detail possible workflow life cycle and show stakeholder relations for each step. To make the description more precise we bear in mind a possible speculative workflow outlined in Figure 94.

User Alice has developed a big data analytics software package that she wants to run concurrently on as many nodes of a federation as possible however within certain cost-utility envelope. The workflow of this big data analytics is as follows: SENSE modules being deployed on nodes collect primary metrics of node operation and feed those to STAT module, which does certain statistical analysis of primary metrics and distributes the results to CTRL modules. Depending on user-configured thresholds the CTRL modules decide on i. configurations of SENSE modules; ii. deployment of new SENSE modules or stopping existing SENSE modules; iii. deployment of new CTRL modules or stopping existing CTRL modules. All workflow modules operate in slotted time. There is always only one STAT and at least one SENSE and one CTRL module. The workflow operation stops after pre-defined number of time slots (normal operation) or on impossibility to continue the operation (fault condition) For the sake of this example it is enough to consider that the cost-utility envelope if being defined within the workflow like this: the STAT module computes certain workflow utility metric, while deployment of each new SENSE of CTRL bears certain cost.

Figure 94: Sample workflow

Workflow Creation

A workflow is being created by a User (a variety of tools are available) following a usual process that includes requirements analysis, design, debugging, and testing. A viable federation can be a part of this creation process as outlined in Table 87.

Role\Phase	Requirements	Design	Debugging	Testing
IO	Formulate requirements in a way conformant to federation SLA	Design an abstract workflow conformant to federation resources and services	Run a workflow in a sandbox	Capture workflow specific metrics
Node	Allow SLA retrieval and analysis of resources and services	n/a	Provide sandbox nodes with capabilities conformant to a federation	Allow configurable measurements in sandbox nodes
Network	Allow SLA retrieval and analysis of connectivity options and KPI's	n/a	Provide sandbox network between sandbox nodes with capabilities conformant to a federation	Allow configurable monitoring of sandbox network
Federator	Commonly agreed SLA terms and conditions	n/a	Commonly agreed layout of sand-boxing	n/a

Table 87: Workflow (maintenance procedure) creation process

Workflow Registration

After a User considers that her workflow is successfully designed and tested she triggers the process of workflow registration with the federation. At the time of this writing, also bearing in mind an example workflow introduced above, it is reasonable to structure the workflow registration into four parts as shown in Table 88

Role\What	WF Schema	Node operation	Network operation	WF data management
User	Abstract WF is digitally signed by a User and published for approval at a federation repository	User either enumerates nodes that might be involved or defaults to any available subset of nodes	If required nodes are listed the connectivity options between them must be detailed, otherwise a user agrees to best effort	Required rollback points of a workflow must be specified.
Node	n/a	Capabilities of enumerated nodes must be checked	n/a	Allow deployment of rollback capacity (additionally)
Network	n/a	n/a	Capabilities of enumerated connections must be checked	Maintain sufficient connectivity between active nodes and rollback points
Federator	Maintain common WF repository	Push node-related parts of WF schema to involved nodes	Push network-related parts of WF schema to Network	Push rollback-related parts of WF schema to involved nodes and Network

Table 88: Workflow registration process

It is obvious that Table 88 outlines a scenario, in which a Federator is a trigger for validation of a newly submitted workflow; in practice a federation might implement another scenario. However this one was selected because it appears to be in-line with the XIFI Use Case 5 [6].

Workflow Eligibility

After a workflow is registered and the Federator has pushed parts of its schema to all involved operators it is possible to launch a process of agreement on workflow execution, after which a workflow becomes eligible. During this process, which might be also specified in detail as a part of OLA, all prerequisites of a workflow invocation are collected from all involved operators. These are:

- terms and conditions of usage of resources and services that might be involved in a workflow execution;
- access policies for the above resources and services for involved nodes and network;
- workflow access control specification (dynamic and / or static assignment of subjects to roles within a workflow as defined in a WAC algorithm (see “Security Benefits” section);
- SLA restrictions that potentially can apply from a deficit of OLA support.

