



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



eXperimental Infrastructures for the Future Internet

D5.3: XIFI nodes operation, maintenance, assistance and procedures updates

Revision: v1.0

Work package	WP5
Task	T5.3, T5.4
Due date	30/06/2014
Submission date	19/08/2014
Deliverable lead	EURES
Authors	Rudolf Vohnout (CESNET), Uwe Herzog (EURES), Bernd Bochow, Mikhail Smirnov, Rudolf Roth (FRAUNHOFER), Joe Tynan, Eamonn Power (WIT), Matthias Baumgart (DT), Fernando López (TID), Matteo Pastorelli, Cristian Cristelotti (TN), Daniele Gai Pron (TI), Riwal Kerherve, Sergio Morant, Erwan Le Bonniec, Anthony Balan (ILB), Thierry Milin (Orange)
Reviewers	Patrik Arlos (BTH), Silvio Cretti (CNET)

Abstract	This deliverable provides an overview of the different activities related to the running of the XIFI nodes and updates the procedures initially defined in D5.1.
Keywords	Nodes operation, OLA, maintenance, support, helpdesk

Document Revision History

Version	Date	Description of change	List of contributor(s)
V1.0	19/08/14	Final and reviewed document version	Uwe Herzog (EURES) 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 Deliverable D5.1.

D5.1 had defined a first version 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, D5.1 also contains requirements and initial drafts of general procedures and procedures for developer support and maintenance.

The set-up and federation of the five initial nodes has enabled evaluating the procedures, protocols and tools defined in D5.1. The hands-on experience gained from that was very helpful to better understand technical, operational and legal requirements of a wide-area federated heterogeneous infrastructure. Besides updating the procedures and protocols defined in D5.1, D5.3 also contains significant enhancements e.g. regarding operations and support.

The extension of the federation of the five initial XIFI nodes has been ongoing. The first of the new nodes joining was the Prague node. A short report on the operation and maintenance of each of these six nodes is included in this Deliverable. At the time of completing this Deliverable most of the other new nodes have joined too or will complete the process of joining shortly. 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.fiware.org/> the current resource status of the XIFI federation is displayed. At the time of writing, the XIFI federation has 2789 federated users, 526 Organizations, 6 federated regions, 1712 CPU cores, 5154 GB RAM and 40607 GB of hard disk space with 997 operational Virtual Machines associated to the federation. So far, requests for support could be sent to a mailing list with administrators and other experts receiving and answering those questions and taking the required corrective actions.

The general procedures that are required to execute the procedures for maintenance and user / developer support are updated in this document from their initial draft in D5.1. The update mainly relates to the identification and assignment of operational roles, but also including some refinements. For example, for each of the 17 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. 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) levels with First Line Support (FLS) being, probably the most common starting level. We suggest to extend OLAs with workflows (WF), and examine some best current practices (BCP) known from various academic fields (e.g. Pegasus [8] software package) to underline

the benefit it yields. 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.

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. In the proposed mapping of the OLA concept onto XIFI we specifically look at the procedures that involve infrastructure operators. The various activities are grouped and categorized into a few domains that form the basis for potential OLAs. Finally regarding operational requirements and procedures, 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 etc. These definitions are built on the experience of XIFI partner TID gained from running FIWARE Lab (i.e. the former FI-Lab) legacy platform in FIWARE. 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 the maintenance process is outlined further detailing how the roles involved are acting on the maintenance subject. XIFI aims to utilize 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 has decided to use the JIRA platform 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. There will be JIRA collectors linked from the "Need help?" in the FIWARE Lab 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.

The document finishes with some conclusions.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	3
TABLE OF CONTENTS	5
LIST OF FIGURES	8
LIST OF TABLES	10
ABBREVIATIONS	12
1 INTRODUCTION	14
1.1 Context, Objective and Scope of this Deliverable	14
1.2 Intended Audience and Reading Suggestions.....	14
2 REPORT ON XIFI NODES OPERATION, MAINTENANCE, AND ASSISTANCE ..	16
2.1 Introduction.....	16
2.1.1 FIWARE Lab and XIFI federation integration.....	16
2.1.2 Federation monitoring of resources and operations.....	17
2.2 Waterford Node	18
2.2.1 Description.....	18
2.2.2 Experience on Support and Maintenance	19
2.2.3 Current Status	19
2.2.4 OpenStack Configuration	20
2.2.5 User-base	20
2.3 Trento Node.....	21
2.4 Berlin Node.....	22
2.5 Brittany Node (Lannion).....	24
2.6 Spain node (Seville/Malaga).....	26
2.7 Prague Node.....	30
2.8 Report on assistance and support.....	32
3 UPDATE ON GENERAL PROCEDURES	34
3.1 Management of nodes.....	34
3.1.1 Berlin	34
3.1.2 Brittany	35
3.1.3 Spain node	35
3.1.4 Trento.....	36
3.1.5 Waterford.....	36
3.1.6 IMINDS	36
3.1.7 ZHAW	37
3.1.8 PSNC	37
3.1.9 Neuropublic	37

3.1.10	CESNET	37
3.1.11	UPRC	38
3.1.12	Com4Innov	38
3.1.13	ACREO Swedish ICT	38
3.1.14	GOWEX	39
3.1.15	WIGNER	39
3.1.16	UTH	39
3.1.17	BTH	39
3.2	Developer support.....	40
3.3	Infrastructure support.....	41
4	UPDATES ON PROCEDURES FOR OPERATING THE FEDERATION.....	42
4.1	Stakeholders and roles in establishing and maintaining operational level agreements	42
4.2	Update on Support Process and Procedures for joining the federation.....	42
4.3	Scope and purpose of operational level agreements	43
4.4	Operational Level Agreements	45
4.4.1	OLA Level 2 Rationale.....	46
4.4.2	Workflows for cross-layer optimisation	49
4.4.3	Security benefits	50
4.4.4	How To Trust by Workflow	52
4.4.5	Future work: Workflow manifesto	56
4.5	OLA implementation in XIFI	56
4.5.1	Introduction.....	56
4.5.2	OLAs for XIFI Infrastructure Operator	57
4.6	Operational requirements and procedures	58
4.6.1	Tenant deployment	58
4.6.2	Basic Tenant deployment procedure.....	59
4.6.3	Tenant Life cycle	64
4.6.4	Traceability of deployed Instances	65
4.6.5	Local catalogue management.....	67
4.6.6	Managing Images.....	68
4.6.7	Managing Blueprints	68
4.6.8	Use Case Handling	73
4.6.9	Tenant customization.....	74
4.6.10	Node administration.....	74
5	MAINTENANCE PROCESS	78
5.1	Relation to the eTOM framework objectives.....	78
5.2	Stakeholders.....	80

5.3	Stakeholder interaction through the help-desk	85
5.3.1	Interaction of maintainers	86
5.3.2	Interaction of maintainers and developers	87
5.3.3	Ticket handling in the interaction of maintainers or developers.....	88
5.3.4	Notifications in ticket handling.....	91
5.4	Sub-systems subject to maintenance.....	92
5.4.1	Infrastructure node.....	92
5.4.2	Communication infrastructure	93
5.4.3	Software components.....	94
5.4.4	Software sub-systems	96
5.4.5	Procedures of the maintenance process	98
5.4.6	Scheduled maintenance (single infrastructure node)	100
5.4.7	Scheduled maintenance (multiple infrastructure nodes).....	102
5.4.8	Unscheduled maintenance (single infrastructure node)	104
6	SUPPORT TO FI-DEVELOPERS.....	106
6.1	Introduction to support levels	106
6.2	FI- developer support process and flows	106
6.3	FIWARE Lab Level 1 helpdesk.....	107
6.4	FIWARE Lab Level 2 / 3 support.....	109
6.5	JIRA ticketing process, flows and responsibilities	110
6.6	JIRA Administration.....	112
6.7	Reporting (JIRA statistics).....	112
6.8	FAQ and beginners guide	113
7	CONCLUSIONS.....	114
	REFERENCES.....	115
	APPENDIX A FURTHER DETAILS ON OLA.....	116
A.1	OLA Scheme Description.....	116
A.2	OLA Computing and Storage Resources Operation and Maintenance.....	118
A.3	OLA Network Connectivity Operation & Management.....	119
A.4	OLA Non-conventional Resources	120
A.5	OLA User Support	121
A.6	OLA Federation Services and Software Management.....	122
A.7	OLA Security & Privacy.....	123
	APPENDIX B PROCEDURE TO ADD THE REQUIRED IMAGES.....	125
B.1.1	Appendix level 3.....	129

LIST OF FIGURES

Figure 1: FIWARE Lab User Cloud portal	16
Figure 2: Operational capacity details.....	17
Figure 3: Regional cloud services status	18
Figure 4: Distribution of users accessing FIWARE Lab, per country (as of July 2014).....	29
Figure 5: Graphical Scheme of the Prague node	31
Figure 6: Usage in August 2014.....	32
Figure 7: Federation support procedures.....	43
Figure 8: Pegasus WF mapping	48
Figure 9: Cross-layer optimisation in D-CAF.....	49
Figure 10: The scope of WAC	51
Figure 11: Workflow as a Unit of Trust in OLA.....	53
Figure 12: Sample workflow.....	53
Figure 13: OLA Categories in XIFI	57
Figure 14: Account part.....	59
Figure 15: Create a network	60
Figure 16: Create a router.....	60
Figure 17: Add an interface.....	61
Figure 18: Create Security Group	61
Figure 19: Add rules.....	61
Figure 20: Define the Keypairs	62
Figure 21: Create an instance	62
Figure 22: Launch an instance	62
Figure 23: Instance Log.....	63
Figure 24: Allocate IP	63
Figure 25: Associate IP	63
Figure 26: Ping	64
Figure 27: Connect via SSH.....	64
Figure 28: Create blueprint template.....	69
Figure 29: Adding tier(s) to a blueprint template.....	69
Figure 30: Create a tier.....	70
Figure 31: Adding software to a tier	71
Figure 32: Selecting the menu to change the software attributes	71
Figure 33: Editing the software attributes	72
Figure 34: Launch a blueprint template	72
Figure 35: Blueprint instances.....	73

Figure 36: Manage tenant – select user	76
Figure 37: Modifications on a tenant	76
Figure 38: Modifications on a tenant II.....	76
Figure 39: Sample Maintenance Stakeholder Interaction (Developer initiated issue request on a sub-system issue)	84
Figure 40: Ticket flow of a reported issue	88
Figure 41: Management of maintenance procedures (most relevant cases of the management process)	99
Figure 42: Outline of a scheduled maintenance procedure affecting a single infrastructure node.....	101
Figure 43: Outline of scheduled federation-wide maintenance procedure affecting multiple infrastructure nodes	103
Figure 44: Outline of an unscheduled maintenance procedure affecting a single infrastructure node.	105
Figure 45: FIWARE Lab Level 1 support scenario for the case triggered by email from FI-developer	108
Figure 46: FIWARE Lab Level 2/3 support scenario, triggered by JIRA collector form.....	109
Figure 47: JIRA ticket flow example	111

LIST OF TABLES

Table 1: Routing table of WIT	19
Table 2: Current WIT node nova flavor-list.....	20
Table 3: WIT node utilization from 2014-07-02 to 2014-07-31	20
Table 4: Trento subnet list.....	21
Table 5: Trento node quota list	22
Table 6: List of tenants on the Trento node.....	22
Table 7: Berlin subnet list	23
Table 8: Berlin quota-defaults.....	24
Table 9: List of tenants on the Berlin node	24
Table 10: List of tenants on the Brittany node	25
Table 11: Seville/Malaga node quota list.....	29
Table 12: Overview of resource usage of the Spain (Seville/Malaga) node	30
Table 13: List of tenants on the Seville/Malaga node	30
Table 14: Berlin contact details.....	35
Table 15: Brittany contact details.....	35
Table 16: Spain contact details.....	36
Table 17: Trento contact details	36
Table 18: Waterford contact details	36
Table 19: IMINDS contact details	36
Table 20: ZHAW contact details.....	37
Table 21: PSNC contact details.....	37
Table 22: Neuropublic contact details.....	37
Table 23: CESNET contact details.....	38
Table 24: UPRC contact details	38
Table 25: Com4Innov contact details.....	38
Table 26: ACREO contact details	38
Table 27: GOWEX contact details.....	39
Table 28: WIGNER contact details.....	39
Table 29: UTH contact details	39
Table 30: BTH contact details.....	40
Table 31: Infrastructure support team	41
Table 32: OLA categories	45
Table 33: Workflow creation process	54
Table 34: Workflow registration process	55
Table 35: Separation of concerns between the major roles.....	55



Table 36: List of users with access to the glance-apache server in Spain.....	67
Table 37: Lannion usage list	77
Table 38: VM list	77
Table 39: eTOM framework references to the XIFI maintenance process	80
Table 40: Federation Maintenance Contact.....	81
Table 41: Infrastructure maintenance contact	82
Table 42: Stakeholder interaction	87
Table 43: Infrastructure Maintenance Escalation Levels	93
Table 44: Communication Infrastructure Maintenance Escalation Levels	94
Table 45: Components under Maintenance.....	96
Table 46: Sub-systems under Maintenance.....	98
Table 47: Responsibility Assignment	110
Table 48: Level 1 Helpdesk team.....	110



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
eTOM	e-business Telecoms Operation Map
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, 12 further nodes provided by the 12 new XIFI partners have been (or still are to be) added to the federation.

A first set of procedures and protocols for XIFI federation has been published as Deliverable D5.1 [5]. D5.1 has defined a first version 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, D5.1 also contains requirements and initial drafts of general procedures and procedures for developer support and maintenance.

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

The set-up and federation of the five initial nodes has allowed us to evaluate the procedures, protocols and tools defined in D5.1. The hands-on experience gained from that was very helpful to better understand technical, operational and legal requirements of a wide-area federated heterogeneous infrastructure. The insights gained have been feeding into the preparation of this document. While it provides an update to the procedures and protocols defined in D5.1, it contains significant enhancements e.g. regarding operations and support. The experience gained in the past months was also feeding into D5.4 which is being prepared in parallel to D5.3. D5.4 has a clear focus on procedures for new nodes joining the federation and as such D5.3 and D5.4 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.

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. 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 D5.4), 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).
- **Conclusions** are finally given in section 7.

Protocols and procedures have evolved since the beginning of the project and will likely continue to do so in the remainder of XIFI project, fed by experience gained from adding further (heterogeneous) nodes and from operating and maintaining a growing federation getting more complex. Also the growing number of users who rely on a secure and highly available platform and their needs in terms of support will provide new insight and a respective refinement of the procedures. Thus, the definitions and specifications contained in this document will be kept up-to-date on the XIFI Wiki as they evolve and serve there 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. 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.

The continuous roll out of the MD-VPN links across the federation denotes the steady growth of the cloud across Europe. Recently one of the new joining XIFI participants, the Prague node, has become active with the federation and is already offering resources to federation users.

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.

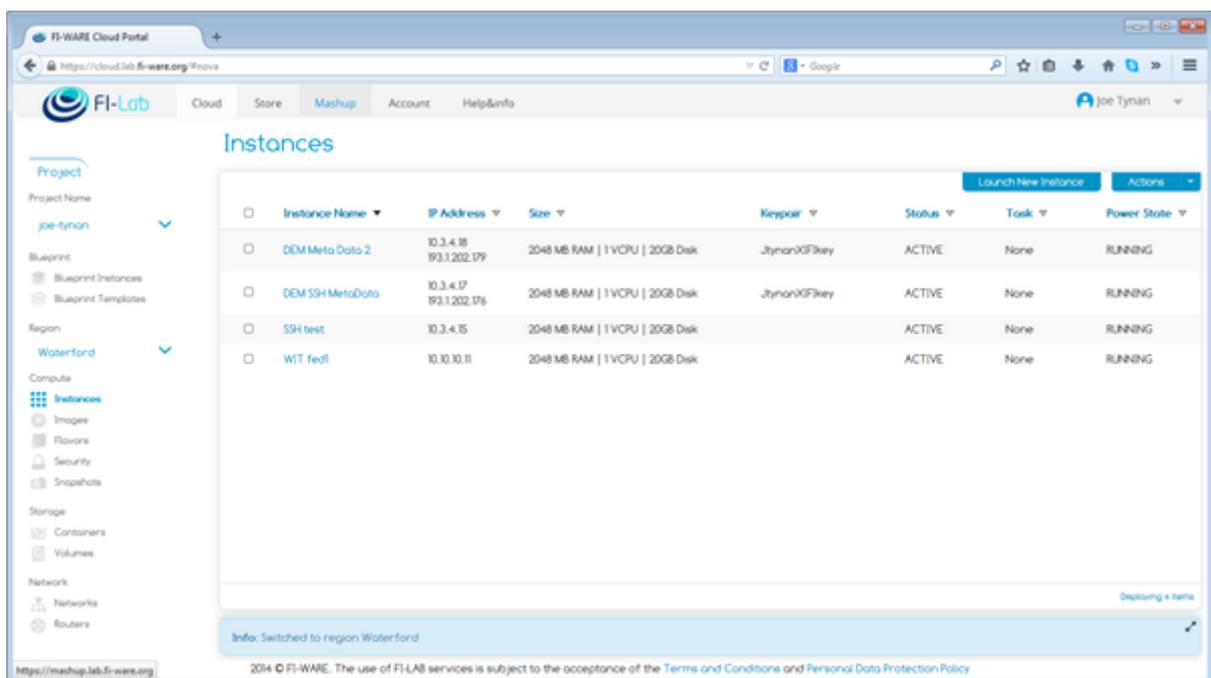


Figure 1: FIWARE Lab User Cloud portal

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. At the time of writing, the XIFI federation has 2789 federated users, 526 Organizations, 6 federated regions, 1712 CPU cores, 5154 GB RAM and 40607 GB of hard disk space with 997 operational Virtual Machines associated to the federation.



Figure 2: Operational capacity details

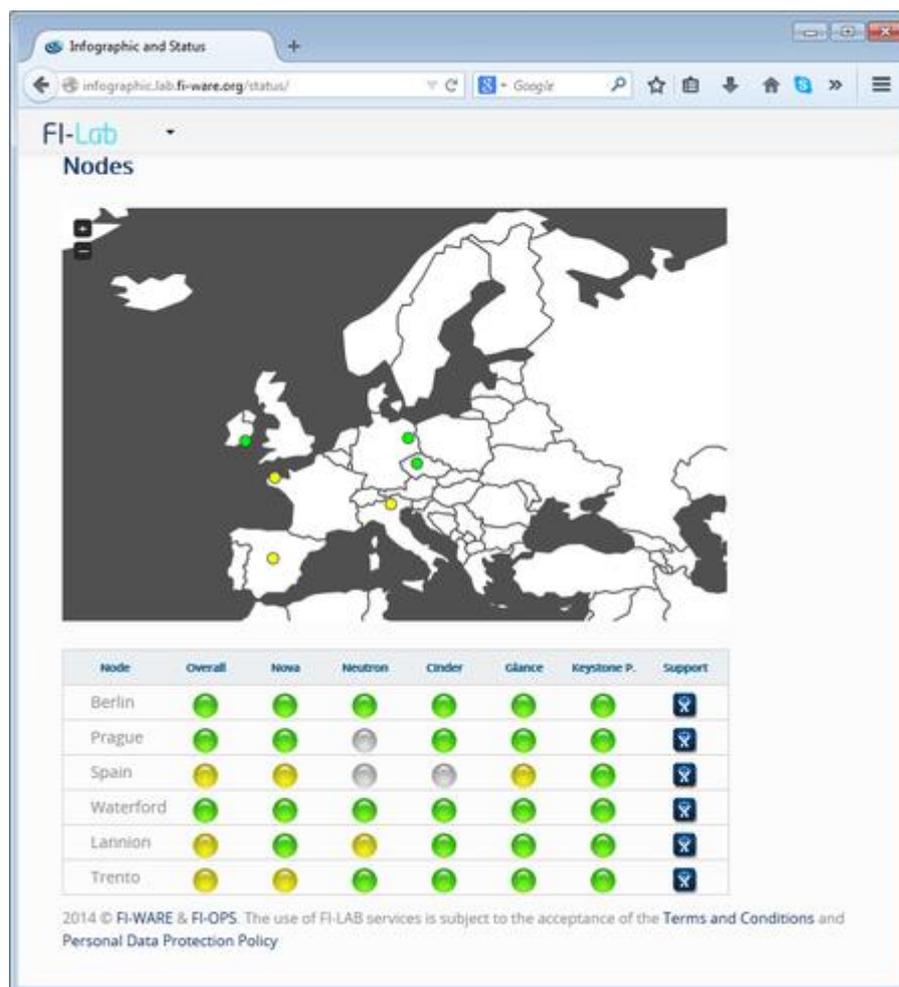


Figure 3: Regional cloud services status

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 need 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
```

Table 1: 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

2.3 Trento Node

The Trento Node has been deployed using FUEL 3.2.1 and manually configured as reported in D5.2. The cluster consists in 6 compute (192 cores), 3 controller (HA deployment), 3 Object Storage, 1 monitoring node and 1 server for FUEL Master node. The sixth node was added after the first deployment using FUEL 3.2.1 and this physical machine is located in another datacentre. The HA deployment has been tested with two controllers and actually restored on the third controller. As stated in Deliverable D5.2, the federation network has been configured in every physical machine. Currently the network of the federation is achieved through VPN connecting directly to Lannion.

quantum --os-region Trento subnet-list

id	name	cidr	allocation_pools
80c95b85-c249-4ec4-ae73-a601cec3f707	ext_net	193.205.211.64/27	{ "start": "193.205.211.70", "end": "193.205.211.93" }
9ec19948-1b03-498e-b511-9c8253976d27	int_net	192.168.111.0/24	{ "start": "192.168.111.1", "end": "192.168.111.253" }
ed50446d-bc0d-4103-8403-9ff5d3e77e89	vpn-int-net	10.0.32.0/22	{ "start": "10.0.32.21", "end": "10.0.35.253" }

Table 4: Trento subnet list

The first installation of the Keystone has been done on the monitoring node 193.205.211.69 and after a period of test the keystone proxy was moved to Spain. In the monitoring node there are installed the following monitoring software/script:

- Nagios
- NGSI Adapter
- Context Broker
- Federation Monitoring API
- NAM
- Security Dashboard

Nagios has been modified to check also the physical machine and set several alerts.

Currently Trento node hosts the XIFI UC5 (Quality of Experience in NaaS), UC7 (Monitoring QoS in the Node), UC10 (Security monitoring), UC11 (GE monitoring).

All the federation images are available on the Trento node. A test node has been set up to test sub-systems before the installation on the production node. The ETICS tool has been developed in the Trento test node and this and the other activities of development was supported through direct mail and phone. The same support was given to the developers for the installation and configuration of the monitoring tools on the monitoring node. Currently the support is provided through the "fiware-lab-help@lists.fi-ware.eu" mailing list until the Jira help-desk will be fully operational. The activity of support regards all issues about cloud portal, creation of vm, creation of project, customization of quotas, allocation of floating IP, etc.

The default quota has been configured as follows:

Property	Value
metadata_items	1024
injected_file_content_bytes	10240
ram	2048
floating_ips	1
key_pairs	100
instances	3
security_group_rules	20
injected_files	5
cores	6
fixed_ips	-1
injected_file_path_bytes	255