These prerequisites are collected by the federator from node and network operators based on the abstract workflow description provided by a user, and stored in a workflow repository so that the structured data set is: <abstract workflow>; <node and network prerequisites>; <WAC conditions> and <SLA restrictions>.

The data set is neither an abstract workflow, nor an executable one; it is a workflow ready for parameterisation with instant values of prerequisites and assignments provided that they are

1. within specified ranges, and
2. the combination of instant values makes them mutually eligible.

Role	WAC assignment	Terms and conditions	Access policies	SLA restrictions
User	X			
Node		X	X	
Network		X	X	
Federator				X

Table 89: Separation of concerns between the major roles

As Table 89 demonstrates there is a clear separation of concerns between the three major roles; a User bears full responsibility for such subject to role assignment (WAC assignment column) that workflow remains eligible, while a federator bears full responsibility for computing possible SLA violations (SLA restrictions column). It should be noted that since the amount of possible combinations of all parameters outlined in Table 89 can be very large it will not be always possible to compute the probability of exact SLA violations. Therefore it can be recommended that a Federator computes a pessimistic estimate of such probability.

Workflow maintenance

In this section we briefly outline important aspects for workflow maintenance that shall help to sustain a XIFI federation.

Workflows are subject to aging and, more important, to multiple exceptions. To cope with these issues a Federator must implement a procedure, when modifications of a workflow require a new registration and subsequent eligibility check so that a workflow clone, while keeping an inheritance relation with a parent workflow, it is nevertheless a new workflow instance. This procedure will eventually reduce the amount of SLA violations during the workflow invocations and by this directly improves the quality of OLA.

Scientific workflows are usually made from interleaving technical and non-technical branches of workflows that can be implemented as lawful interrupts, during which a user does certain off line operations. This will require that workflows allow human-to-cloud communications, and a reasonable choice for those nodes in a workflow would be rollback points.

The above process of workflow eligibility check must be considered as the first step in permanent workflow validation (run-time testing and debugging), which naturally leads to a system that shall support workflow evaluation in terms of its current reputation and root cause analysis of detected anomalies and conflicts between workflows.

As the first step towards inter-workflow conflict detection and avoidance it is reasonable to perform rigorous information modelling of all eligible workflows – the way to guarantee compliance to OLA. This topic however is beyond the scope of current work; we outline possible directions to this in the next section.

A.6 Future work: Workflow Manifesto

This section outlines possible future work towards the precise definition of a commonly agreed workflow format – workflow manifesto - to be used for its registration. This future work shall be based on the study of BCP's and on a commonly agreed content of the previous section (A.5), since most important elements of XIFI-relevant workflow descriptions should be tailored to the project; obviously we need more examples to learn from. In particular, it appears important to collect opinions from:

- XIFI infrastructure providers on the importance and on associated difficulties of the proposed approach – this will allow to set priorities and to use available effort to implement most critical part(s) of the proposed work.
- XIFI partners and/or their customers on the importance¹¹ of the proposed work for their business developments and/or relations beyond the life of the project.
- XIFI partners and/or their customers on the opportunities of the proposed work within standardisation.

¹¹ The importance may differ for the two cases: i) workflows as part of OLA, ii) Workflows without OLA.

Appendix B Further Details on OLA

B.1 OLA Scheme Description

The following presentation and discussion of the service categories for infrastructure operators is oriented along the ITIL SLA/OLA template [16] which has been adapted to the purpose here. The OLA template includes the sections:

- **OLA Name**
OLA ID and name allows for an identification of the agreement
- **Stakeholders**
In this section we list the stakeholders involved in the OLA. For the specific clearance information with regard to infrastructure operators, XIFI is maintaining an internal wiki-page “Fi-ppp:Management of XIFI Nodes”.

OLAs often define a duration for the contract. Within XIFI agreements should range over the lifetime of the project¹² and should see revisions and updates as needed.
- **Service Description**
This section should give a short description of the services covered together with a rationale and the context in which it is performed. It identifies processes and workflows that are connected to the service and describes targets that should be achieved in the execution.
- **Preconditions**
Requirements and conditions that need to be fulfilled for the service to be operational.
- **Communication**
Regular reports on the service execution should be generated. This requires specifying which information has to be gathered and the time interval when reports should be delivered.

Procedures for handling exceptions should be defined including agreed response times and escalation procedures.

It could also include procedures for measuring goal satisfaction, and review of the services and the agreement on a regular basis.

For maintenance operations by the XIFI infrastructure operator it is important that such activities are announced federation wide to peer nodes, as well as to the node users. There should be XIFI wide information policy in place that defines the media (XIFI portal, direct email) and the information times (e.g. minimal period before scheduled maintenance)
- **Criticality**
In order to respond to incidents accordingly, it is import to rank the criticality of services and incidents which allows defining countermeasures and reaction times that respect the

¹² After the XIFI project ended, a different situation applies since both stakeholders (e.g. infrastructure operators) and business objectives may change (e.g. assurance of SLAs instead of best effort). However, the approach described here does not change.

prioritization of a detected incident. The classification scheme should reflect the business impact caused by a loss of the service or related data assets. Vital business functions and critical assets should receive prioritized effort and reaction times.

- Service times
This includes the available times, e.g. regular business hours, and possible exceptions such as public holidays
- Required types and levels of support
This specifies conditions under which the service is provided, e.g. areas where the service available, user groups which are entitled to use the service, technical requirements that need to be fulfilled on the customer side, prioritization schemes with response times.
- Operational level requirements / targets
 1. Availability targets and commitments
The definition of availability targets requires first an exact definition on when and under what conditions the service is considered available. Availability targets are then calculated based on agreed service time and downtime. Reliability targets are typically expressed as MTBF (Mean Time Between Failures) or MTBSI (Mean Time Between Service Incidents). Maintenance is characterized by MTRS (Mean Time to Restore Service) and OLAs will include specification and metrics for maintenance downtimes such as number of allowed down times, pre-notification period for scheduled maintenance, allowed maintenance windows.
The OLA should specify procedures to handle incidence and emergency changes and how to announce such unplanned service interruptions.
 2. Capacity/ performance targets and commitments
Service delivery should also be specified in quantitative measures expressing provided capacity and performance. Such quantitative measures are specific to the service.
Monitoring results of the measured availability and service performance should be published in regular reports.
 3. Service Continuity commitments
This defines metrics that measure the availability of the service after disruptions due to incidents. Commitments regard the time until a certain defined level of service has to be re-established and by which full service levels must be restored.
Service disruption should be minimized and the system should be engineered to allow for graceful degradation, preferably offering at least lower level of service instead of complete service failure.
- References
Within the context of XIFI this section mainly references related procedures and processes. It may include also further technical standards, or e.g. specification of the service interface.
- Responsibilities
This includes the respective duties of the provider and consumer of the services, responsibilities of service users e.g. to comply with the federation security policies.
XIFI Infrastructure operators are required to be compliant with the procedures and workflows for node operation, infrastructure monitoring and GE hosting as set out in

D5.1: Procedures and Protocols for XIFI federation, section 1.4 “XIFI federation”.

- Pricing model

This OLA section specifies the modalities of service charging, the cost for the service provision and possibly rules for penalties, charge backs and compensations. Since XIFI services in the project are delivered on an as-is or best-effort basis, accounting and charging doesn't apply at the moment, however when commercializing a XIFI system and offering a carrier-grade service platform those aspects become relevant.

In the following sections we describe a few examples of OLAs. It should be noted that OLAs are a long term feature, to be defined, reviewed and fine-tuned over time. It is clear that also other discussions at business levels have to be performed for that [4]. Below sections are thus a useful and needed step on the way towards achieving that.

B.2 OLA Computing and Storage Resources Operation and Maintenance

OLA Name

Computing and Storage Resources Operation and Maintenance

Stakeholders

Infrastructure Operator, Federator

Service Description

XIFI infrastructure operators offer computing and storage resources to the XIFI federation on which the XIFI platform can be operated.