```
| security_groups | 10 |
+-----+-----+
```

Table 5: Trento node quota list

The utilization of the node is described in the following table.

Tenant ID	Instances	RAM MB-Hours	CPU Hours	Disk GB-Hours
0009	1	344064.86	672.00	0.00
0038	1	329.39	0.64	16.08
0053	3	20520.68	20.04	200.40
0081	10	467763.96	845.09	685.14
000135	1	344064.86	672.00	0.00
000140	1	344064.86	672.00	0.00
000347	3	6978.84	6.82	146.63
000442	7	3635570.74	3550.36	44374.89
000841	8	1034439.53	1584.64	10893.82
000852	1	1376259.46	1344.00	20160.05
000853	1	1376259.46	1344.00	20160.05
000984	1	32.43	0.06	0.00
002682	2	688129.73	1344.00	0.00
002782	22	15293864.31	15120.60	263238.97
002785	1	833739.16	814.20	20354.96
002798	4	1452521.52	1466.86	20234.53
002862	1	16515113.52	5376.01	67200.17
003014	1	7.11	0.01	0.00
003146	1	344064.86	672.00	0.00
003191	1	710.26	1.39	0.00
003273	1	175062.76	341.92	0.00
003373	5	1141880.01	1245.09	16835.67
003437	1	688129.73	672.00	6720.02
003449	4	1527.89	2.98	0.00
003954	14	9218983.16	4501.46	135043.70
004013	3	582709.18	1138.10	28452.60
004047	2	380396.07	371.48	9287.01
004242	1	168042.37	164.10	1641.04
004287	5	14237.58	27.81	0.00
dbb1d0ef27704663b2336e60cafbb8db	2	5505037.84	2688.01	53760.14

Table 6: List of tenants on the Trento node

Future work is focused on upgrading the Master Node to the last version of ITBOX maintaining the environments created. Another important step is the deployment of the MD-VPN because actually Trento is connected to the federation with a VPN through Lannion Node. Trento node is still waiting for the activation of the MD-VPN from the Italian NREN. The activation has been delayed due to a technical problem. The most critical future work is the OpenStack version upgrade, from Grizzly to IceHouse.

2.4 Berlin Node

A dedicated infrastructure has been set up for the Berlin node as documented in the scope of D5.2 using an ITBox assisted set-up with manual modifications. The datacentre currently consists of a controller node (nova, glance, quantum, horizon, keystone), a swift storage node, a monitoring node, and 3 compute nodes (72 cores). Storage is implemented on top of NFS by a fully redundant NetApp metro cluster (6 TB).

The 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 connect through fibre (1 Gbit/s) and utilize a dedicated VPN tunnel for L3 connectivity through this fibre. Both sites connect to the MD-VPN on distinct addresses each in the 10.0.16.0/20 range, while the Fraunhofer site provides public IPv4 addresses in the 193.175.132.32/27 range. The addresses 193.175.132.41 to 193.175.132.62 are currently available for the OpenStack floating IP pool. More addresses could be made available as soon as the XIFI federation management network has been moved to the MD-VPN since currently all exposed hosts have both an MD-VPN and a public address.

quantum --os-region Berlin subnet-list

id	name	cidr	allocation_pools
74d81e30-bdd2-4792-9763-a58578dd8dc6	ext_net_public	193.175.132.32/27	{"start": "193.175.132.41", "end": "193.175.132.62"}
b290ca77-129c-4d0e-a76b-db9814bfc1ca	ext_net_federation	10.0.16.0/24	{"start": "10.0.16.130", "end": "10.0.16.254"}

Table 7: Berlin subnet list

Monitoring of DT and Fraunhofer site takes place through a shared context broker that provides both data centre monitoring data and wireless testbed monitoring data through the same API towards the federated monitoring sub-system. Both sites internally utilize their own dedicated Nagios infrastructures for internal monitoring and export data needed for federation monitoring.

A dedicated testbed has been set up with minimum configuration which is used for testing and reference purposes. It is currently used to test the OpenStack release migration from Grizzly (current configuration for the XIFI federation) and Icehouse (candidate for the next upgrade).

The Berlin node has implemented all required GEs and currently operates under "close to production" conditions, hosting with a variety of users ranging in their skills from beginners to experts. Support is provided through the "fiware-lab-help@lists.fi-ware.eu" mailing list until the Jira help-desk will be fully operational. The Berlin node currently hosts mainly the XIFI UC1 (E-Health in a Smart City Environment) and developments of the FISTAR and FI-CONTENT UC projects (a number of other UC project participants may be also active on the Berlin node but are not yet recognized).

First experiences have been made regarding operations and maintenance of the node and a number of issues have been identified. Most urgently the following issues need to be addressed:

- Security issues. There are issues evolving regarding the identity of users that utilize the Berlin node infrastructure without being approved by the Berlin node administration, since identity management is not under control of individual nodes yet. Besides security issues this might cause privacy and legal issues as well.
- Management of floating IPs. Since floating IPs are very scarce resource they have to be used dynamically (i.e. with frequent allocation and release by a tenant). For each allocation access control currently must be maintained manually, in particular regarding the management of access control lists on the Fraunhofer FOKUS edge router (firewall rules are bound to IP addresses). This is time-consuming and any flaws in managing the correlation of OpenStack security rules, edge router access control lists and floating IP address may result in severe security issues.
- Configuration issues and OpenStack issues. Currently, configuration updates and bug fixes create a continuing issue since they often must be applied on short notice with minimum impact on the production environment. This so far has caused several server reboots (rarely) and service restarts (quite often). Due to this it was not yet possible to establish and maintain a more regular maintenance schedule as planned.

Quotas have been applied in consequence of the public IP shortage. The Berlin node currently applies the following limits for regular, individual tenants:

- We used our local Nagios to internally monitor OpenStack services and functionalities
- Our C6220 servers include IPMI
- Node federation supervision:
 - 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. There is 2 VMs used for the federation supervision: One VM for CB and NGSI and one VM for DEM, NAM and NPM
- Node federation:
 - Keystone Proxy: 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
 - Second external network: For the need of the federation, we needed to configure two external networks: 1 network for the MD-VPN and one Public network.
- Quotas:
 - Due to our limitation on providing Public IPs, we decided to allow only 1 Public IP per tenant. We also decided to restrain the global size of disk to 50GB by default for a tenant. This default configuration can be modified for a specific tenant in case needed.
- Maintenance tasks:
 - We fixed some bugs related to HA
 - We had to recover (from scratch) one of the 2 controllers after a disk failure
- Creation of local catalogue:
 - We uploaded on our glance repository a list of cloud images that were needed for XIFI.

The usage of the Lannion node (from 2014-07-09 to 2014-08-07) is shown below.

Tenant ID	Instances	RAM MB-Hours	CPU Hours	Disk GB-Hours
00000000000000000000000000000009	1	15.79	0.03	0.00
00000000000000000000000000000356	2	5505026.87	2688.00	53760.03
000000000000000000000000000002559	1	1376256.72	672.00	13440.01
000000000000000000000000000002983	11	6920713.87	4937.39	57197.53
000000000000000000000000000002988	3	1961143.00	957.59	19151.79
000000000000000000000000000003437	5	11010053.75	5376.00	107520.06
000000000000000000000000000003449	12	35234.28	65.85	19.77
000000000000000000000000000003478	3	24772620.94	8064.00	161280.08
000000000000000000000000000003847	1	11010053.75	5376.00	107520.06
000000000000000000000000000003851	3	329177.43	642.36	3.79
000000000000000000000000000003965	4	109132.66	87.69	836.37
000000000000000000000000000003997	20	4456176.22	3036.19	0.00
000000000000000000000000000004004	3	3872950.10	1894.69	45644.61
000000000000000000000000000004012	1	344064.18	672.00	0.00
000000000000000000000000000004019	1	344064.18	672.00	0.00
000000000000000000000000000004098	1	2544645.50	1242.50	24850.05
000000000000000000000000000004287	2	30.86	0.06	0.00
000000000000000000000000000004291	1	659629.89	322.08	6441.70
38aec686f107485ebc1ca9763d96d958	5	6881283.59	3360.00	67200.04

Table 10: List of tenants on the Brittany node

UC deployment

The Lannion node hosts some Use Cases and participates actively in their deployment. To do so, we created a specific tenant for each of them, with their own private subnet, private rules and specific flavour to fit the specific needs of each Use Case.

- UC Fi-Content "Smart City Guides": 3 VMs has been deployed with one connected with a Public IP
- Fi-Content "Connected TV": This Use Case is still under deployment, but basically it consist of 1 VM with 1 Public IP
- XIFI UC12 "From Fire to the Fi-PPP":3 VMs has been deployed for this Use Case and each of them needed to have a Public IP
- XIFI UC10 "Security Monitoring": the need was to install one VM with access to the MD-VPN. So we placed the VM in the Imaginlab Tenant with the VMs used for the federation monitoring

2.6 Spain node (Seville/Malaga)

Architecture

The Spain node was deployed manually using the Essex version of OpenStack, at the time of the installation of it (beginning of FI-WARE project) no FIWARE Ops tool was available. This version of OpenStack include the different versions of DCRM, the Cloud Portal and the IdM – Keyrock and it was evolved during the last year in order to be the Master node of the federation using the PaaS and SDC like central service of all XIFI nodes. Initially the Spain node was compound of the following equipment:

- 28 servers with a capacity of 2 CPUs, 4 cores per CPU, 16Gb RAM and 150Gb HDD for internal storage, in which we deploy the different instances of the nova-compute.
- 12 servers with a capacity of 2 CPUs, 6 cores per CPU, 128 Gb RAM and 2 uplinks of 10 Gb Ethernet with 600 Gb of SAS storage used to put there the nova-controller, nova-network, and glance.
- External storage with a capacity of 10 Tb in NFS and dual network interface.
- Physical Firewall to connect to Internet.
- Internal Router to connect the nova-compute instances to the network.
- 10 Gbps interface

An extension of infrastructure was developed during this year in order to extend the capacity and starting the migration of the OpenStack version, the new hardware added is the following:

- 16 Servers with 2 CPUs, 8 Cores/CPU, 128 Gb RAM, 2 uplinks 10 Gb, 600 Gb SAS, 6 Tb DAS SAS and 2 FC 8 Gb used to install nova-compute services
- 32xServers with 2 CPUs, 8 Cores/CPU, 128 Gb RAM, 2 uplinks 10 Gb, 2 Tb MDL-SAS and 2 FC 8 Gb used to install nova-compute instances
- 4xServers with 2 CPUs, 8 Cores/CPU, 128 Gb RAM, 2 uplinks 10 Gb, 600 Gb SAS, GPU and 2 FC 8 Gb used specifically for GEs applications that required specific computational characteristics.
- 2 Cabins with 20 Tb SAS, 80 Tb NL-SAS, 24 uplinks, 2 Controllers, 2 File Servers and 20 Gb Eth

Together with this configuration, a replication of optic fibre was developed in order to provide a redundant link to the Spain node.

Experience on support and maintenance

During this period of time lots of maintenance and support activities have to be done due to the huge amount of resources used during this period of time. Below bullets summarise those activities:

- Hardware maintenance: Replacement of the defective or failure hardware. Mainly, power supplies or hard disks.
- System Operations: Upgrade and maintenance.
- Security: Checking log security, and verifying the security problems and notifications of illegal ones.
- Network maintenance: Because Spain node is a distributed node, there are some maintenance tasks over the optical channels that connect the physical data centres.
- FIWARE Lab Migration: Migration of the whole Spanish node from previous physical hosts in Santander DC to Seville DC. Including the addition of the NOVNC proxy (which wasn't previously configured on that node).
- Increase the number of available Floating IPs: Some bunches of public IPs were added to Spain node of FIWARE Lab in order to complete 2 class-C networks (130.206.82.0/23) in the Spanish node.
- Provision / Unprovision of physical hosts: Depending on events / availability of physical hosts we added some hosts or removed them migrating VMs
- Manual handling of used/unused VMs: Removing unused VMs when space was needed, removing the corresponding floating IPs
- Adjustments in Databases: There are known issues with Essex version of OpenStack which requires manual maintenance tasks on databases and/or in libvirt.
- Provision of KVM Virtual Host for Access Control Component: Provided a Virtual Host (not controlled by OpenStack) for Access Control Component and its DNS name az.lab.fiware.org
- Solving problems: Sometimes, unexpected problems may happen, for example, the nova-controller host runs out of disk space and some services dies, the keystone proxy doesn't respond to queries anymore, sometimes something goes slower... and these problems must be solved at once.
- Intrusion and abuse detection: Sometimes, some virtual hosts user have been compromised and the intrusion has had to be detected, the VM owner must be warned and the firewalls must be updated manually in order to prevent further problems. Different actions should be taken depending on each case.
- Users support: This tasks include the support to developers in the test node, changing quotas for projects, support on issues regarding the cloud portal (creation of VM, access to VMs, project authorization etc.), installations and configurations of the monitoring tools.
- Software updates have been applied to the Cloud Portal. These updates include bug fixing, new features, and improvement of service response times.
- Configuration files to adapt to changes in the architecture. These changes include new addition of nodes in FIWARE Lab.
- Cloud Portal service was rebooted several times due to software updates, service malfunctioning and changes in the configuration.
- Cloud Portal server was rebooted a few times due to infrastructure maintenance tasks.
- Several updates in the Security Proxy. These updates include bug fixing, new features, and improvement of service response times.
- Configuration files were changed due to the addition of new nodes to the environment.
- Configuration files were changed due to changes in credentials to access Cloud services.

- A database was created to temporally store authentication OAuth tokens.
- Security Proxy service was rebooted several times due to software updates, service malfunctioning and changes in the configuration.
- Security Proxy server was rebooted a few times due to infrastructure maintenance tasks.
- Software updates have been applied to the IDM. These updates include bug fixing and new features.
- Configuration files are changed every month due to security requirements: changing user password to access the local database.
- Database and asset backups have been created recently to allow the accomplishment of Security Law in Spain (LOPD) are created every day.
- IDM service was rebooted several times due to software updates.
- IDM service is rebooted every month to apply changes in the configuration.
- IDM server was rebooted a few times due to infrastructure maintenance tasks.

OpenStack configuration

Currently, we are working without HA in the Spain node. We are working in the migration of the old compute-node to the new infrastructure and then we start to configure the new version of OpenStack (IceHouse) which will support HA in the nova-controller, database, cloud portal and IdM – Keyrock. We are working over the Ubuntu 12.04 LTS version and the virtualization solution adopted is KVM.

A summary of the situation is shown below:

- Node local administration and supervision:
We are using a VM instance in which we have deployed a Nagios program in order to monitor the infrastructure and important GEs that we are using in the Spain Node. It helps us to obtain alarms when some of the core services are down for any reason.
- Node federation supervision
In order to monitor the infrastructure, the different components (NAM, DEM and NPM) were deployed in order to recover information of the Spain node and progress them to the federation.
- Network addresses.
The currently available addresses are:
 - Fixed IP range: 10.0.0.0/21
 - Floating IP range: 130.206.82.0/24 (with some exceptions and IPs got for maintenance or operations)
- Quotas
Due to the high demand of resource, we had to limit them accordingly to the quotas shown in the following table.

Quota	Limit
metadata_items	128
volumes	10
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

Table 11: Seville/Malaga node quota list

- **Global services**
We have to develop specifically the PaaS Manager, SDC Manager, Cloud Portal and IdM Keyrock in order to be used by all the nodes of the federation. In the same way, activities were done in order to offer all these components in HA which will be finished with the migration to the new infrastructure.
- **Global Catalogue**
Related to the list of images and GEs, the Spain node is the reference for the rest of the XIFI nodes. Any change related to new images or new version of GEs has to be progresses to the rest of the IO. Some activities were developed in order to provide a server to access the different images available there. Currently we are defining a protocol with the rest of IO in order to automatically publish any change related to the image and/or GE.

Current Status

The node in Spain has implemented all GEis that are authorised by the FI-Ware project and currently operate under production conditions, hosting a variety of users from different countries, ranging in their skills from beginners to experts. Figure 4 highlights the different countries from where FIWARE Lab is used. The sole means of requesting support so far has been the sending of an email to the "fiware-lab-help@lists.fi-ware.eu" mailing list which was distributed to administrators of all nodes. The implementation of a JIRA ticketing system will replace this interim solution. However, the option of sending emails to a support email address will be maintained even after the introduction of JIRA. This is mainly in order to provide also the email contact option for user convenience, although use of JIRA will be encouraged. In case of GE-related issues we are using the Stackoverflow portal [17] to publish information about the resolution of incidences related to these components.

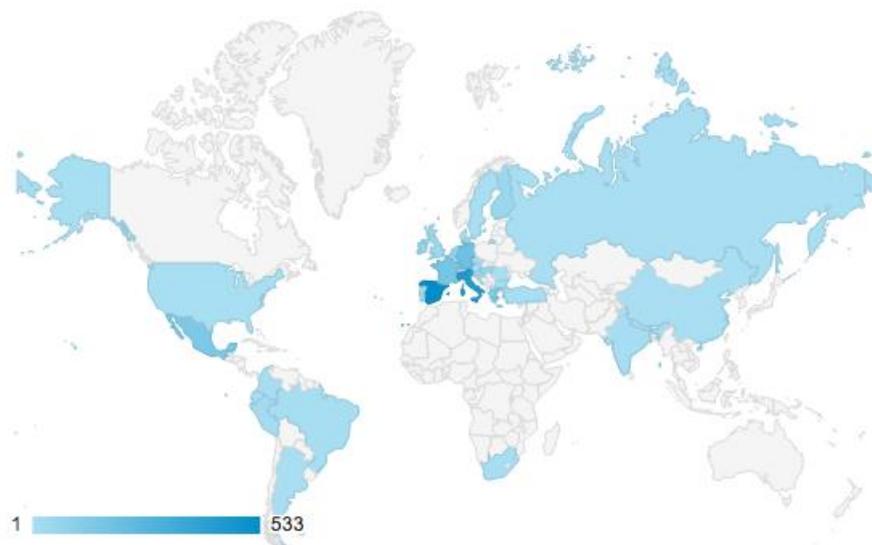


Figure 4: Distribution of users accessing FIWARE Lab, per country (as of July 2014)

including NGSI Adaptor or Context Broker (0.14.1) are also present. The system is not based on ITBox solution. All components has been installed and configured manually to ensure optimal system performance, stability and reliability.

At the time of writing, CZIFI node is linked to the XIFI OpenStack components services like Keystone identity service provider, Glance and other monitoring and node state reporting systems (<http://infographic.lab.fi-ware.org/status/>). System is also connected to the FIWARE Lab and users are able to select Prague node at the menu when making decision where to create VM.