The activities related with the provisioning of computing and storage resources are monitoring and supervision of correct operation and system health, as well as service levels associated with maintenance processes. Maintenance comprises scheduled maintenance processes which can be planned and announced in time to cause minimal disruptions, e.g. system upgrades; unscheduled maintenance, such as the need for reconfiguration due to performance problems or replacement of defective components; and responses to incidents, e.g. power outages, damages to hardware etc.

Preconditions

The provisioning of computing and storage resources requires that the infrastructure operator has successfully completed the joining procedure to the XIFI federation as described in Procedures and Protocols for XIFI federation, Deliverable D5.1, chapter 5: Procedures for Joining the federation, in particular the operator has to fulfil the requirements as defined in chapter 5.2.

Communication

Regular reports on the service execution should be generated. This requires specifying which information has to be gathered and the time interval when reports should be delivered.

Also procedures for handling exceptions should be defined including agreed response times and escalation procedures.

It could also include procedures for measuring goal satisfaction, and review of the services and the agreement on a regular basis.

Criticality

Computing and storage are fundamental services on which all others build on. Failures can cause major disruptions, however depending on higher layer services, failures of basic components can be masked from the user by fast failover mechanisms.

Service times

The basic infrastructure should be available in general 24/7. Service outage due to maintenance operations should be minimized and performed preferentially during off office hours to cause minimal disruption.

Operational level requirements / targets

Availability and performance data for XIFI nodes is captured through the XIMM (XIFI monitoring middleware) service, which will provide mechanisms for multi-domain measurement and unified control and access to performance metrics of the infrastructures.

In particular, the XIMM-Datacentre and Enablers Monitoring (XIMM-DEM) Module focusing on datacentre-based metrics can be used to collect performance data from hosts and services to validate OLA availability and performance targets. Performance targets must be defined prior to implementing SLAs. Basic information on the OpenStack installation is gathered by OpenStack Data Collector module. Information collected is capacity data as number of virtual machine deployed, number of cores available, size of ram and size of disk, number of users/tenders registered.

B.3 OLA Network Connectivity Operation & Management

OLA Name

Network Connectivity Operation & Management

Stakeholders

Infrastructure operators among each other, Federator, Network Connectivity Provider

Service Description

Networking belongs besides computing and storage to the fundamental resources offered in the XIFI cloud. Network connectivity connects the nodes in the federation and can in principle be extended to include users as 3rd parties. The distributed nature of XIFI cloud services is one of the distinctive features of the platform as it allows users to conduct large scale networking trials across Europe.

Connectivity between nodes is provided via a 3rd party network connectivity provider, which in the case of XIFI in general is provided by the national NRENs and GÉANT. This means that the OLA concerned with networking is dependent on and requires alignment with the underpinning contracts (UC) with the 3rd parties.

The activities related with network connectivity are matching those of other fundamental resources. They comprise monitoring correct operation, verifying connectivity and maintenance processes. Maintenance comprises scheduled maintenance processes, e.g. capacity upgrades and reconfigurations, unscheduled maintenance, e.g. to solve performance problems, or actions triggered as reaction to incidents on other nodes, and responses to incidents, like damages to hardware, equipment failures or loss of connectivity due to cable breaks.

Preconditions

The node has successfully joined the federation and is connected to the federation VPN, monitoring is in place

Communication

Regular reports on the connectivity status should be generated. Such statistics should cover general reachability and quantitative measures of capacity and QoS characteristics, such as delay, jitter and loss rates.

The XIMM monitoring system includes modules for active and passive network monitoring which allows collecting the necessary raw data. Federation monitoring makes the information accessible to

peer nodes and service users.

Criticality

Disruption cause wide-range unavailability of all node services, unless there is redundant node connection which allows for re-routing traffic, such that line failures can be mitigated.

Service times

The basic infrastructure should be available in general 24/7. Service outage due to maintenance operations should be minimized and performed preferentially during off office hours to cause minimal disruption.

Required types and levels of support

Infrastructure operators need to comply with the technical and operational requirements. Maintenance of the physical communication infrastructure needs to be performed according to the respective policies defined in D5.1, Sec. 5.2.1 Physical Infrastructure.

Operational level requirements / targets

Availability and performance data for XIFI nodes is captured through the XIMM (XIFI monitoring middleware) service, which provides mechanism for multi-domain measurement and unified control and access to performance metrics of the infrastructures.

In particular, the XIMM-Network Active Monitoring (XIMM-NAM) Module and XIMM-Network Passive Monitoring (XIMM-NPM) Module provide performance data with respect to connectivity which allow validating OLA availability and performance targets.

B.4 OLA Non-conventional Resources

OLA Name

Non-conventional Resources

Stakeholders

Infrastructure Operator, Federator, XIFI user

Service Description

XIFI infrastructure operators may offer specialized resources besides classical cloud resources. Such capabilities may cover resources such as a Sensor Network, Mobile Network and in general other features different from the conventional data centre. In general XIFI assumes in those cases a more direct interaction between infrastructure operator and user of non-conventional resources. The underlying federation model is that of the federation acting as broker for the non-conventional resources and no longer being a central integrator.

For non-conventional services, the integration into the XIFI federation currently extends only to providing brokerage and supporting resource discovery. The resource catalogue provides generic contact and availability information, while the negotiation of the offer details is performed on direct peer-to-peer basis, and hence there is established a direct SLA set-up between resource user and infrastructure operator.

A future larger integration of non-conventional resource will require an inclusion of resource and usage monitoring to those special resources.

Preconditions

Integration of the non-conventional resources into the federation services, especially extension of the XIMM to collect relevant monitoring data from those resources.

Communication

The procedures should be comparable to those used in standard resource provisioning. It should include regular reports on the service execution and procedures for handling exceptions.

Criticality

It is expected that non-conventional resource will be available only by few nodes in the federation, in many cases just by a single node. In such case there is little redundancy, and service disruptions cannot be masked by migrating users to different nodes.

Service times

Availability may be more restricted as in comparison to standard resources, e.g. certain resources may require the availability of human supervisors and hence would be restricted to regular business hours.

Operational level requirements / targets

Availability and performance data should be provided similar to standard resources. This requires the integration of monitoring tools for those resources via specific adaptation components into the XIMM (XIFI monitoring middleware) service.

B.5 OLA User Support

OLA Name

User Support

Stakeholders

Infrastructure Operator, Federation helpdesk, XIFI User, Level 3 Experts

Service Description

The XIFI federation provides technical support for Future Internet developers as XIFI users through the helpdesk on the XIFI federation office portal. Infrastructure operators act here as level 2 support for the infrastructure users. The support team further analyses, diagnoses and isolates the problem. The process may involve an escalation to level 3 where the issue is delegated to experts either outside (e.g. FIWARE expert) or inside to infrastructure experts for further troubleshooting and resolution of the problem.

Preconditions

A user ticket has been forwarded from the helpdesk to the infrastructure operator for 2nd level user support.

Communication

The infrastructure support team is integrated into the overall user support workflow as described in the project-internal Wiki page "Procedures and Protocols for XIFI federation", Sec. 4.3: Flow Description. It involves problem analysis, possibly problem delegation to external or internal experts, solution validation and solution forwarding.

Criticality

The User Support OLA is directly related to XIFI user SLAs, hence there must be close alignment between response times agreed internally within the federation and those committed to the user.

Times need to respect the severity of problem and whether there exist already known solutions in the helpdesk knowledge database, in which case such information or references to solution can be forwarded immediately. The detected problem may be localized to single user, but may also be just the first announcement and hint to a more severe disruption which potentially could affect also other users.

Service times

Level 2 support service from the infrastructure operator should be available during regular business hours.

Required types and levels of support

User support is provided to registered XIFI users that make use of resources offered by the infrastructure operator.