CESNET has also been chosen to test MD-VPN service provided by GÉANT and uses assigned 10.0.96.0/20 subnet.

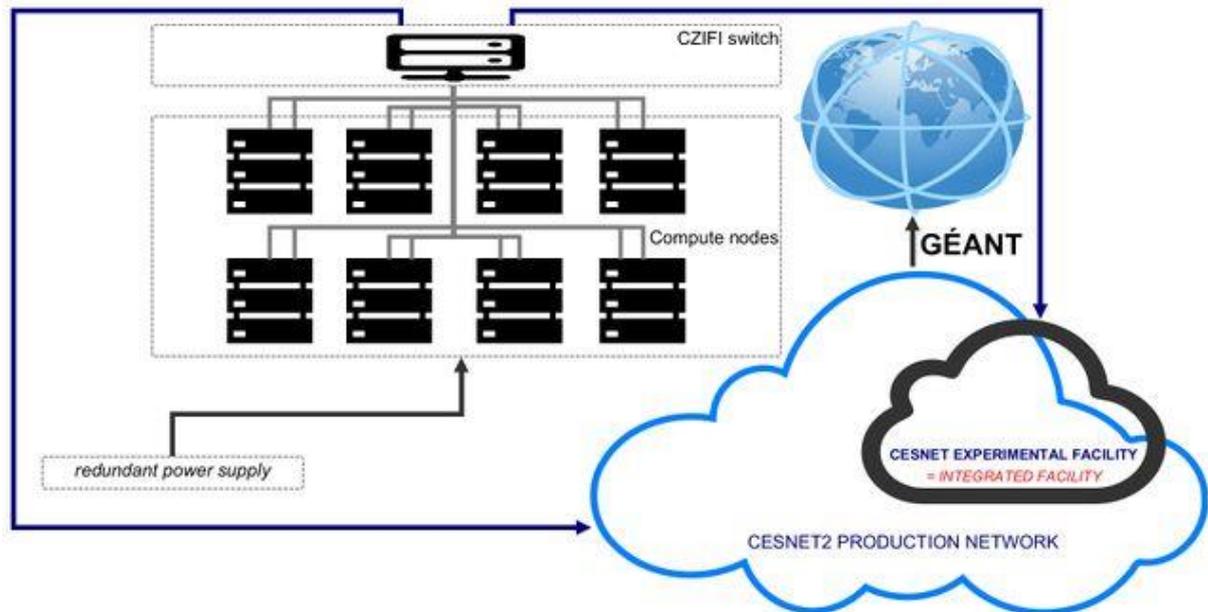


Figure 5: Graphical Scheme of the Prague node

- Maintenance

So far, system is stable and working. However, it still has not been fully utilized and to prove its stability under full production load. The hardware configuration of the node is planned to be upgraded in near future, where the main upgrade covers RAM upgrade and SSDs for Cinder volumes (connect to IceHouse release testing).

The setup uses collect for monitoring essential system components. CZIFI is also linked to CESNET Nagios network monitoring.

Areas to be addressed are similar to other nodes cover following issues:

- Floating IP addresses management.
- Keystone Proxy, a.k.a. Identity Federation.
- Permanent 1st level support solution for the federation.

- Assistance

There was no need for the user assistance yet during CZIFI resources utilization. Service that helps and assists users also from other nodes in the federation is the RT system.

- Several Facts

- CESNET node (CZIFI) is currently running and operating all components, except local monitoring (hosted temporary in Trento) and Neutron.

- CZIFI has own RT for tracking users requests, problems and suggestions as well as serves as 1st level support for things related to CZIFI node.
- Temporary serves also for the federation as a main tracking system.
- CZIFI has up and running VPN-Proxy (aka. MD-VPN temporary solution).

Tenant ID	Instances	RAM MB-Hours	CPU Hours	Disk GB-Hours
00000000000000000000000000000049	1	14.55	0.03	0.00
00000000000000000000000000000081	1	344064.02	672.00	0.00
00000000000000000000000000000140	1	344064.02	672.00	0.00
000000000000000000000000000000356	1	2752512.15	1344.00	26880.00
0000000000000000000000000000003576	2	688128.04	1344.00	0.00
0000000000000000000000000000003697	1	344064.02	672.00	0.00
0000000000000000000000000000003732	1	344064.02	672.00	0.00
0000000000000000000000000000003817	1	2752512.15	1344.00	26880.00
0000000000000000000000000000004116	1	344064.02	672.00	0.00
0000000000000000000000000000004291	1	11010048.60	5376.00	107520.01
0000000000000000000000000000004409	7	24156654.35	4794.50	55052.26
0000000000000000000000000000004412	114	8155246.43	15486.99	2941.50
000000000000000000000000000000CESNET	7	2408448.13	4704.00	0.00
000000000000000000000000000000services	1	1376256.07	672.00	13440.00
30373ff0b1ef47c4ab8276edde7fe325	1	344064.02	672.00	0.00

Figure 6: Usage in August 2014

2.8 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 or inefficiencies have been encountered.

So far, requests for support could be sent to a mailing list with administrators and other experts receiving and answering those questions and taking the required corrective actions.

A number of questions have been received that 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.
- Requirements from users
 - Increasing Quotas (floating IPs, disk, number of instances, etc...).
 - Names in DNS.
- Questions about different Generic Enablers
 - Usually redirected to the StackOverflow.
 - Tags created at the moment in StackOverflow (fiware, fiware-orion, fiware-wirecloud, filab).
 - Rest of GEi redirected to the owner.
- 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...
- Bugs and feedback.

Currently we implement a feature that enables the traceability of the emails that we receive in order to know when and how we resolved a question, and to follow the steps to the resolution of an incident. We are defining a procedure to use Jira as tracker system and try to resolve the issues that we have identified. Details are described in section 6 Support to FI-Developers.

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 [9]: [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 14: 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 15: 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

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

Table 16: 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 17: 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 18: Waterford contact details

3.1.6 IMINDS

Role	Contact	Availability
Node Manager	ThijsWalcarus	Mon-Fri 9h-17h CET
Authoritative contact	ThijsWalcarus	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 19: 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 20: 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 21: 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 22: 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

System and Network Administrator	Jan Kundrat	24/7
Node Help Desk	Local XIFI Support Team	24/7

Table 23: 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 24: UPRC contact details

3.1.12 Com4Innov

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

Table 25: 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	24/7

Table 26: ACREO contact details

3.1.14 GOWEX

Role	Contact	Availability
Node Manager	Luis M. Calvo	Mon-Fri, 09h-17h CET
Authoritative contact	Jenaro García	Mon-Fri, 09h-17h CET
System and Network Administrator	Miguel Egido	Mon-Fri, 09h-17h CET
Node Help Desk	Miguel Egido	Mon-Fri, 09h-17h CET

Table 27: GOWEX contact details

3.1.15 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 28: WIGNER contact details

3.1.16 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 29: UTH contact details

3.1.17 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	Patrik Arlos	Mon-Fri, 09h-17h CET

Administrator		
Node Help Desk	Switch Board	Mon-Fri, 09h-17h CET

Table 30: BTH 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 6.5.

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 31: 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 [15]. 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 are described in subsequent sections.

4.2 Update on Support Process and Procedures for joining the federation

The Federation Office, as defined in [8], 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 D5.4 [6].

As an update of D5.4, this paragraph shows an example scenario that is related to the interaction of a new node and XIFI – specifically FIWARE Ops Helpdesk and task5.5 coach – in the Methodological support process defined in D5.4.

In Figure 7 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.

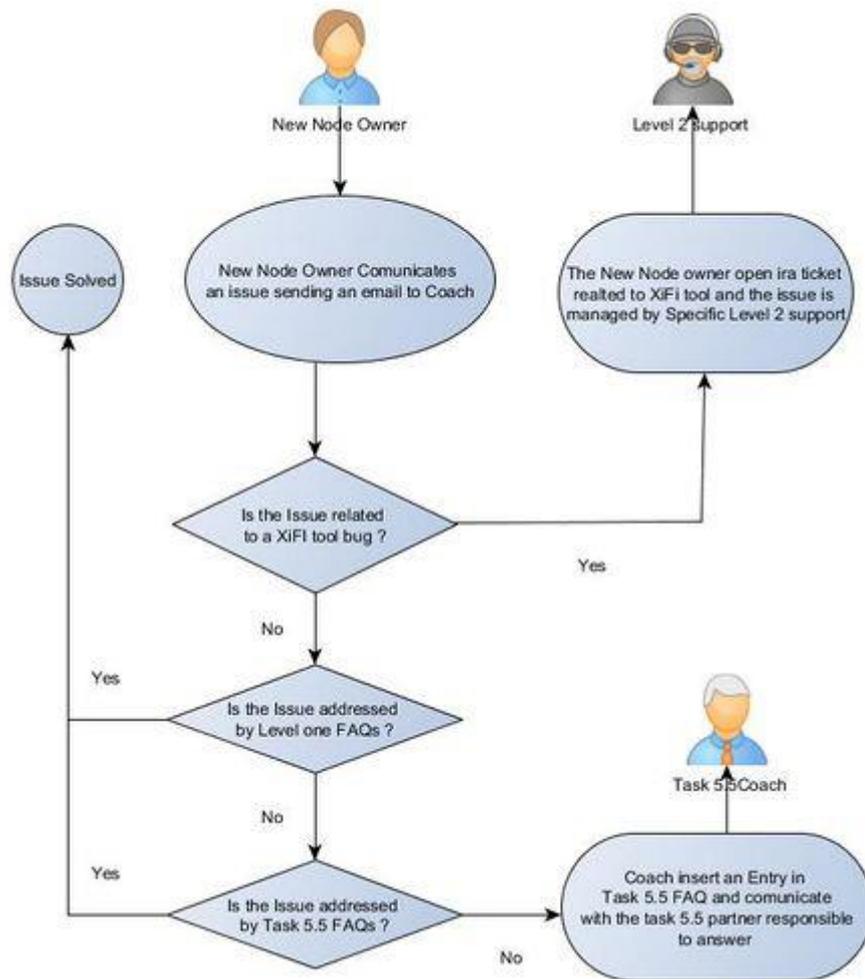


Figure 7: Federation support procedures

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
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 32: 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 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 services³ 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 EuroiGrid).

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).

³ In OLA all resources contribute to externally visible services, hence in workflow language we could talk only about services.

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.

4.4.1 OLA Level 2 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 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 FI-Ware 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 [8] 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 8, taken from [8].

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.

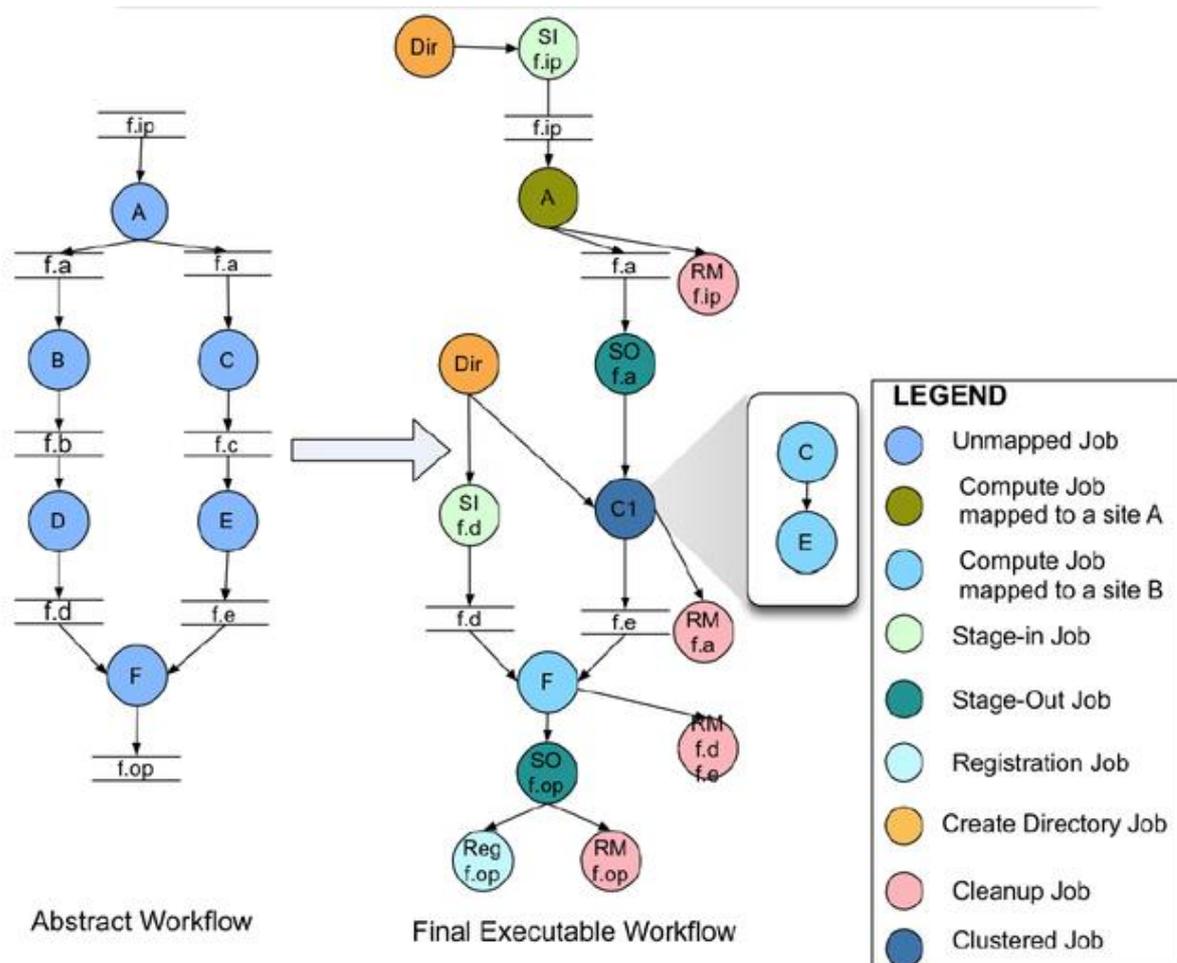


Figure 8: Pegasus WF mapping

The second BCP to mention is a workflow platform at myexperiment.org [11], 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 [11] 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.

4.4.2 Workflows for cross-layer optimisation

The need for a cross-layer optimisation was well understood long ago: an example is Random Early Detection (RED)[13]- 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)[12]. 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 valuate [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 9, which basically exhibits a multi-source (utility) policy exchange.

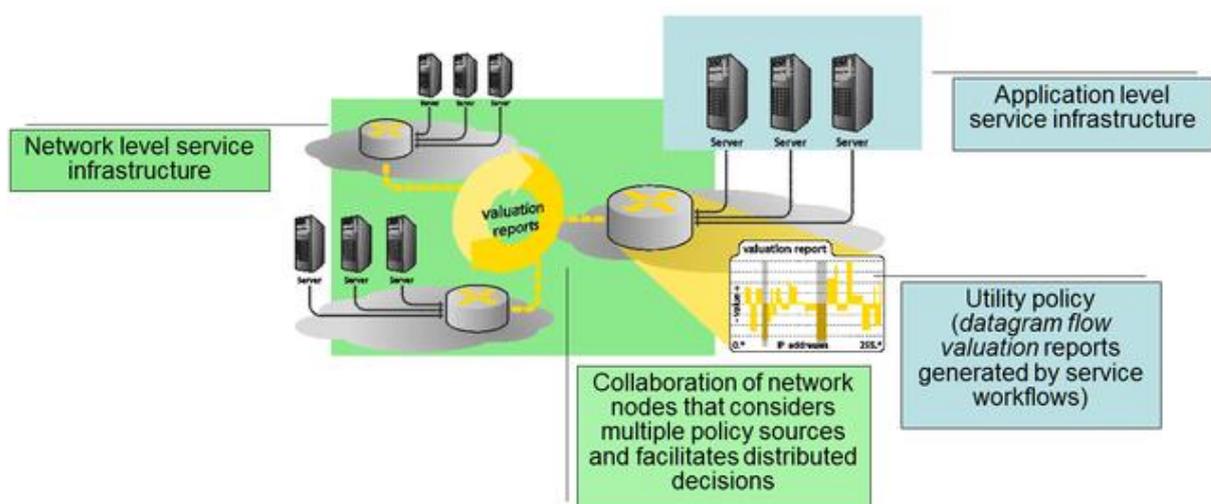


Figure 9: Cross-layer optimisation in D-CAF

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 policies⁴?

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.

4.4.3 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 federations. Second, being based on a structured set of roles⁵ it automates permission propagation schemes as shown below in Figure 10. However RBAC is also known to have conflicts mainly due to multiple permissions inheritance and to multiple dynamic roles assignments also shown below in Figure 10. 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 3 will be considered as two different workflows.

⁴ 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.

⁵ Reflecting the structure of an organisational hierarchy, or reflecting the structure of relations between stakeholders, otherwise.

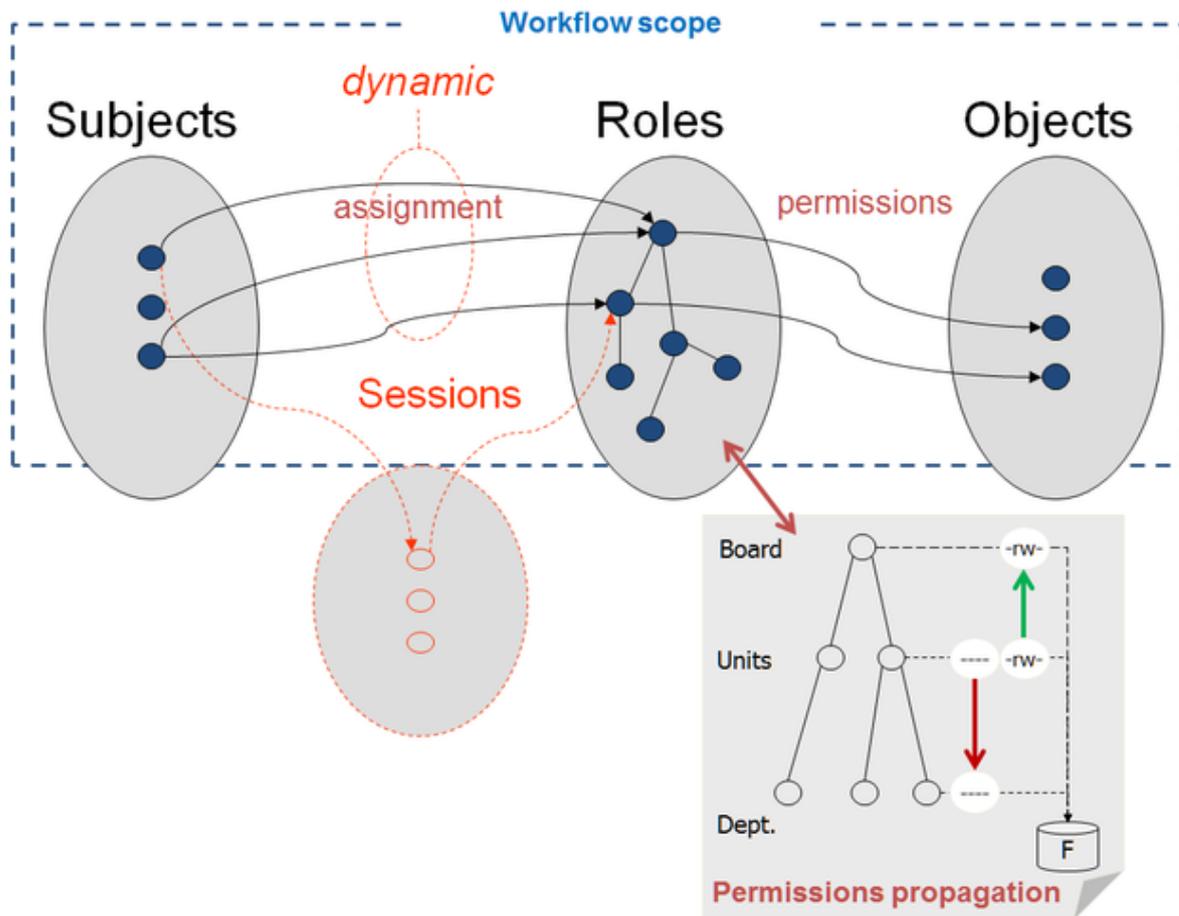


Figure 10: The scope of WAC

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.

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.

4.4.4 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 need 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, [16] 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 11 demonstrates (in a form of a concept map) the relations of the stakeholders to a workflow are not symmetric.

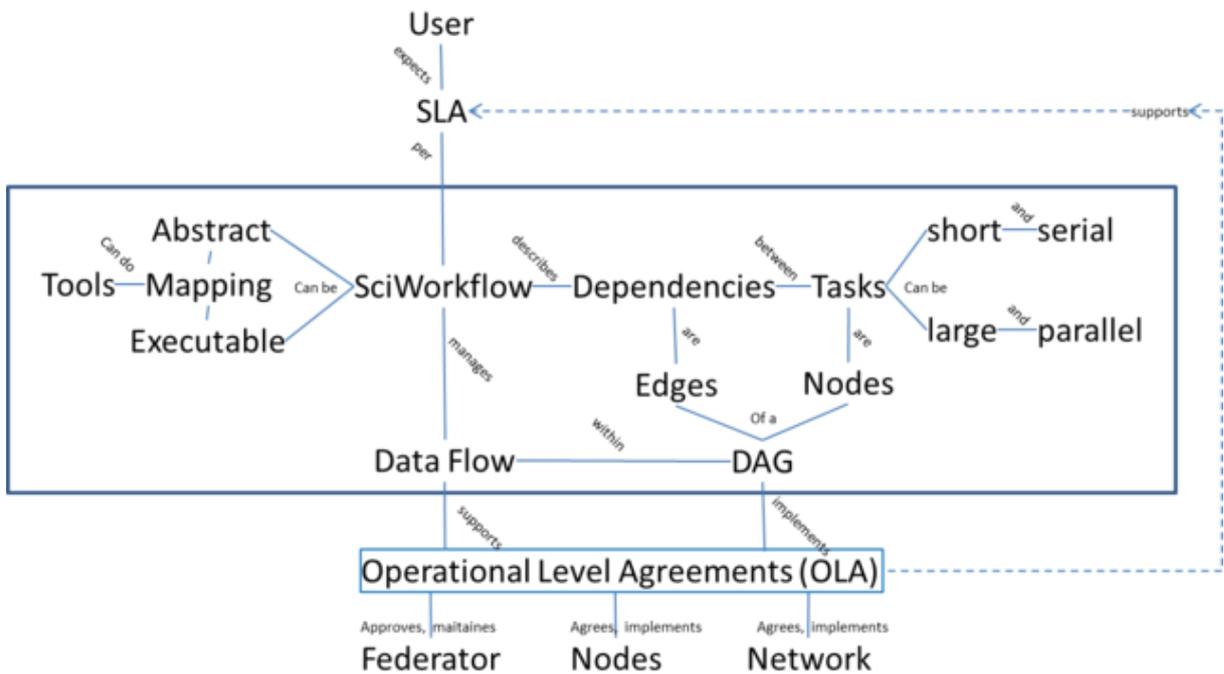


Figure 11: Workflow as a Unit of Trust in OLA

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 12.

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 5 Sample workflow

Figure 12: 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 33.

Role\Phase	Requirements	Design	Debugging	Testing
User	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 33: Workflow 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 34.

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	Maintain sufficient connectivity between active nodes and

			checked	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 34: Workflow registration process

It is obvious that Table 34 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 35: Separation of concerns between the major roles

As Table 35 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 35 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.

4.4.5 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 (4.4.4), 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 importance⁶ 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.

4.5 OLA implementation in XIFI

4.5.1 Introduction

XIFI foresees the setting up of SLAs between the XIFI federation and its customers, i.e. the developers using the platform. Ensuring Service Levels is seen as an important characteristic in order

⁶ The importance may differ for the two cases: i) workflows as part of OLA, ii) Workflows without OLA.

to provide reliable and stable services that are attractive to users of XIFI. However in order to fulfil the SLAs it requires coordinated activities between the federation members.

OLAs (operational level agreements) are considered as a means to ensure the coordination that is required inside an organization to achieve the SLAs. The correct delivery of services involves various groups; their activities need alignment, responsibilities must have been clarified, and all involved members must be aware of their obligations and the timeframes in which services have to be delivered. OLAs transfer the SLA concept into an organization. Even if it is not a legally binding contract as it is the case between customer and service provider, those agreements use the same structure and define similar content.

OLAs help to formalize internal processes and objectify their execution. This is achieved by establishing agreement on responsibilities and fixing the terms beforehand. Steps undertaken in resolving issues are documented and hence, the performance in service delivery can be evaluated against pre-specified KPIs. Through such performance data, problems and issues can be detected early on and solutions can be instigated for a continuous optimization and improvement process. Making suitable excerpts from performance values visible to the customers or to the general public can demonstrate the capabilities and help to build confidence and trust in the services.

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.

In this section the OLA concept is mapped onto XIFI, specifically looking at the procedures that involve infrastructure operators. The various activities are grouped and categorized into a few domains that form the basis for potential OLAs. First we present a rough categorisation of the procedures. Depending on the area, it involves different stakeholders. A general scheme is derived from ITIL SLA / OLA template that allows analysing the identified areas in terms of future OLAs. For those areas we discuss various aspects and identify requirements. It provides a basis on which future concrete OLAs for XIFI infrastructure operators can be constructed.

4.5.2 OLAs for XIFI Infrastructure Operator

Infrastructure Operators in XIFI perform complex interactions among each other and with other stakeholders in the federation. For the discussion of OLAs for infrastructure operators it is helpful to categorize their activities into logical domains which in turn determine who the stakeholders are that need to interact and coordinate in the provision of the respective services. For each domain or sub-domain specific OLAs can be defined that specify the required service level for this group of services.

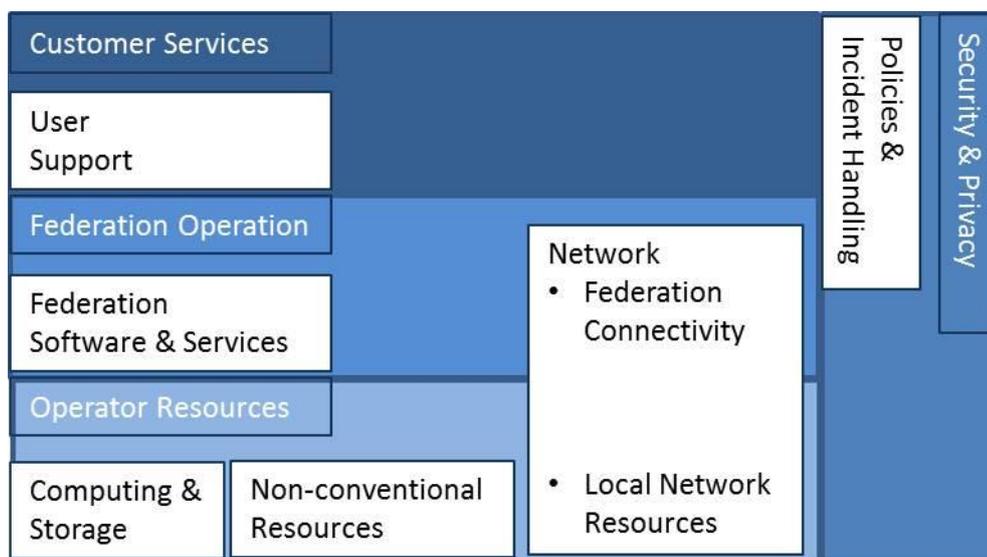


Figure 13: OLA Categories in XIFI

Figure 13 shows the decomposition of the services supported by XIFI infrastructure operators. Logically one can distinguish three layers:

At the bottom layer we have the provisioning of basic resources comprising the classical cloud related resources of computing, storage and networking, and in addition specialized 'non-conventional' resources that might be offered by some infrastructure operators, such as e.g. access to extensive sensor or mobile networks. Such resources will be available only on selected nodes. They create additional value for the federation by raising the attractiveness for using XIFI. The stakeholder interaction on the resource level is essentially between infrastructure operator and federation office.

On top of the basic infrastructure we find the services that are operated by federation as a whole. This includes the maintenance of federation-wide deployed software components, which depends on coordinated procedures in order to guarantee consistency and interoperability across the federation and services which are performed in a delegated manner, such as identity and user management. Networking extends to both levels, it includes the provisioning of network access with adequate bandwidth capacities which relates to the single infrastructure operator, but also on the federation level, where the federation as a whole has to operate and maintain the private network which interconnects the XIFI nodes. Stakeholder interaction is between infrastructure operator and federator, but also between infrastructure operator and the whole group of infrastructure operators. In addition it involves the network providers as 3rd parties and the respective underpinning contracts (UC) need to be in alignment with OLA.

The services visible to the XIFI developer are based on the federation services. This includes the access to federation resources and services and user support in the form of the XIFI helpdesk, which is centralized in the XIFI federation office and involves the infrastructure operators as level 2 support. Here the OLA between infrastructure operator and the federation office relates directly to SLAs between XIFI users and the XIFI federation, hence KPIs and service targets, e.g. response times, need to be closely aligned to assure SLA fulfilment.

Vertically to those services layers are services that cross all three levels. Security and privacy issues need to be respected on all levels and require comprehensive policies and rules and well-matched processes.

A presentation and discussion of the service categories for infrastructure operators, oriented along the ITIL SLA/OLA template [13], is provided in Appendix A. In this appendix also a number of examples of OLAs are given.

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 TID gained from running the FIWARE Lab legacy platform in FI-Ware project. The information in this section complements the information in the handbook Deliverables D2.1 and D2.4.

4.6.1 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.

- Use of 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)

These numbers are default values and might be changed by any IO in order to fit the dimensioning of his node.

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.

- Volumes:

Attachment of volumes is an ongoing problem that need to be solved and this is why it has not been detailed in the below procedure.

4.6.2 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.

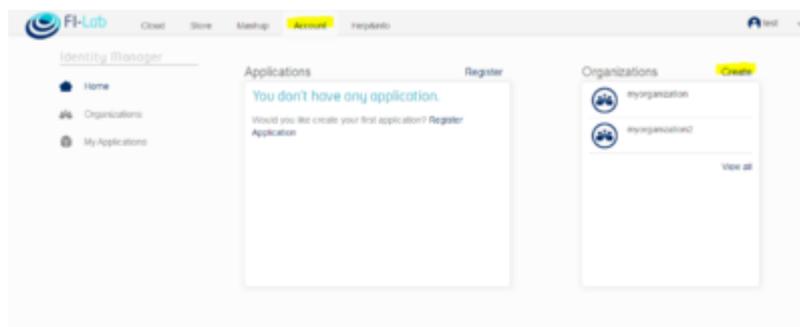


Figure 14: Account part

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 bellow)

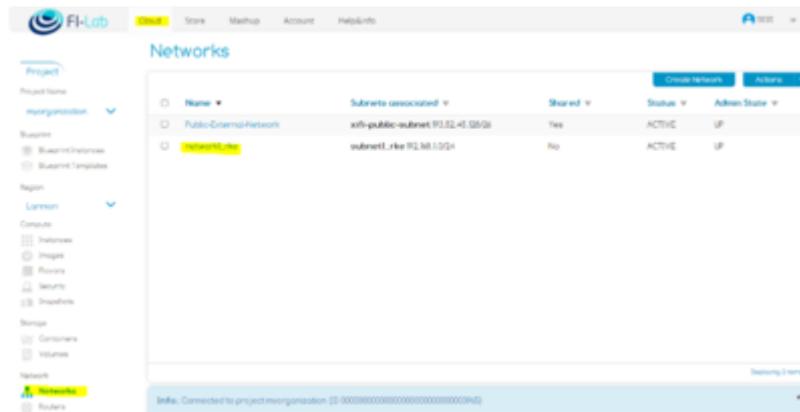


Figure 15: Create a network

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"

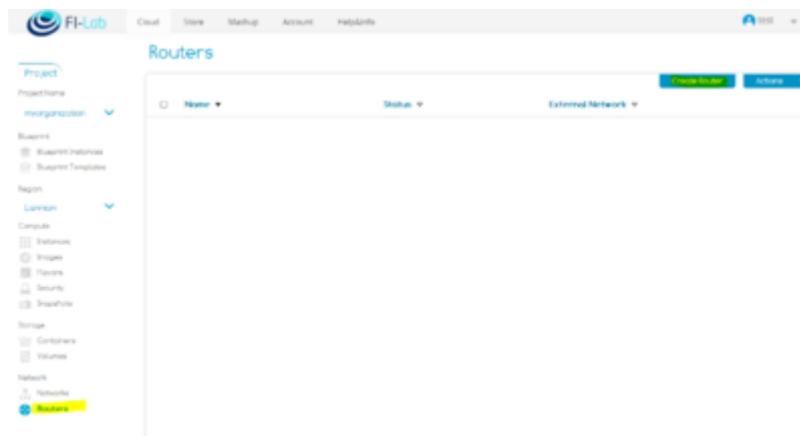


Figure 16: Create a router

- 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.

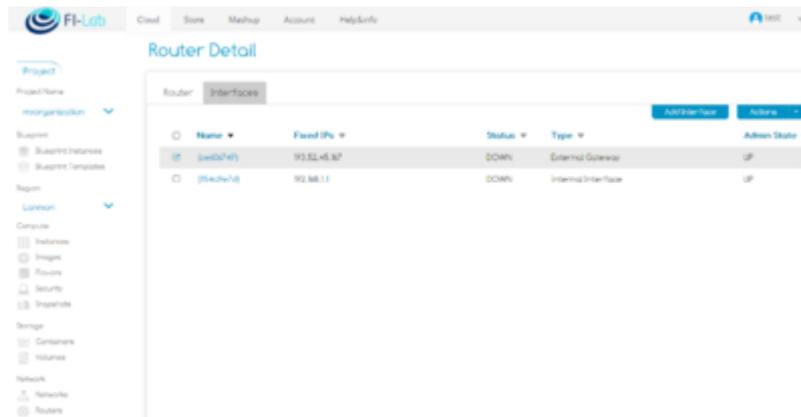


Figure 17: Add an interface

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 18).

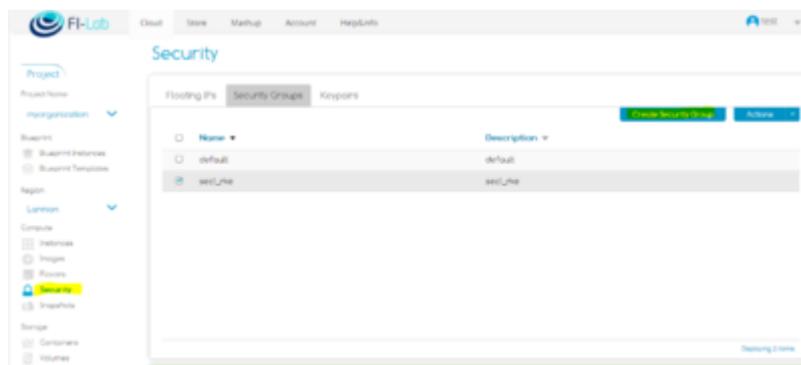


Figure 18: Create Security Group

Add (some) rules to your security rules, see Figure 19.

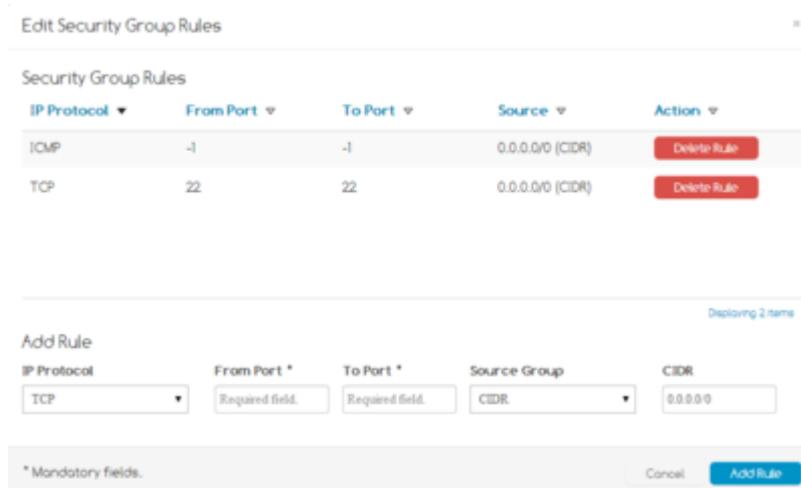


Figure 19: Add rules

Create the Keypair

Click on "Keypairs" Tab (Figure 20), 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.

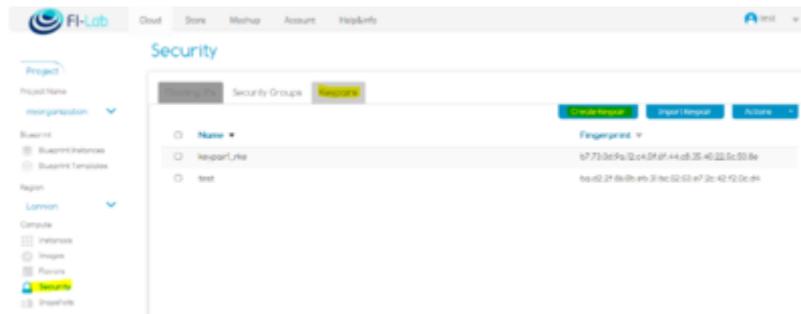


Figure 20: Define the Keypairs

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 21), select "Instances", then click on "launch New Instance"

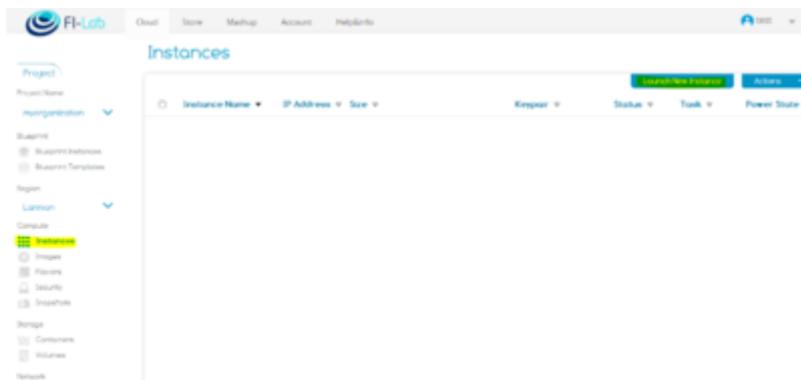


Figure 21: Create an instance

- 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

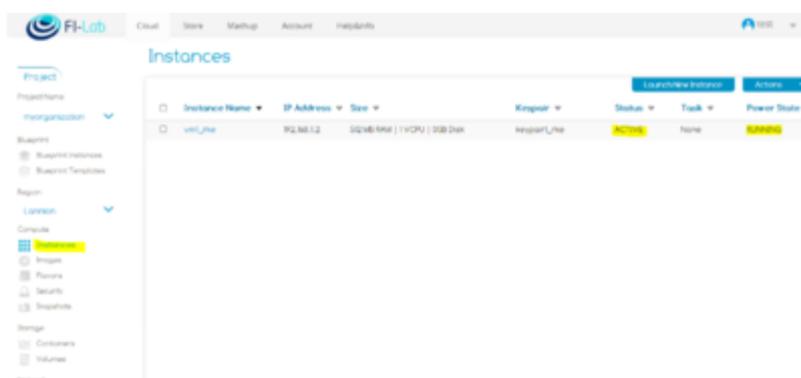


Figure 22: Launch an instance

Once launched (Figure 22), you can click on it, to check the logs and that you instance has been created successfully, Figure 23.

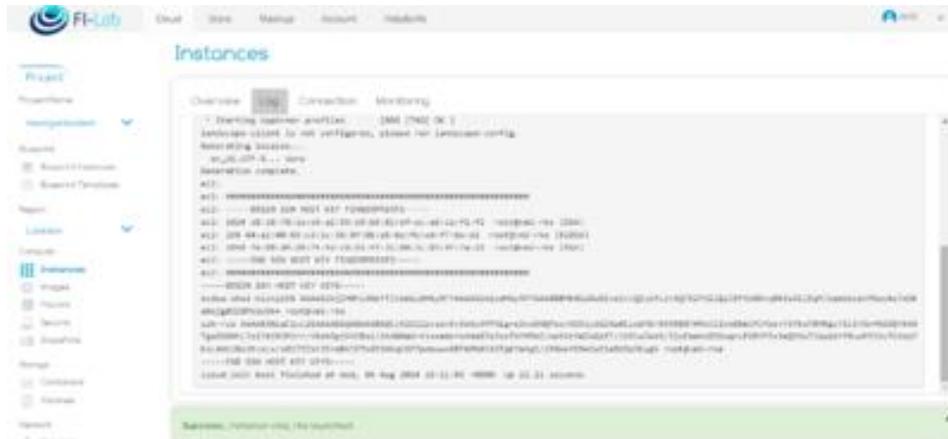


Figure 23: Instance Log

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 24)

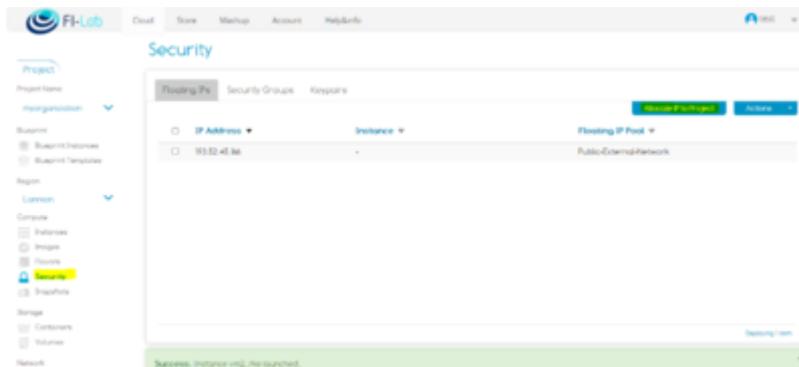


Figure 24: Allocate IP

- 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 25)

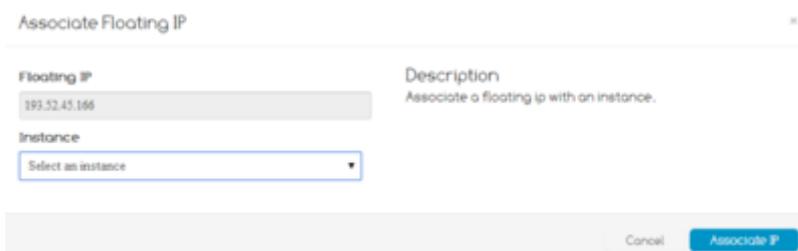


Figure 25: Associate IP

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

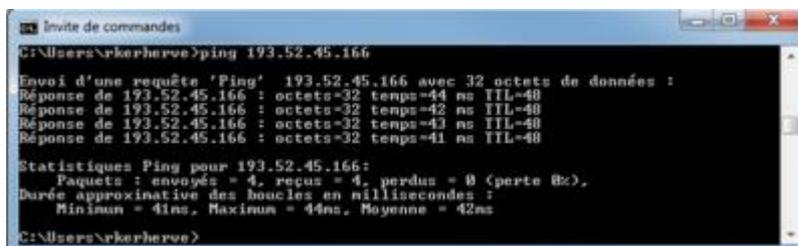


Figure 26: Ping

- Connect to your instance via SSH:

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

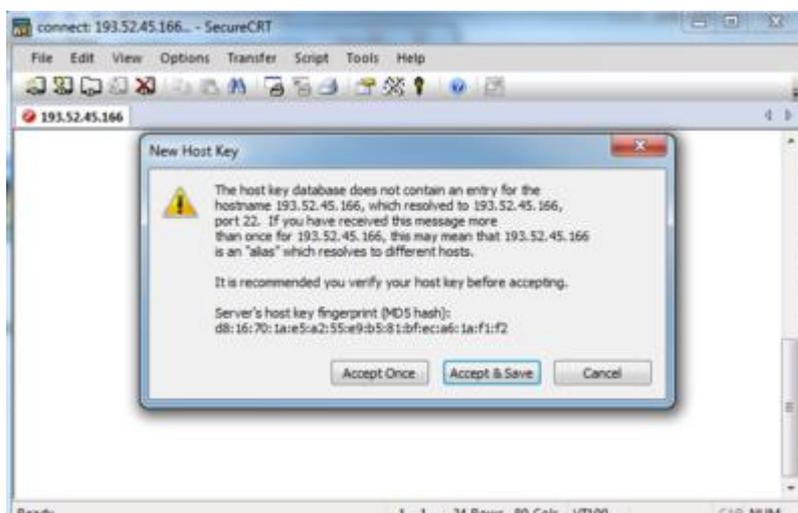


Figure 27: Connect via SSH

4.6.3 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.

4.6.4 Traceability of deployed Instances

This section deals with the IO's capability to identify who has allocated a resource on the IO's infrastructure at a given time, in the present or past.

- Correlation between Instance and public IPs in real time

You can use the nova command `nova list --all-tenants` to list all IPs used and their matching instances on a region:

```
nova list --all-tenants
```



ID	Name	Status	Networks
a8d0937f-b905-4f2f-9ad2-b35cf155979b	Access control	ACTIVE	ILB-MonitoringNet=192.168.0.7, 10.0.48.46
9a93b080-6502-4f33-8308-18436541e7b6	CEP-PTRAK	ACTIVE	panos=192.168.3.4
96d829fb-6e46-4901-98aa-544d9a69c8b5	ConnectedTV-v1	ACTIVE	UC-FI-CNT2-PrivateNetwork=192.168.103.10
311a72ce-6b75-4627-9c44-8b13c6e9af7b	Fire2FIPPP01	ACTIVE	fire4fipp_private-network=192.168.101.102, 193.52.45.137
47b0004d-15aa-4c44-a3ac-3e9c90d91155	Fire2FIPPP02	ACTIVE	fire4fipp_private-network=192.168.101.100, 193.52.45.138
3ad85423-0b32-4a2e-9030-ab05a30e7c76	Fire2FIPPP03	ACTIVE	fire4fipp_private-network=192.168.101.103, 193.52.45.139
ab5002a0-c462-4108-9550-1a2c5e68e901	KURENTO2	ACTIVE	panos=192.168.3.5
8a0f8004-c816-4e47-9814-d88bb79add8d	LecLOUDestLavié	ACTIVE	
64a7c6b4-f927-473b-a055-19b66c83fca3	MARKET-TEST	ACTIVE	panos=192.168.3.6
6b0cca3f-1213-4633-b202-d3e6e51706cc	MARKETS	ACTIVE	panos=192.168.3.7
f667c3da-927f-4d97-88a8-beb12c9c4304	MQTT_IAE	ERROR	
47d96eb5-8030-4944-b672-010ae38ee95e	Monitoring_second_ext	ACTIVE	ILB-net01=192.168.234.2, 193.52.45.144
1970c750-fb87-4404-9a3d-4cb26b30e409	NAM_NPM_DEM	ACTIVE	
3df1192a-d076-45fd-9f37-ff55b206f2a0	NAM_NPM_DEM	ACTIVE	ILB-MonitoringNet=192.168.0.2, 10.0.48.31
1fc44ab3-6ccf-496d-9151-0d99e8ebfcdc	NGSI_CB	ACTIVE	ILB-MonitoringNet=192.168.0.4, 10.0.48.32
42e259b-0d28-4088-bc6b-0d6ddf09ee14	NGSI_CB0rion	ACTIVE	
4af6e993-8ba0-491a-a59a-41c0c718921a	QT	ACTIVE	Public-External-Network=193.52.45.146
d154d8d3-0b02-445a-aa63-d34d1fc61e73	REGISTRY_PTRAK	ACTIVE	panos=192.168.3.2
308ea322-c547-41f4-a184-236443ffb6de	SecurityProbe	ACTIVE	ILB-MonitoringNet=192.168.0.6, 10.0.48.34
479a9fb5-2ff2-4cbb-ac25-aad12cc88acf	SecurityProbe	ACTIVE	
f3737266-ec57-4208-aa3b-109894233e59	TOAB2	ACTIVE	Public-External-Network=193.52.45.136
11407fc0-e19e-498d-b950-ad112b58ddf7	ubuntu	ACTIVE	
c8439b5e-dfe5-4f93-887b-572e77c5cc42	ubuntu	ACTIVE	ILB-MonitoringNet=192.168.0.5, 10.0.48.33
937583ec-ae3b-4390-98aa-c9ca47042ffc	cdva_check	ACTIVE	Public-External-Network=193.52.45.145
d3ff59ca-623d-4fe9-b21e-ead74dzed14	complex-event	ACTIVE	Public-External-Network=193.52.45.135
be08092d-0a31-40cd-9955-a3e9d5146b3f	disposable_lanmon	ACTIVE	Public-External-Network=193.52.45.161
7faae8d0-81bc-4e22-aa7e-8543b1b32008	keystone-test_image	ACTIVE	ILB_cluster_network=192.168.102.104, 10.0.48.53
3931c3d0-a3e2-4fb5-9472-8e31f437fb0e	kiwanol01	ACTIVE	Public-External-Network=193.52.45.132
ca08c183-af02-4f98-bc10-3c47d2d85ed5	mv1-v2	ACTIVE	UC-FI-CNT2-PrivateNetwork=172.16.26.15
9fcf78c3-7a1d-4995-b158-22afe146c30e	pajamakids	SUSPENDED	Public-External-Network=193.52.45.154
4a2d0151-212f-4b08-b735-20c4d2e3a706	reperio-v1	ACTIVE	UC-FI-CNT2-PrivateNetwork=172.16.26.16
al3026de-76f7-40f0-83bd-869d7f964590	solr-v1	ACTIVE	UC-FI-CNT2-PrivateNetwork=172.16.26.17
163d0c38-8723-4fa0-8f38-90aa42f3d6d0	test-erwan2-163d0c38-8723-4fa0-8f38-90aa42f3d6d0	ACTIVE	Network-Erwan_Test=192.168.222.2
eb1144b8-e990-4c04-9071-1b367e386dcb	test-erwan2-eb1144b8-e990-4c04-9071-1b367e386dcb	ACTIVE	Network-Erwan_Test=192.168.222.3
5250601f-803e-47ab-b486-be889e7c3639	ubuntu	ACTIVE	Public-External-Network=193.52.45.157
e6912443-e086-4a6a-835e-e0645b00988c	ubuntu	ACTIVE	Public-External-Network=193.52.45.163



- Identification of Instance & Instance's owner to which a public IP has been affected

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. In case of a malicious usage of a Public IP there is no possibility for an IO to obtain the related user data other than by starting a legal case against the malicious user.

4.6.5 Local catalogue management

Currently, the images that have to be located on each local catalogue come from the Spain node (FIWARE catalogue). For the purpose of copying the various images to the rest of IO, we have installed an Apache server in the Spain glance instance controlled by user and administrator password in order to access to it and to download the corresponding images, as described below. The server is located at <https://glance.lab.fi-ware.org>.

Currently, there are only users for Lannion, Waterford, Berlin and Trento. If any other IO wants to access it, he should contact the administrator of this server (Fernando López, fernando dot lopezaguilar at telefonica dot com). In the following table you can see the responsible of each IO

The list of persons with authorized access per IO is the following:

Nodes	Responsible	Account
Lannion	Riwal Kerherve	glance2
Lannion	engineering@imaginlab.fr	glance5
Berlin	Thomas Günther	glance4
Trento	Cristian Cristelotti Coll	glance3
Federator	Joe Tynan	glance1

Table 36: List of users with access to the glance-apache server in Spain

A personalized email was sent to the users with the password to be used for the server. The administrator of this server maintains the correlation between users and password for all users.

We are evaluating the alternative that the Spain administrator node will automatically update the images for the other nodes but we are still in the process of defining the procedure to do it.

Images to be added to the local catalogue

A list of cloud images is required to be present on the catalogue.

- repository-image-R3.2-2
- dbanonymizer-dba
- marketplace-ri_2
- meqb-image-R2.3
- cep-image-R2.3
- datahandling-ppl
- orion-psb-image-R3.3
- registry-ri
- ofnic-image-R2.3
- kurento-R4.2.2

- kurento-image-4.0.0
- kurento-image-R3.3
- cdva-image-R2.3

NID is a property metadata and anytime we create a new image, this NID number need to be associated to its glance metadata.

Procedure to add the required images

Below we show at the example of the dbanonymizer-dba how the required images can be added. The full information on all images is given in Appendix B.

- 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>
```