Operational level requirements / targets

OLA response times need to be aligned with the response times agreed in the user SLAs. It is important, that a user receives a fast early response. This signals that the ticket is being processed and has been forwarded to the responsible support team. Direct contact for feedback and further problem exploration can be established. The processing times will depend on the severity of the problem, whether the problem is known and that there are existing validated solutions or whether an escalation to support level 3 is required.

The ticket system allows tracking the steps undertaken and to monitor whether agreed response times are fulfilled.

B.6 OLA Federation Services and Software Management

OLA Name

OLA Federation Services and Software Management

Stakeholders

Infrastructure Owner among each other, Federator

Service Description

Federation related services span federation networking services building on L3 MD-VPN and alternative solutions and the common services available over this infrastructure. It includes the operation and maintenance processes for core backbone connectivity, maintenance of federation tools and FI services.

Associated procedures for federation networking and federation service maintenance have been defined in D5.1, Sec 3.4 and Sec. 5.2, and validation tests for compliance to technical and operational requirements are defined in D5.2.

Those activities require close coordination between the infrastructure operators and the federation office. Compliance to the procedures and workflows is highly necessary, since failures in this area at one node may even affect the correct operation of other nodes, and nodes may risk losing connectivity, hence all of their customer services will become unavailable. Software updates must respect the fixed time frame to avoid inconsistency within the federation.

Preconditions

The infrastructure operator is member of the XIFI federation.

Communication

General information on the availability and status of federation resources and services is made accessible to peer nodes and service users via federation monitoring.

This includes information on the status of the federation network and the services available on the node.

Maintenance procedures with risk of service disruption and unavailability should be announced federation wide according to pre-defined policies.

Criticality

Since disruption and failures on the federation level may cause wide range service unavailability, incidents on this level have highest criticality.

Service times

Services should be available in general 24/7. Service outage due to maintenance operations should be minimalized and performed preferentially during off office hours to cause minimal disruption.

Required types and levels of support

Mutual infrastructure support should be provided according to the policies defined under D5.1 Sec. 3.5 Federation Joining Support Levels

Operational level requirements / targets

The XIFI federation aims at achieving an availability of > 95%, which corresponds to an accumulative downtime of < 3 weeks per year.

Infrastructure operators need to comply with the minimal technical requirements for connectivity and resources.

Maintenance targets should define the time frame within which federation wide upgrades need to be performed.

Responsibilities

Infrastructure operators need to comply with the technical and operational requirements defined for the XIFI federation

B.7 OLA Security & Privacy

OLA Name

Security & Privacy

Stakeholders

Infrastructure Operator, Federator, other Infrastructure Operators, XIFI users, XIFI technology providers

Service Description

Security and privacy are cross-domain concerns that extend to all interactions with other stakeholders that the infrastructure operator is involved with.

The XIFI federation supports security functions for federated security comprising functions for identity management, authentication (single sign on), authorization, access control, security proxy and security monitoring. Access to security related monitoring data is provided via the XIFI Security Dashboard. Security Probes component are responsible to collect security monitoring data and send them to the master node

Preconditions

Security probes, security related components and Dashboard in place.

Communication

Reports on security risks are accessible via the Security Dashboard.

Criticality

In general incidents related to need to receive high priority and fast responses. Security monitoring is based on CVSS (Common Vulnerability Scoring System) which supports in assessing the severity of

vulnerabilities.

Service times

Depending on the severity of incidents, responses outside of regular business hours may become necessary.

Required types and levels of support

Security incidents need to be communicated to peer nodes and to potentially affected XIFI users.

Operational level requirements / targets

Service Continuity commitments include metrics characterizing the availability of the service in the event of a disaster. Corresponding maintenance procedures for recovery from security-related events are currently under development in task 5.4

Responsibilities

Stakeholders need to comply with the terms of use, security and privacy policies and acceptable use policies in place for the federation

Appendix C Procedure to Add the Required Images

- kernel_repository-image-R3.2:
 - Public: Yes
 - Protected: No
 - Name: kernel_repository-image-R3.2
 - Status: active
 - Size: 3941424
 - Disk format: aki
 - Container format: aki

\$ glance image-create --name kernel_repository-image-R3.2 --disk-format aki --container-format aki --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=58 --file <name of the file of the corresponding downloaded image>

- ramdisk_repository-image-R3.2:
 - Public: Yes
 - Protected: No
 - Name: ramdisk_repository-image-R3.2
 - Status: active
 - Size: 23330374
 - Disk format: ari
 - Container format: ari

\$ glance image-create --name ramdisk_repository-image-R3.2 --disk-format ari --container-format ari --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=58 --file <name of the file of the corresponding downloaded image>

- repository-image-R3.2-2:
 - Public: Yes
 - Protected: No
 - Name: repository-image-R3.2-2
 - Status: active
 - Size: 10737418240
 - Disk format: ami
 - Container format: ami

\$ glance image-create --name repository-image-R3.2-2 --disk-format ami --container-format ami --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=58 --property kernel-id=<id of aki image:kernel_repository-image-R3.2> --property ramdisk-id=<id of ari image:ramdisk_repository-image-R3.2> --file <name of the file of the corresponding downloaded image>

- dbanonymizer-dba:
 - Public: Yes
 - Protected: No
 - Name: dbanonymizer-dba
 - Status: active
 - Size: 3339124736
 - Disk format: qcow2
 - Container format: ovf

```
$ glance image-create --name dbanonymizer-dba --disk-format qcow2 --container-format ovf --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=64 --file <name of the file of the corresponding downloaded image>
```

- marketplace-ri_2:
 - Public: Yes
 - Protected: No
 - Name: marketplace-ri_2
 - Status: active
 - Size: 10737418240
 - Disk format: ami
 - Container format: ami

```
$ glance image-create --name marketplace-ri_2 --disk-format ami --container-format ami --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=95 --property kernel-id=<id of aki image:kernel_repository-image-R3.2> --property ramdisk-id=<id of ari image:ramdisk_repository-image-R3.2> --file <name of the file of the corresponding downloaded image>
```

- kernel-meqb-image-R2.3:
 - Public: Yes
 - Protected: No
 - Name: kernel-meqb-image-R2.3
 - Status: active
 - Size: 4960752
 - Disk format: aki
 - Container format: aki

```
$ glance image-create --name kernel-meqb-image-R2.3 --disk-format aki --container-format aki --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=142 --file <name of the file of the corresponding downloaded image>
```

- ramdisk-meqb-image-R2.3:
 - Public: Yes
 - Protected: No
 - Name: ramdisk-meqb-image-R2.3
 - Status: active
 - Size: 14207719
 - Disk format: ari
 - Container format: ari

```
$ glance image-create --name ramdisk-meqb-image-R2.3 --disk-format ari --container-format ari --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=142 --file <name of the file of the corresponding downloaded image>
```

- meqb-image-R2.3:
 - Public: Yes
 - Protected: No
 - Name: meqb-image-R2.3
 - Status: active
 - Size: 10737418240
 - Disk format: ami
 - Container format: ami

```
$ glance image-create --name meqb-image-R2.3 --disk-format ami --container-format ami --min-disk
0 --min-ram 0 --is-public True --is-protected False --property nid=142 --property kernel-id=<id of aki
image:kernel-meqb-image-R2.3> --property ramdisk-id=<id of ari image:ramdisk-meqb-image-R2.3>
--file <name of the file of the corresponding downloaded image>
```

- cep-image-R2.3:
 - Public: Yes
 - Protected: No
 - Name: cep-image-R2.3
 - Status: active
 - Size: 4028891136
 - Disk format: qcow2
 - Container format: ovf

```
$ glance image-create --name cep-image-R2.3 --disk-format qcow2 --container-format ovf --min-disk
0 --min-ram 0 --is-public True --is-protected False --property nid=146 --file <name of the file of the
corresponding downloaded image>
```