4.6.6 Managing Images

If you need to manage an image, i.e. modify it permanently, you can use guestfish [20]. 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 to do then is to type "run" which 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 leave the guestfish tool. You can now use your modified image file.

4.6.7 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).

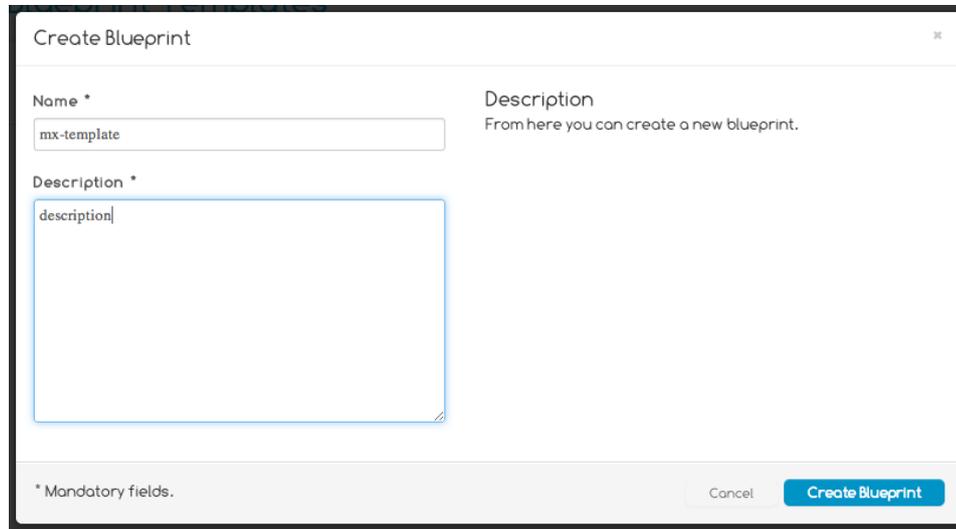


Figure 28: Create blueprint template

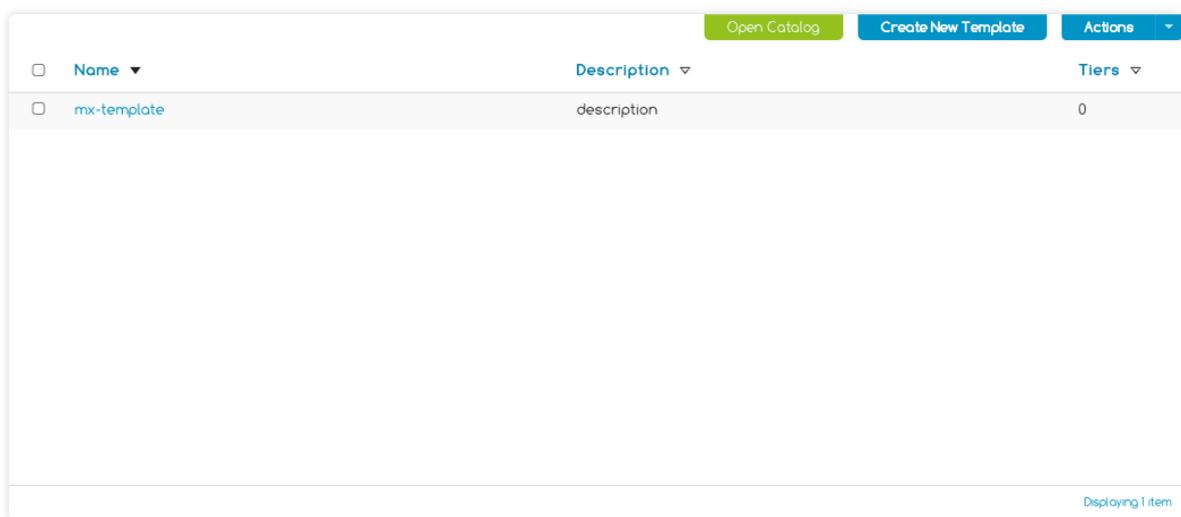
You have to complete the information related to the name of this template and a description in order to know afterward what the purpose of this template was. Then you should click on the "Create Blueprint" button to finalize the creation of the template.

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



Name	Description	Tiers
mx-template	description	0

Figure 29: 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).

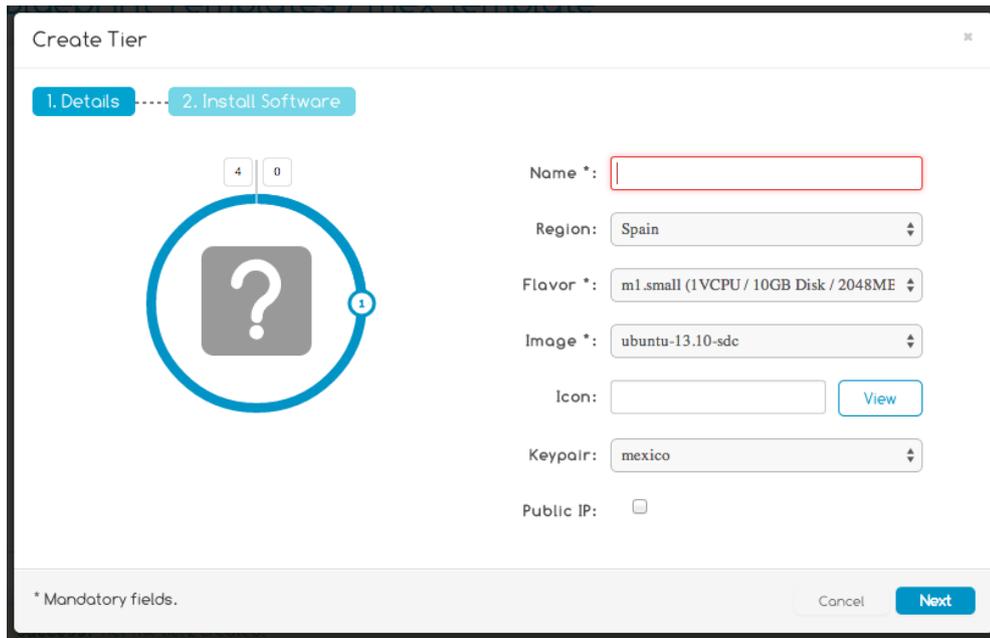


Figure 30: Create a tier

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 31: Adding software to a tier

To finish the template, click “Create Tier”.

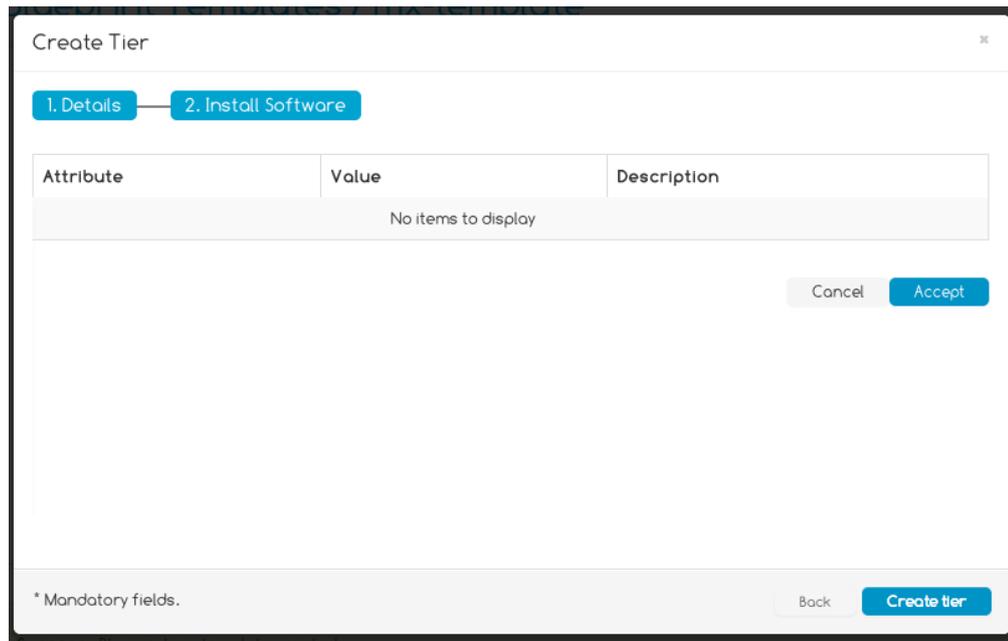
Editing/Creating the software attributes.

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 32: 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.



Attribute	Value	Description
No items to display		

Figure 33: 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 29. 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).

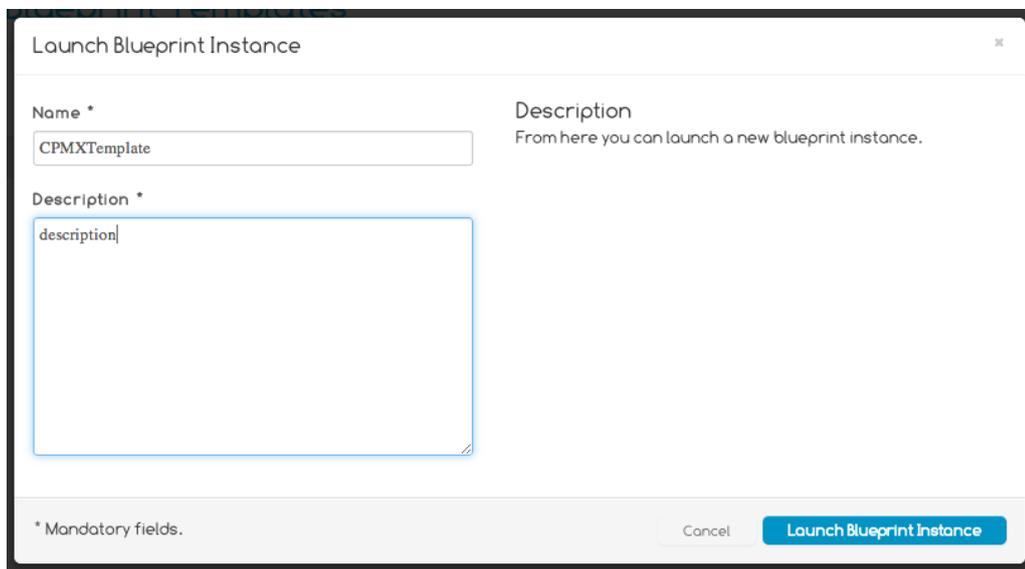
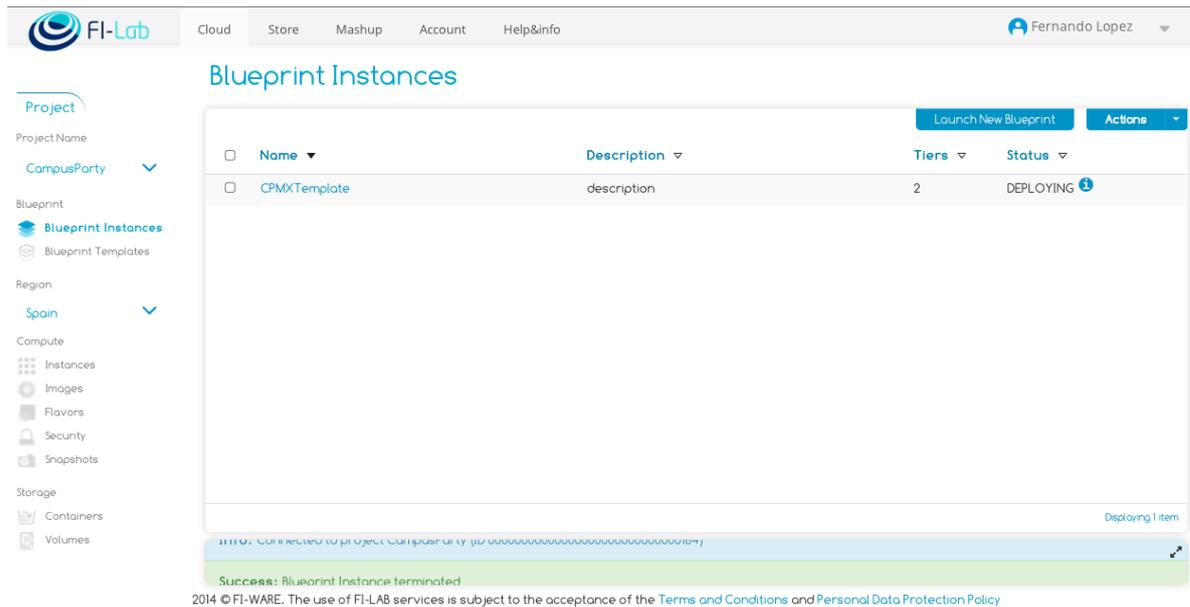


Figure 34: Launch a blueprint template

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.