- datahandling-ppl:
 - Public: Yes
 - Protected: No
 - Name: datahandling-ppl
 - Status: active
 - Size: 3339124736
 - Disk format: qcow2
 - Container format: ovf

```
$ glance image-create --name datahandling-ppl --disk-format qcow2 --container-format ovf --min-disk
0 --min-ram 0 --is-public True --is-protected False --property nid=216 --file <name of the file of the
corresponding downloaded image>
```

- orion-psb-image-R3.3:
 - Public: Yes
 - Protected: No
 - Name: orion-psb-image-R3.3
 - Status: active
 - Size: 4056023040
 - Disk format: qcow2
 - Container format: ovf

```
$ glance image-create --name orion-psb-image-R3.3 --disk-format qcow2 --container-format ovf --
min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=344 --file <name of the file
of the corresponding downloaded image>
```

- kernel_registry-ri:
 - Public: Yes
 - Protected: No
 - Name: kernel_registry-ri
 - Status: active
 - Size: 4960752
 - Disk format: aki

- Container format: aki

\$ glance image-create --name kernel_registry-ri --disk-format aki --container-format aki --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=465 --file <name of the file of the corresponding downloaded image>

- ramdisk_registry-ri:
 - Public: Yes
 - Protected: No
 - Name: ramdisk_registry-ri
 - Status: active
 - Size: 14207719
 - Disk format: ari
 - Container format: ari

\$ glance image-create --name ramdisk_registry-ri --disk-format ari --container-format ari --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=465 --file <name of the file of the corresponding downloaded image>

- registry-ri
 - Public: Yes
 - Protected: No
 - Name: registry-ri
 - Status: active
 - Size: 10737418240
 - Disk format: ami
 - Container format: ami

\$ glance image-create --name registry-ri --disk-format ami --container-format ami --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=465 --property kernel-id=<id of akiimage:kernel_registry-ri> --property ramdisk-id=<id of ariimage:ramdisk_registry-ri> --file <name of the file of the corresponding downloaded image>

- ofnic-image-R2.3:
 - Public: Yes
 - Protected: No
 - Name: ofnic-image-R2.3
 - Status: active
 - Size: 10737418240
 - Disk format: qcow2
 - Container format: ovf

\$ glance image-create --name ofnic-image-R2.3 --disk-format qcow2 --container-format ovf --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=497 --file <name of the file of the corresponding downloaded image>

- kurento-R4.2.2:
 - Public: Yes
 - Protected: No
 - Name: kurento-R4.2.2
 - Status: active
 - Size: 5421465600
 - Disk format: qcow2

- Container format: ovf

\$ glance image-create --name kurento-R4.2.2 --disk-format qcow2 --container-format ovf --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=855 --file <name of the file of the corresponding downloaded image>

- kurento-image-4.0.0:
 - Public: Yes
 - Protected: No
 - Name: kurento-image-4.0.0
 - Status: active
 - Size: 5248581632
 - Disk format: qcow2
 - Container format: ovf

\$ glance image-create --name kurento-image-4.0.0 --disk-format qcow2 --container-format ovf --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=855 --file <name of the file of the corresponding downloaded image>

- kurento-image-R3.3:
 - Public: Yes
 - Protected: No
 - Name: kurento-image-R3.3
 - Status: active
 - Size: 10737418240
 - Disk format: raw
 - Container format: bare

\$ glance image-create --name kurento-image-R3.3 --disk-format raw --container-format bare --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=855 --file <name of the file of the corresponding downloaded image>

- cdva-image-R2.3:
 - Public: Yes
 - Protected: No
 - Name: cdva-image-R2.3
 - Status: active
 - Size: 3361538048
 - Disk format: qcow2
 - Container format: ovf

\$ glance image-create --name cdva-image-R2.3 --disk-format qcow2 --container-format ovf --min-disk 0 --min-ram 0 --is-public True --is-protected False --property nid=1099 --file <name of the file of the corresponding downloaded image>