The screenshot shows the FI-Lab interface for managing Blueprint Instances. The left sidebar contains navigation options: Project (CampusParty), Blueprint (Blueprint Instances, Blueprint Templates), Region (Spain), Compute (Instances, Images, Flavors, Security, Snapshots), Storage (Containers, Volumes), and a user profile for Fernando Lopez. The main area is titled 'Blueprint Instances' and features a table with the following data:

Name	Description	Tiers	Status
CPMXTemplate	description	2	DEPLOYING

Below the table, a green notification bar states: 'Success: Blueprint Instance terminated'. At the bottom, a copyright notice reads: '2014 © FI-WARE. The use of FI-LAB services is subject to the acceptance of the Terms and Conditions and Personal Data Protection Policy'.

Figure 35: Blueprint instances

4.6.8 Use Case Handling

Information to get

When a Use Case should be deployed within a region, the person in charge of the Use Case needs to contact the IO of the node if the quota applied to the node does not fit the needed dimensioning. Indeed, most of the time, quota and flavours on a node are quite limited to prevent abuse. These can easily be changed or extended by the IOs, but it requires manual intervention by the IO.

Below is a list of the information that an IO needs in order to adapt quota, flavour or configuration applied to a tenant.

- Global Architecture presentation from the Use Case
- Description of Instances:
 - Numbers of instances needed
 - For each of them, give the dimensioning needed:
 - Memory,
 - Disk,
 - Number of processors
 - etc.
 - Snapshot availability in case of migration from an existing server to XIFI
 - Network connectivity:
 - Number of network interfaces per instances,
 - Configuration of the interfaces wanted,
 - Ports to be opened:
 - Management ports: Give the public IP needed in order to do the filtering
 - Service ports that will need to be opened (Service provided by the Use Case)
 - Login:
 - FIWARE 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

4.6.9 Tenant customization

Quota

- You might need to list your default quotas, so use the following commands (respectively for compute/network/block storage):

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

- If you need 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 8192 <Tenant_ID>
quantum quota-update --tenant_id<Tenant_ID> --network 5
cinder quota-update--gigabytes 50 <Tenant_ID>
```

Flavors

- You might need to list your default quotas, so use the following command:

```
novaf Flavor-list
```

- In case you want to add a new flavor, just 2 steps are needed:
 - Create your new tenant:

```
novaf Flavor-create --is-public
<true|false><flavor_name><ID><ram><disk><vcpus>
```

E.g.:

```
novaf Flavor-create --is-public false Test-flavor auto 2048 0 2
```

- Then your freshly created flavor needs to be available for the tenant, use the following command:

```
novaf Flavor-access-add <flavor><tenant_id>
```

E.g.:

```
novaf Flavor-access-add a5abb478-9672-46f1-979e-99d2ca023fdc
00000000000000000000000000000000xxxx
```

4.6.10 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 36). Then choose "switch session" and click on the tenant you would like to modify roles for some users.

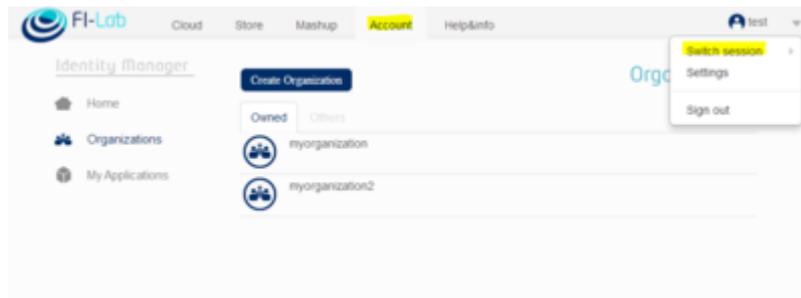


Figure 36: Manage tenant – select user

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 37: Modifications on a tenant

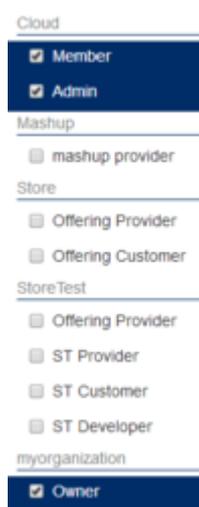


Figure 38: Modifications on a tenant II

- 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 37 for an example from Lannion node, 2014-07-15 - 2014-08-13.

Tenant ID	Instances	RAM MB-Hours	CPU Hours	Disk GB-Hours
0009	2	53271.77	104.05	0.00
0049	1	64407.91	125.80	0.00
00356	2	5505024.48	2688.00	53760.00
002559	1	1376256.12	672.00	13440.00
002983	11	6720124.52	4993.41	54212.24
002988	3	2578811.02	1259.19	25183.70
003437	5	11010048.97	5376.00	107520.01
003449	9	33405.01	62.28	19.77
003478	3	24772610.17	8064.00	161280.01
003847	1	11010048.97	5376.00	107520.01
003851	1	344064.03	672.00	0.00
003940	1	195.70	0.10	1.91
003965	6	269624.50	388.78	918.84
003997	26	5758557.76	3816.22	0.00
004004	9	4930476.50	2411.06	58553.86
004012	1	344064.03	672.00	0.00
004019	1	344064.03	672.00	0.00
004098	1	2752512.24	1344.00	26880.00
004287	3	74.67	0.15	0.00
004291	1	1277297.91	623.68	12473.61
004351	2	595610.87	290.83	5816.51
38aec686f107485ebc1ca9763d96d958	5	6881280.60	3360.00	67200.01

Table 37: 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 38 shows an example result.

nova-manage vm list

instance	node	type	state	launched	image	kernel	ramdisk	project	user	zone	index
LeCloudC	node-3	m1.s	active	22/04/2014	760d4409-	38aec686f	f7b2f1315c4a47b284aa142fb0728d43			None	0
estLaVie		mall			731c-4009-	107485eb					
					b368-	c1ca9763d					
					4a7ad78d83	96d958					
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 38: VM list

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

The maintenance process outlined in subsequent sections relates in some of its procedures to the e-business Telecoms Operation Map (eTOM) framework defined by the Tele Management Forum. The following table summarizes the eTOM operations processes in scope of the maintenance process.

Please note that the term "developer" used throughout this section refers to the FI developer as a user of the XIFI federation for the purpose of developing an FI application. Further stakeholders and roles are described in the next section.

Please also note that the eTOM framework is focussing on a customer perspective. With regards to the maintenance process described here, it often conveys customer requirements into requirements on the implementation of the detailed maintenance procedures (i.e. the artefacts of the maintenance process). Hence, eTOM processes very often translate into Quality Assurance processes (for maintenance) rather than maintenance procedures themselves. There has been substantial work on detailing maintenance procedures in the scope of WP5 Task 5.4. This will be documented by D5.5, which will update this Deliverable.

eTOM process grouping	Implementation in XIFI	References
Developer Contact Management	FI developers interact with dedicated maintainer roles for components, sub-systems and infrastructures. Maintainers are visible to other maintainers and to the FI developer. Component, sub-system and infrastructure owners (if any) are only visible to their respective maintainer.	The interaction between maintainers and between maintainers and owners is outlined in the maintenance procedures. The interaction between FI developer and maintainers is outlined by the procedures too.
Developer Problem Management	This is part of the Quality Assurance Management process and is integrated with the JIRA helpdesk. It requires to trace a problem from the moment the issue is raised until it is resolved, gathering various quality	There is no Quality Assurance process described yet. This will be

	metrics along its lifetime to evaluate optimize the issue handling process.	future work.
Developer Self Empowered Maintenance	This is not foreseen in XIFI. FI developers need to interact with maintainers through the JIRA helpdesk in order to request a maintenance procedure. This is required to protect the XIFI federation from harm.	The interaction between FI developers and maintainers is outlined by the maintenance procedures.
Developer SLA Management	This is part of the Quality Assurance Management process and is integrated with the JIRA helpdesk. It refers to the evaluation of the timeline of issue handling and to the appropriateness of the issue resolution.	There is no Quality Assurance process described yet. This will be future work.
Fault Management	Fault management is finally handled by unscheduled maintenance processes. Indication of a (potential) fault can be given by FI developers through interaction with the JIRA helpdesk or through internal quality assurance processes based on scheduled maintenance processes. Incident handling is partly covered by the XIFI maintenance processes, but mainly relies on a node infrastructure's internal maintenance which is out of scope for this document. A node's incident handling procedures effectiveness is nevertheless considered in the process of quality assurance.	Fault management is described in the scope of applicable maintenance procedures. It is currently covered in the scope of help-desk interaction.
Interaction Management	This is achieved through the procedures regarding the interaction between maintainers and between maintainers and the FI developer. To optimize this interaction, the quality assurance process monitors the effectiveness of these procedures.	The interaction between maintainers and between maintainers and owners is outlined by the maintenance procedures described. The interaction between FI developers and maintainers is outlined by the procedures. The Quality Assurance process is described too.
Personalize Developer Profile	This is handled by the JIRA helpdesk and is referring to the information that can support the interaction between FI developer and maintainers mainly to optimize this interaction in course of a quality assurance process. This is similar to a CRM profile.	Interaction with the JIRA helpdesk is described in section 5.3.
Resource Performance Management	Resource performance management is partly in scope of the quality assurance process and partly it is described by existing maintenance procedures. Resource performance is mainly addressed by scheduled maintenance procedures but may be part of unscheduled processes or part of an incident handling in case the performance is significantly affected by a spurious events.	XIFI maintenance procedures are described. The Quality Assurance process is described too.
Resource Test Management	The maintenance process utilizes component or sub-system tests as a means to evaluate the need to initiate fault management or unscheduled maintenance procedures. Maintainners establish the interaction	The interaction between maintainers is described by the maintenance procedures and by

	between the test infrastructure (represented through testbed maintainers and test owners) and component, sub-system, infrastructure owners to conduct test and to identify next step stakeholders.	stakeholder interaction through the help-desk in section 5.3.
Service Performance Management	Handled by the same procedures as applied to resource performance management.	XIFI maintenance procedures are described. There is no Quality Assurance process described yet. This will be future work.
Service Problem Management	Handled by the same procedures as applied to fault management.	Described in the scope of applicable maintenance procedures.
Service Quality Management	This is handled by the quality assurance process consisting of monitoring, performance evaluation, fault detection and issue handling jointly taking part in an optimization loop on the maintenance procedures.	There is no Quality Assurance process described yet. This will be future work.
Validate Developer Satisfaction	This is part of the Quality Assurance Management process and is integrated with the JIRA helpdesk.	There is no Quality Assurance process described yet. This will be future work.

Table 39: eTOM framework references to the XIFI maintenance process

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 40: 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
GOWEX	megido@gowex.com
WIGNER	xifi-support@wigner.mta.hu
UTH	nitlab@inf.uth.gr
BTH	xifi-helpdesk@bth.se

Table 41: 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.

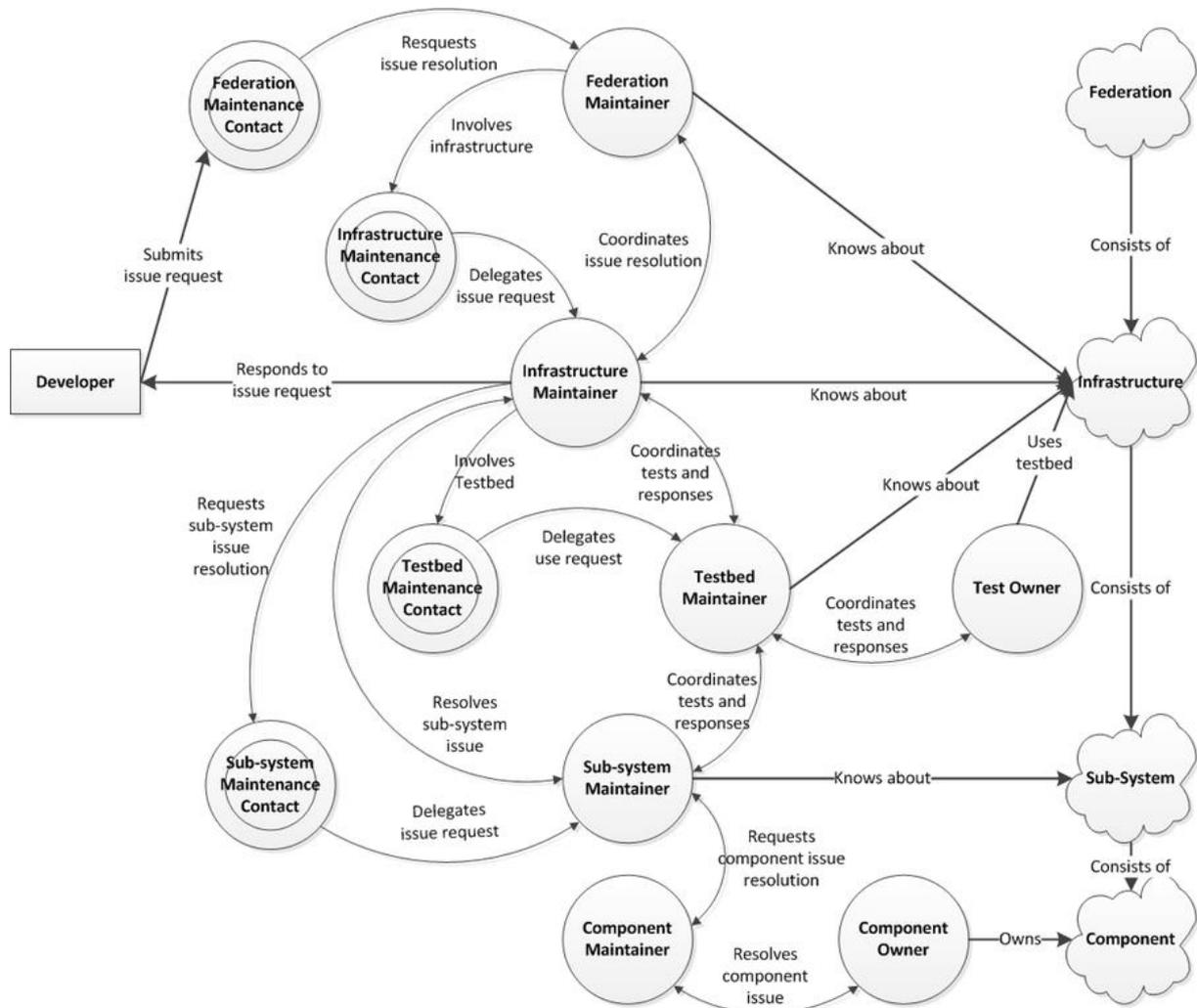


Figure 39: Sample Maintenance Stakeholder Interaction (Developer initiated issue request on a sub-system issue)

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

XIFI aims to utilize 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.

In general, ownership of a ticket and creation of sub-tasks in the maintenance process (e.g. performing a sequence of maintenance procedures) reflects both responsibilities and activities. For example, if the

federation maintainer hands over responsibility to an infrastructure maintainer, the ownership of the corresponding ticket may reflect who (currently) is responsible for performing a maintenance procedure. If this infrastructure maintainer now delegates a sub-task to one or more sub-system maintainers, a dedicated ticket for each of the sub-system maintainers may be created and the infrastructure maintainer has to wait for completion of all delegated tickets until he can close or forward ownership of his ticket. Thus, the flow of the ticket reflects the workflow of maintenance.

Maintenance procedures are not directly initiated by developers. Nevertheless, it may be required to initiate a maintenance procedure in consequence of a due issue resolution (based on the developer's issue reporting). In that, the JIRA help-desk is considered the main interface between developer and maintainers. In that case a ticket is generated by the developer and ownership of the ticket is taken by the federation maintainer, infrastructure maintainer or, more rarely, by a sub-system maintainer as specified by the scope of the issue reported. Subsequently, ticket ownership is as handled as summarized above. When the issue is finally resolved, the ticket is closed and the developer is notified accordingly. This may or may not coincide with the termination of the maintenance procedure(s) initiated along with the developers issue report depending on the procedures taken.

Notifications are generated along with a maintenance procedure. The exact time of notification depends on the classification of the maintenance process: scheduled, unscheduled or in the course of handling an incident. The JIRA help-desk is one of many channels utilized for notification. It is not the only one because only developers and maintainers may be reached via this path. Users in general may be out of reach and will need different consideration.

5.3.1 Interaction of maintainers

The process described here is considered part of the operational level agreement (OLA) between infrastructures. It is complemented by further service level agreements and procedural agreements considered out of scope for this document/section.

When implementing a maintenance task, stakeholders interact as initially described in the previous section through the helpdesk⁷. The following table maps possible actions between stakeholders to a suitable interaction with the help-desk.

Help-desk implementation of stakeholder interaction		
Procedure	Short description	Help-desk implementation
Respond	Keeping contact with the issuer of a request (i.e. another maintainer) until the issue has been resolved, taking responsibility to resolve the issue.	Picking a ticket as an assignee, resolving the issue under the ticket, reporting on the resolution and replying to the issuer before closing the ticket.
Initiate	Initiate a maintenance task, taking responsibility to monitor the task until successful completion.	Creating a task (or sequence of tasks) under the ticket and reporting on the result.
Coordinate	Coordinate activities of one or more other maintainers in resolving an issue until the issue has been resolved, taking	Follow processing of a particular ticket, monitor progress, detect and resolve deadlocks, and report on the progress to

⁷ There is not yet a final decision for a particular trouble ticketing system but preference has been given to share the JIRA help-desk facility as outlined by subsequent section. In particular during early deployment and operations it is reasonable to assume that most maintenance issue will be reported by federation users. Sharing the Jira between both user support and maintenance process seems appropriate to reduce management overhead. It is possible that this may change in the future.

	responsibility to monitor the progress and to collect and communicate intermediate results.	the assignee of the ticket.
Delegate	Delegate responsibility on resolving an issue to one or more other maintainers. If delegating to multiple maintainers (i.e. partitioning of the problem) this may include nomination of a Responder and Coordinator.	Handing over the ownership of a ticket to a new assignee or create new tickets (for sub-tasks under the ticket) but keeping ownership of the originating ticket. In case of handing over ownership, the recipient (i.e. the new assignee) should inform the creator of a ticket on the change of ownership.
Escalate	Delegate responsibility on resolving an issue to a supervisor (e.g. a higher authority). The supervisor then may re-assign responsibility by delegation to more suitable other maintainers after re-evaluating the issue.	Handing over the ownership of a ticket to a new assignee. There should be rules how often this is allowed to happen.

Table 42: Stakeholder interaction

5.3.2 Interaction of maintainers and developers

Following up on the maintenance stakeholder interaction example given in Figure 39 and considering that there exists an interaction between developers (i.e. federation users) and maintainers in consequence of an issue report, we need to map the handover between user support and federation maintenance as well as their interaction in course of an issue resolution onto a ticketing process that could be realized by the JIRA ticketing system. The process described here is considered part of the service level agreement (SLA). It is complemented by further service level agreements and the implementation of usage terms and conditions⁸.

Reporting problems will be done using a ticketing tool and, for a question of commodity, the same tool used by FI-WARE was decided to be used. Jira is this ticketing tool and its flexibility and the possibility to add a large panel of add-on permits to limit the restriction we could have in the definition of the issue reporting process.

The following figure describes the different states of a reported issue during its life cycle when passing from one hand to another.

⁸ Terms of use are considered out of scope for this document but are still an element and prerequisite for implementing service level agreements.

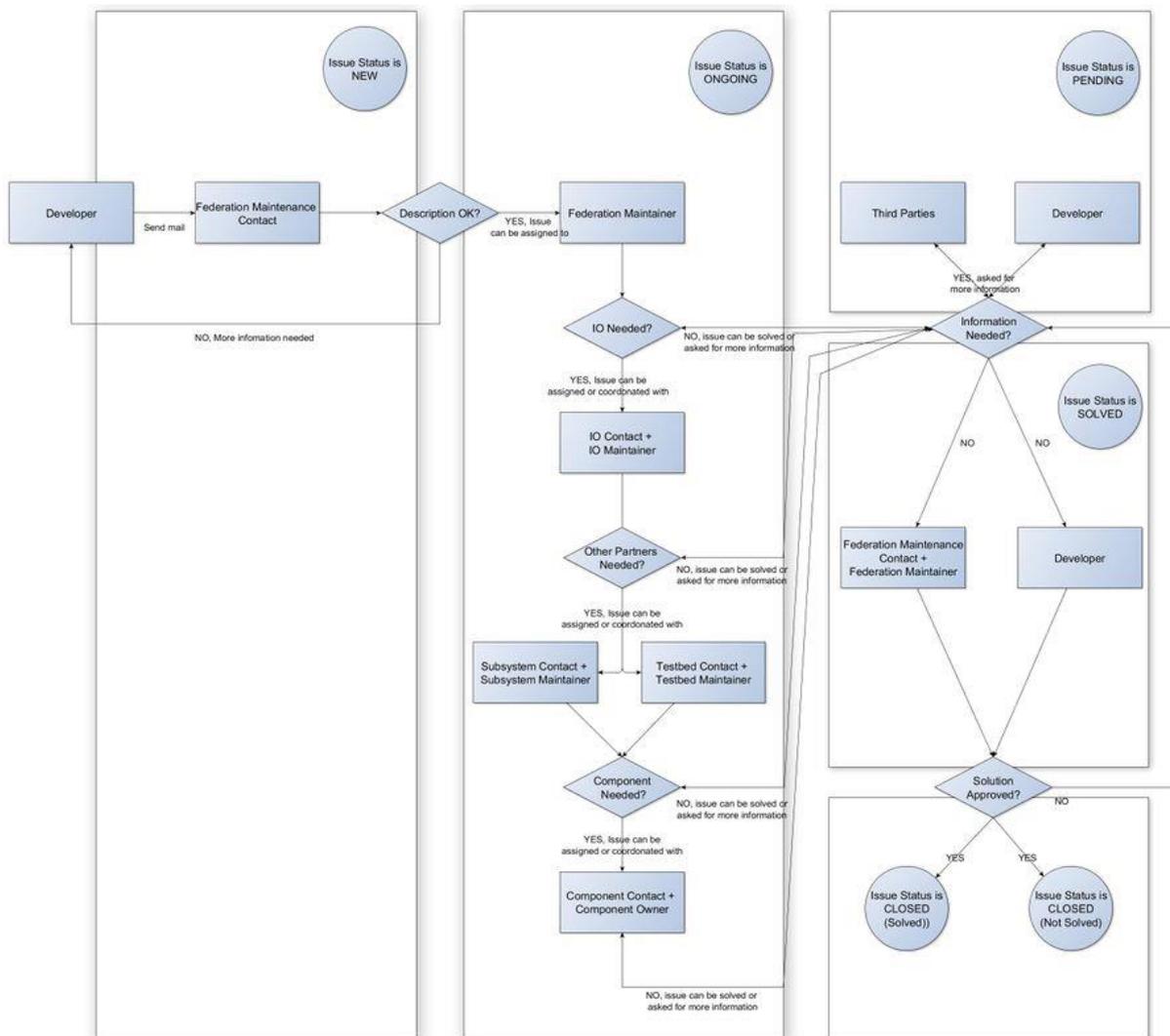


Figure 40: Ticket flow of a reported issue

5.3.3 Ticket handling in the interaction of maintainers or developers

In all cases, when passing an issue from a main contact to a maintainer (example: from an infrastructure maintenance contact to an infrastructure maintainer), it will be treated by Jira by a special tag where it will be possible to add the maintainer’s name to whom the issue shall be assigned. This tag will be called “Maintainer assigned”. It is up to each stakeholder (federation, infrastructure, sub-system, test-bed and component) to decide if they judge this as a suitable and sufficiently flexible solution or if it is necessary to use an internal ticketing tool to support this part.

In this process the priority of a ticket will be defined as follows:

HIGH

Highest priority of a ticket. At this level, the problem reported should be treated at highest priority and should be resolved urgently. This level must be used only for reported problems involving infrastructure and end-user service failure and may cause subsequent unscheduled maintenance or even incident handling.

MEDIUM

At this level, the problem reported should be treated with reasonable priority and may cause scheduling of a maintenance task (scheduled or unscheduled maintenance).

LOW

Lowest priority of a ticket. This level is used only to report minor problems that could be solved in a low priority and may cause maintenance tasks in scope of scheduled maintenance. Problems involving end-user service failure shall not be reported with this priority.

Further during the life cycle of a reported problem, the corresponding ticket can assume one out of the following states:

NEW

A developer raises an issue by sending a mail. For a practical reason of Jira's licenses, no developer will get an account in Jira. Jira will automatically transform email into a ticket:

- This ticket will be created and its status will be set to NEW.
- The issuing developer or maintainer will be the requester
- The federation maintenance contact will be the person assigned to the ticket as well as a watcher
- The initial priority is set to LOW (as the machine routine cannot decide upon that)
- The title of the ticket will be the title of the mail
- The description of the ticket will be the body of the mail.

The federation maintenance contact will decide if the ticket contains sufficiently detailed information to be assigned to a federation maintainer or not. If not, he/she will create a follow-up asking the developer for more information.

The developer will be notified by mail and is assumed to respond to this mail. Jira will automatically transform the answer into a follow up in scope of the ticket created earlier.

At this stage, the federation maintenance contact can assign this ticket to an infrastructure or sub-system maintainer, passing the ticket status to ONGOING and defining the priority as either HIGH, MEDIUM or LOW.

ONGOING

At this stage, the status of the ticket will be as follows:

- The ticket status will be set to ONGOING.
- The issuing developer or maintainer will be the requester
- The federation maintenance contact will be the person assigned to the ticket in addition to being a watcher of the ticket
- The tag "maintainer assigned" will contain the name of the federation maintainer assigned to the ticket
- The title of the ticket will be the title of the mail
- The description of the ticket will be the body of the mail.

Then, the federation maintainer will decide if an infrastructure maintainer or multiple infrastructure maintainers will be required to process to the ticket:

- If the answer is NO, then it means that either the federation maintainer has decided that more detailed information is required before it could be processed, resolved or passed to another stakeholder, or can be resolved immediately without involving other stakeholders. The status of the ticket then will pass to either PENDING or SOLVED.
- If the answer is YES, the federation maintainer will assign the ticket to one or more infrastructure maintainers depending on the issue and on the need for coordination between infrastructure maintainers. The following options exist:
 - Assign one or more infrastructure maintainer contacts to the ticket
 - Optionally remove himself from the list of stakeholders assigned to the ticket
 - Create new tickets and assign them to different infrastructure maintenance contacts, and link them to the main ticket.

- The status of the ticket will be defined as ONGOING
- In all cases, the federation maintenance contact will stay as a watcher, since he/she still is responsible to find a solution in the process.
- For each infrastructure maintenance contact assigned, there will be an infrastructure maintainer added under the tag “Maintainer assigned”.

The same procedure is applied to handle the following issues:

- Is a Subsystem Maintainer needed to solve the issue?
- Is a Test bed Maintainer needed to solve the issue?
- Is a Component Maintainer needed to solve the issue?

PENDING

At any time while a ticket is in status ONGOING, the person assigned to the ticket (federation maintainer, etc.) can decide if supplementary information is required from the issuer of the ticket (developer or maintainer). In this case:

- The state of the ticket will change from ONGOING to PENDING.
- The person(s) previously assigned to the ticket will change from assignee(s) to watcher(s)
- The maintainer or developer will be the person assigned to the ticket.

When maintainer or developer provides the information requested, ticket status and person(s) assigned to the ticket will change back to their previous states and assignments. This step should take place by issuing a response to the notification received, as a developer won't have an account to JIRA.

In some cases, it can happen that support or additional information is required from a “third party” to proceed in the resolution of an issue. In this case:

- The status of the ticket will change from ONGOING to PENDING.
- The person(s) previously assigned to the ticket will change from assignee(s) to watcher(s) and the third party will be the person assigned to the ticket.
- When the third party will provide his contribution, the status and the person(s) assigned to the ticket will go back to their previous states. This step should be done by doing an answer to the notification received, as the third party won't have an account to JIRA.

As an example, an NREN could be one of these third parties. Many of them are not participating in the project but infrastructure maintainers dependent on their support.

SOLVED

If one of the Stakeholders was able to solve the issue, the ticket will receive the following modification:

- It will contain a documentation of the issue resolution in the field meant for this.
- Its state will change from ONGOING to SOLVED
- The person(s) assigned to the ticket will change to watcher(s)
- The developer or maintainer issuing the ticket and the federation maintainer will become assignees. Their assignment will approve the resolution.
- If the resolution would not be approved by at least one of these 2 parties, the ticket would move change to the previous state (state ONGOING, assignee(s) and watcher(s)). The ticket would also contain the reason why the resolution was not approved.
- If the resolution has been approved then it will change to its final status CLOSED.

Note: There can be cases where a ticket requires activities of more than one component owner, or of a component owner and an infrastructure owner. In such case the activities can be either in *parallel* where new and separate tickets will be created for each involved stakeholder, 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.

CLOSED

The status of a ticket is changing from the SOLVED to CLOSED finally when the issuer of the ticket gives his approval to the resolution. This step should be done by the assignee responding to the notification that created the ticket.

In some cases, the federation maintainer will have to enforce closing a ticket when solved. One of the reasons could be that the developer or maintainer issuing the ticket does not reply to a request to approve.

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.

When updating a ticket by creating a follow-up the assignee will have the choice to make this follow-up readable for the developer (or rarely for issuing maintainers) that issued the original ticket. Two different types of follow-ups then will be known to the ticketing tool:

- A private follow-up is accessible only by internal stakeholders and is invisible for a developer or for third parties;
- A public follow-up is visible to everybody.

5.3.4 Notifications in ticket handling

A notification here is considered a mail that consists of

- A Subject that contains ticket ID and title of the ticket
- A body part consisting of
 - The follow-ups, ordered from the newest to the oldest or
 - The ticket description containing creation/opening date, title, priority, status, issuer, watcher(s), assignee(s) and description of the issue

During the life-cycle of a ticket, the following states can be reached causing particular notifications to be released when entering or leaving the state:

When entering NEW state

Issued upon creation of a ticket

- when an issue request mail is sent by a developer or maintainer to the XIFI support - in return a notification mail should be sent to the issuer informing him
 - about the ticket number opened for his request
 - that his request is taken into consideration and will be treated shortly
 - Notification to the Federation Maintenance contact

or issued upon an update of the ticket adding information

- when a notification should be sent to either the developer or maintainer issuing the ticket or to the federation maintenance contact depending on the context.

When entering ONGOING state

- A notification should be sent to the developer informing him that the ticket is in an ONGOING status
- No notification should be sent to the developer when there is an update in the watcher or assignee list
- A notification should be sent to the person(s) assigned and to the watcher(s)

- Anytime there is public follow up, a notification should be sent to the developer, watcher(s), and to the assignee(s)
- Anytime there is private follow up a notification should be sent to watcher(s), and to the assignee(s)

When entering PENDING state

- A notification should be sent to the developer informing him that the ticket is in a PENDING status.
- A notification should be sent to the third party(ies) and to the watcher(s)
- Anytime there is follow up, a notification should be sent to the developer, watcher(s), and to the assignee(s). It must be taken into account that in some case the developer and the assignee might be the same person.

When entering SOLVED state

- A notification should be sent to the developer informing him that the ticket is in a SOLVED status and that he should reply to this mail to approve or not the solution proposed.
- A notification should be sent to the person(s) assigned and to the watcher(s)

When entering CLOSED state

- A notification should be sent to all persons involve in the ticket containing the new status of the ticket and the solution proposed.

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 43: 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 Escalation Cause
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 44: 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` and (internally) under `XIFI:Components`. 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`, including both Generic Enablers developed by XIFI and FI-WARE. 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	[1]	unpublished
Access Control GE	yes	yes	XIFI	THALES	published	published
Big Data GE	yes	yes	FIWARE	TID	published	published
Cloud Portal	yes	no	XIFI	UPM-DIT	published	published
Context Broker GE	yes	yes	FIWARE	TID	published	published
DEM Adapter	yes	yes	XIFI	SYNELIXIS	published	published
Deployment and Configuration Adapter	yes	no	XIFI	SYNELIXIS	[7]	published
DCRM GE	yes	yes	XIFI	CREATE-NET, FI-Ware	[9]	unpublished
DNS as a Service	yes	no	XIFI	WIT, TSSG	[10]	unpublished
Federation Manager	yes	yes	XIFI	TUB	published	published
Federation Monitoring	yes	yes	XIFI	CREATE-NET	[12]	published
FIWARE Lab App template	yes	no	XIFI	CREATE-NET	published	published
Identity Management GE	yes	yes	XIFI	UPM-DIT, ENG, CREATE-NET	published	published
Infographics and status pages	yes	no	XIFI	CREATE-NET	published	published
Infrastructure Toolbox	yes	yes	XIFI	CREATE-NET	published	published
Interoperability tool	yes	yes	XIFI	IT-INNOV	published	published
Monitoring Dashboard	yes	no	XIFI	WIT	[19]	unpublished
NAM Adapter	yes	yes	XIFI	UPM	published	published
Network Provisioning Manager	yes	no	XIFI	TID	[21]	unpublished
NGSI Adapter	yes	yes	XIFI	TID	published	published
NPM Adapter	yes	yes	XIFI	TI	published	published
OpenNaaS	yes	no	XIFI	I2CAT	[24]	unpublished
OpenStack Data Collector	yes	yes	XIFI	CREATE-NET	published	published
Path Computation Element	yes	no	XIFI	TID	[26]	unpublished
Platform as a Service Manager GE	yes	no	XIFI	TID, FI-Ware	[27]	published
Quick Online Test	yes	yes	XIFI	WIT	[29]	unpublished

Resource Catalogue and Recommendation tools	yes	no	XIFI	ATOS, UPM-SSR	published	published
Software Deployment and Configuration GE	yes	yes	XIFI	TID, FI-Ware	[31]	published
Security Dashboard	yes	no	XIFI	ATOS, THALES	published	published
Security Monitoring	yes	yes	XIFI	ATOS	published	published
Security Proxy	yes	no	XIFI	UPM-DIT	published	published
SLA Manager	yes	yes	XIFI	ATOS, SYNELIXIS	published	published
<p>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:</p> <p>Master node: yes, All nodes: no – The component under consideration must be present on the master node. 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 45: 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
Deplo viment	Infrastructure Toolbox	Sub-system	Sub-system	Delegate to component maintainer or to other sub-system maintainer, or escalate to infrastructure maintainer

Sub-system		First Level Maintenance contact ^{Note1}	Responsible Maintainer ^{Note2}	Possible next action ^{Note3}
	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 GE	Federation	Sub-system	Delegate to component maintainer or 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 46: 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 responds 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 41: 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.

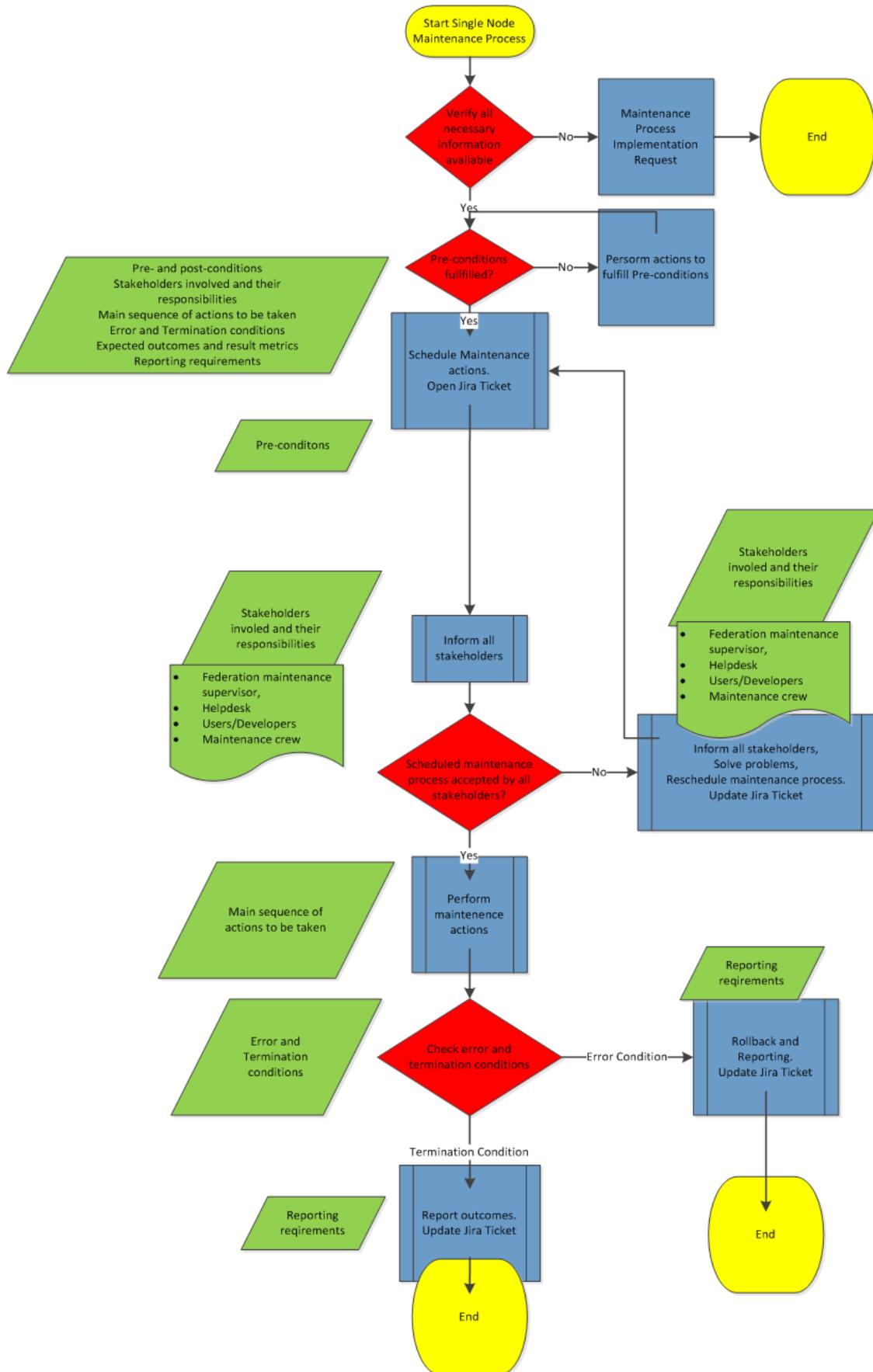


Figure 42: Outline of a scheduled maintenance procedure affecting a single infrastructure node

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.

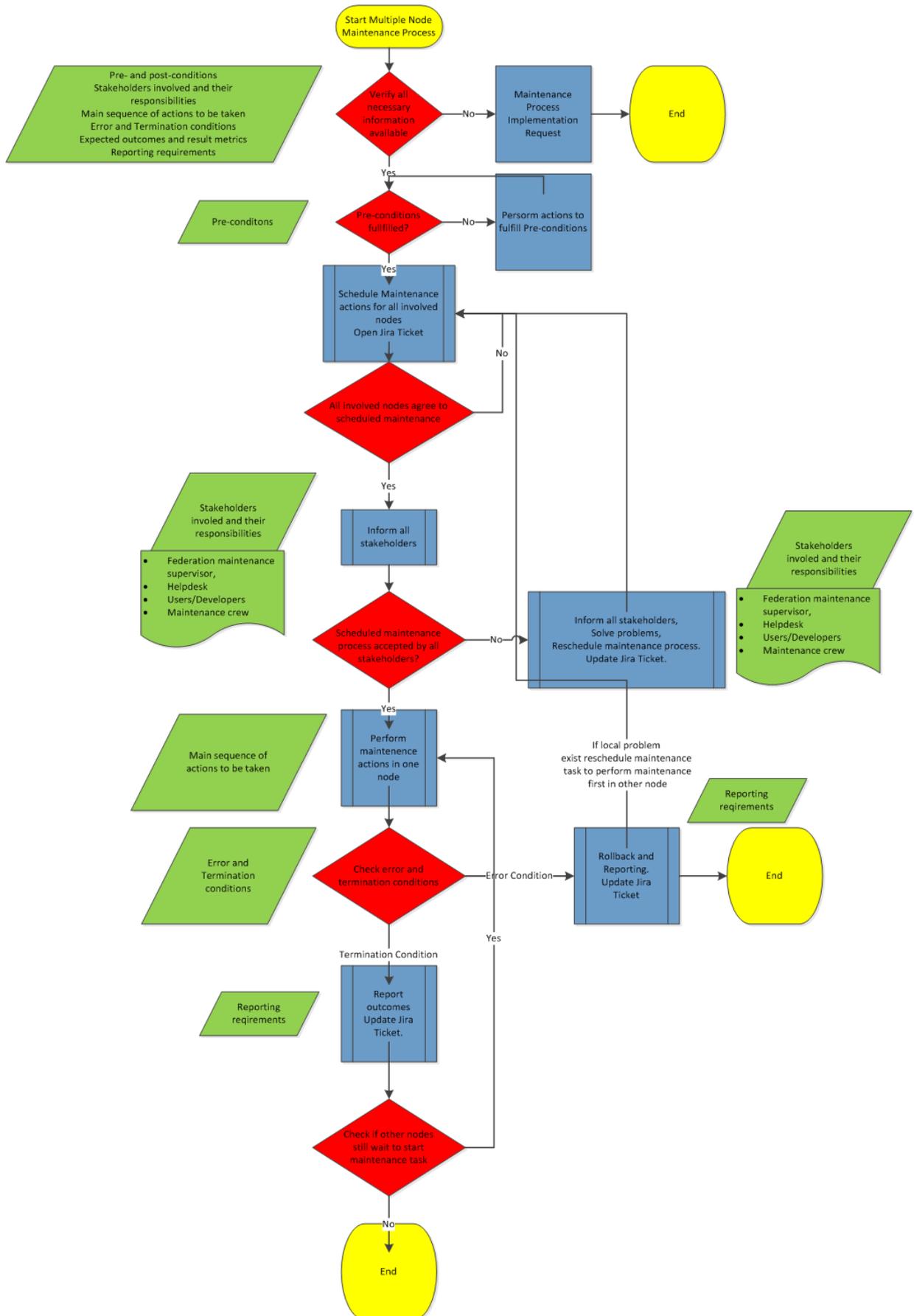


Figure 43: Outline of scheduled federation-wide maintenance procedure affecting multiple infrastructure nodes

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.

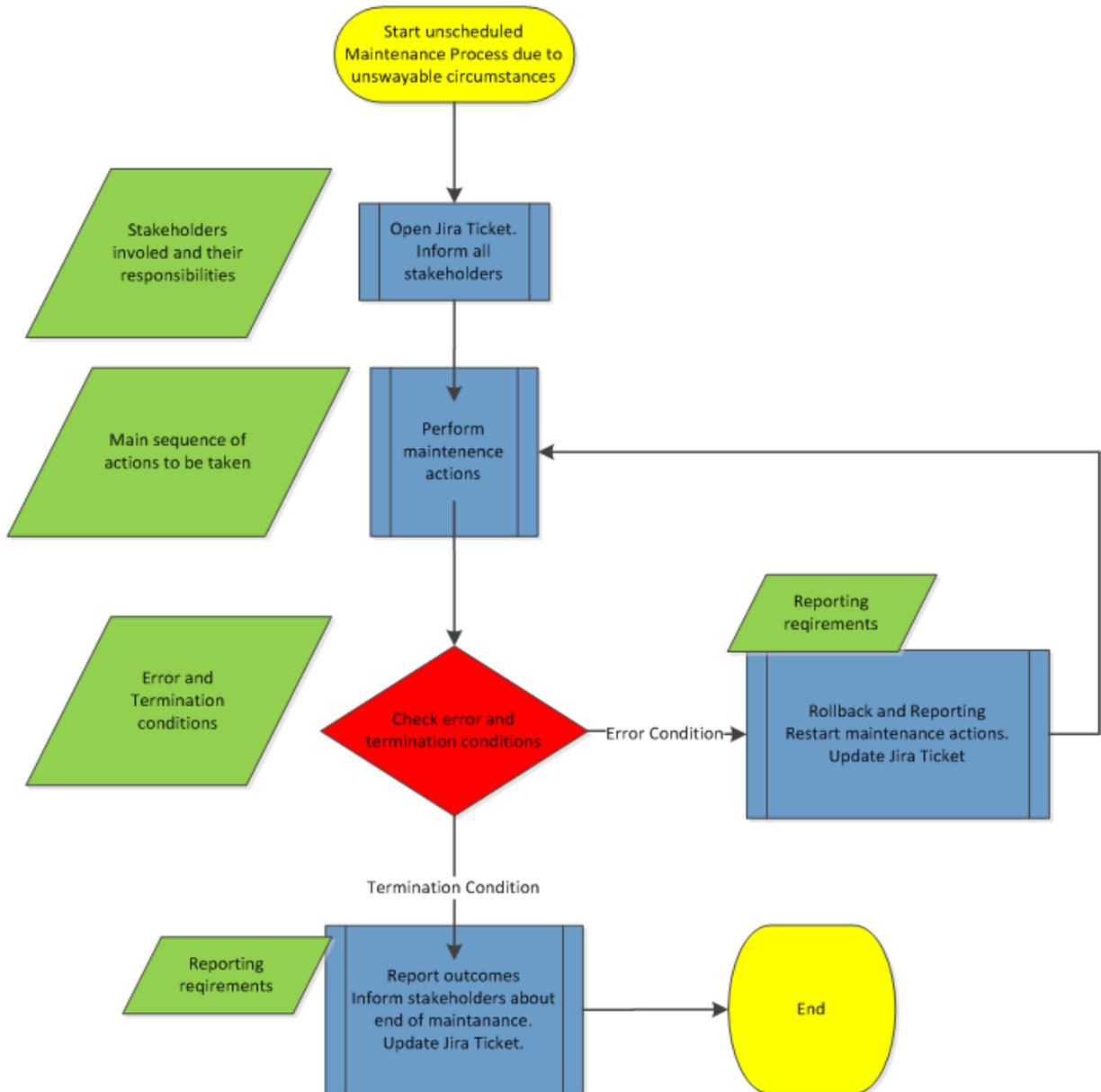


Figure 44: Outline of an unscheduled maintenance procedure affecting a single infrastructure node

6 SUPPORT TO FI-DEVELOPERS

Section 6 defines the process of providing support to FI-developers. While the above section 5 that described 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 the 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 between support and maintenance definition that was thought of should be briefly clarified here.

6.1 Introduction to support levels

In this introduction we summarise the general concepts related to helpdesk Levels from 0 to 3. These definitions will be used 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 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 FI- developer support process and flows

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 / Level 3 support process and flows.

- Level 1 support is provided by a helpdesk team. The team is in charge of filtering incoming tickets, managing all requests, issues and problems coming from FI-Developers. An issue is passed from Level 1 to Level 2 or Level 3 helpdesk if it is related to:
 - SW/HW problem on a specific node: Level 1 passes the issue to the specific Level 2 team of the node (in case this was not automatically done by default).

- Software component problem related to a GE / GEi: Level 1 passes the issue to the respective Level 3 support.
- 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. There is one specific Level 2 group at each node. Level 3 type support in the Node Help Desk – in the sense of above definition (section 6.1) – is provided by system or network administrators if need be, in case of more complex issues. In XIFI we do however not formally distinguish between Level 2 and 3 for the node helpdesk. The helpdesk team takes care of all node-related issues of any complexity level.
- 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).

Referring to above definitions of helpdesk levels (section 6.1), at the current state of helpdesk process implementation, Level 0 helpdesk is currently not in place yet. However, as soon as an FAQ list of questions and related answers has been prepared this will be a first element providing Level-0-type of support. It should also 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 FIWARE Lab Level 1 helpdesk

In the following flow chart a scenario is depicted in which Level 1 FIWARE Lab Helpdesk is contacted through the FIWARE Lab support email address. Sending a mail to this mailing list will automatically create a related Jira Ticket, assigned to Level 1 helpdesk. Level 1 helpdesk is then doing a number of checks as a first step:

- 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?)
- Is the issue related to a known problem, already described and answered in the FAQs or by a related existent Jira ticket?
- Addresses the issue a non-technical aspect, e.g. of administrative or legal nature for which e.g. the Federation Office is in charge of?

In the example scenario in Figure 45 we identify the correct interaction between FIWARE Lab Helpdesk structure and FI-Developer. There are two potential scenarios:

- There is already a Jira ticket: Level 1 helpdesk will reply to the user's email address feedback of the resolution of the issue, once resolved.
- The issue is new, i.e. not addressed in the FAQ and not covered by another ongoing ticket.

Level 1 helpdesk will forwarded (assign) the ticket to Level 2 or 3 only after checking that the issue cannot be solved directly, and after identifying what the specific support team is (e.g. which of the nodes?) that should deal with the issue.

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 45 below depicts the scenario in high level.

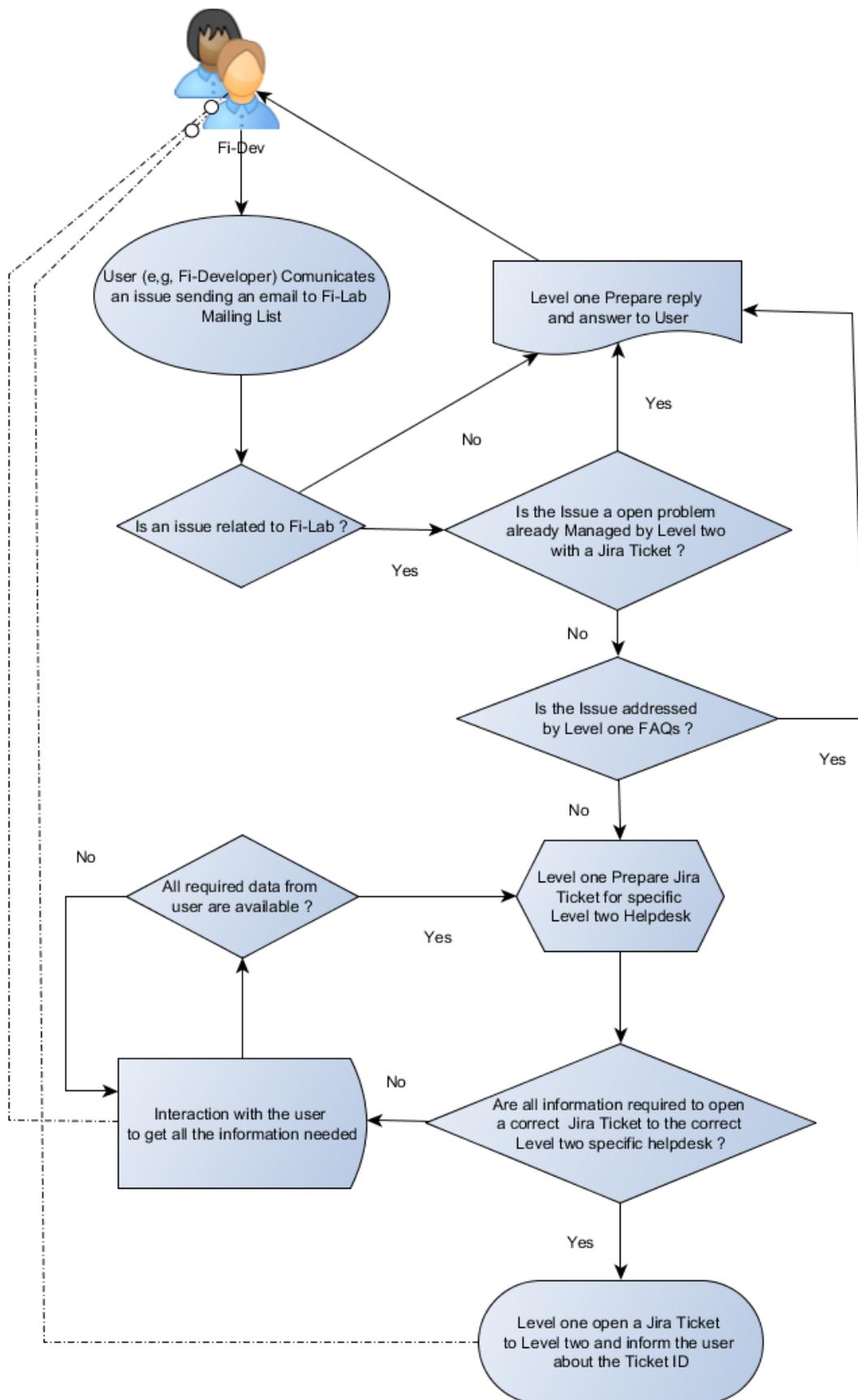


Figure 45: FIWARE Lab Level 1 support scenario for the case triggered by email from FI-developer

6.4 FIWARE Lab Level 2 / 3 support

As said above, Level 2 support is provided by the node helpdesks of all nodes. In the scenario here the FI-developer does not request support in email format, as in the previous scenario, but by filling an issue collector form. In the issue collector form the user (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 to 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.

In case the issue is related to a GE / GEi component (not depicted in below figure), and if the developer has indicated the GE in the collector form, then the ticket is for Level 3 support and directly assigned to the GE / GEi owner in analogy to Level 2 support.

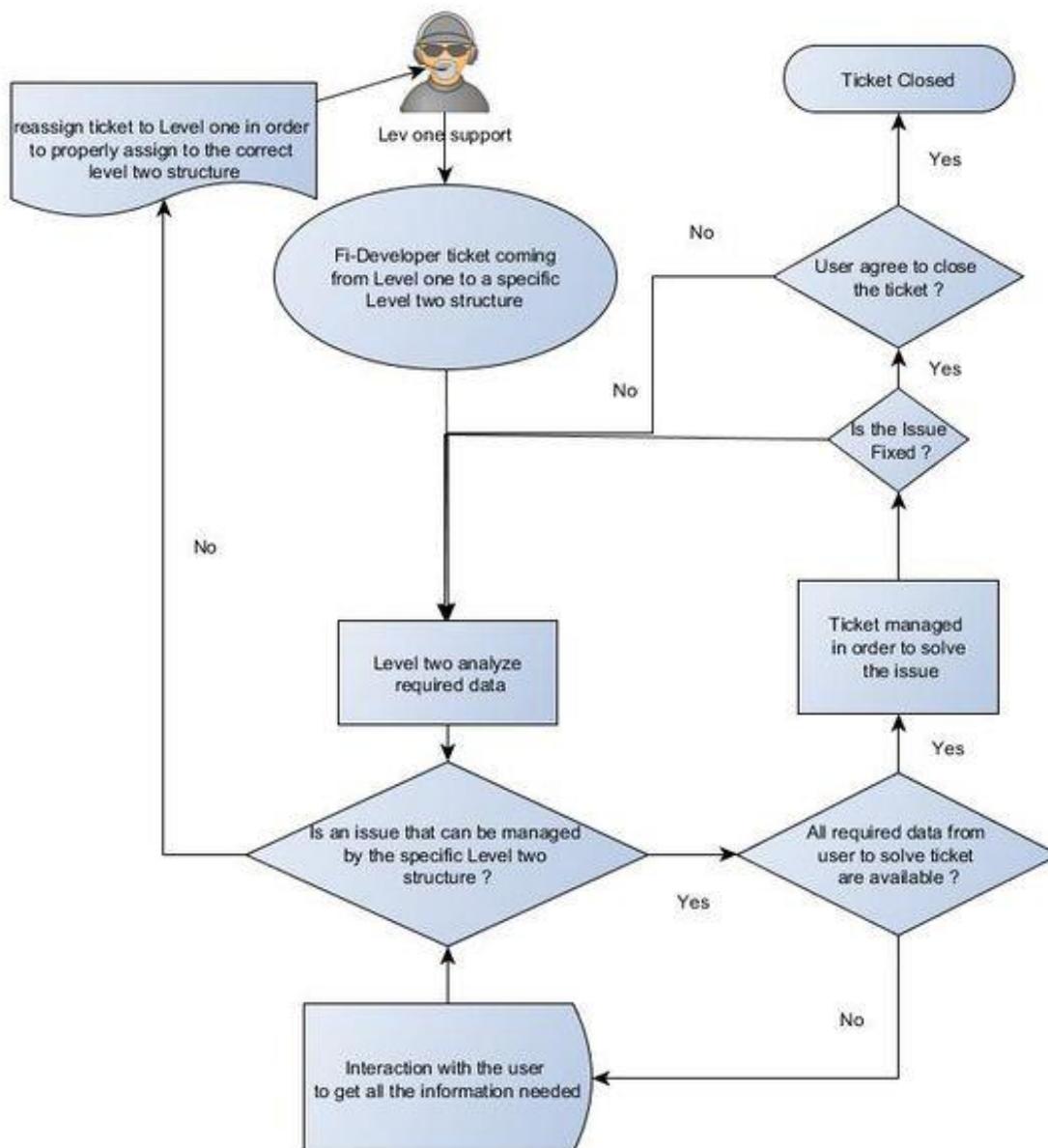


Figure 46: FIWARE Lab Level 2/3 support scenario, triggered by JIRA collector form

6.5 JIRA ticketing process, flows and responsibilities

As said before, XIFI has chosen to use JIRA as ticketing system for FIWARE Lab jointly with FIWARE / FI-CORE.

Users, i.e. FI-developers, will be given two options (formats) to submit a support request. The preferred form 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 in the form enables the assignment of the created ticket directly to the proper support team best suited to address the issue, i.e. to the respective node location or GE owner respectively. The second option, alternatively to the collector, is to send an email to a support email address. Emails sent to this address are converted into a JIRA ticket by use of a dedicated JIRA plugin.

JIRA collectors will be placed in 3 places: i) On the FIWARE Lab support page, ii) on the FIWARE Lab nodes status page, and iii) in the FIWARE catalogue on the page of every GE. The email contact option will be provided only on the FIWARE Lab support page from the “Need help?” section.

Queries of general nature that are not FIWARE Lab related will be dealt with by FIWARE / FI-CORE. 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.

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	Details are listed in section 3.1 “Management of nodes” on page 34.
GEi	GE / GEi owners are identified in a list maintained by FIWARE / FI-Core project, see the list at [21]
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 (FIWARE partner, for support on Linux level)
JIRA support (Application Admin Level)	Florian Rommel (EURESCOM), Manuel Escriche (TID)
Liaison with FIWARE Lab Portal	Javier Cerviño Arriba (UPM), Fernando López Aguilar (TID)

Table 47: Responsibility Assignment

Full name	Organisation
Miguel Carrillo (Lead)	TID
Florian Rommel	EURES
Marco Cipriani	TI
Fernando López	TID
Daniele Santoro	CREATE-NET
Aristi Galani	UPRC
Sándor Laki	Wigner

Table 48: Level 1 Helpdesk team

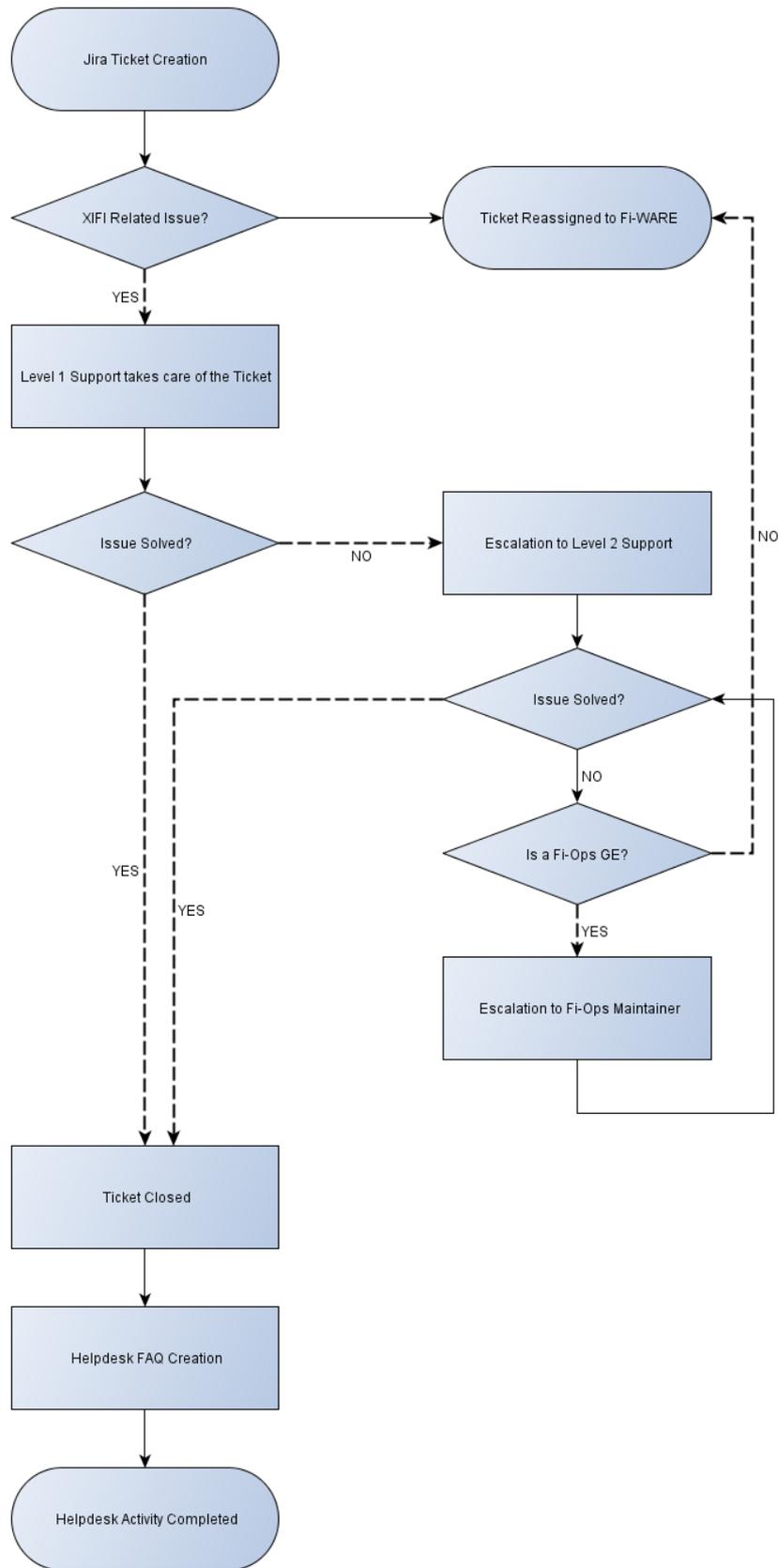


Figure 47: JIRA ticket flow example

6.6 JIRA Administration

The JIRA platform will be hosted in one of the servers of the FIWARE Lab infrastructure, currently in the Spain node. TID is responsible of the installation and covers the licence costs. Costs associated with plugins for email and notification handling are relatively low (a few hundred Euro) and TID covers also these. CREATE-NET also offered to take over these costs and this is kept as the backup option, as it is easier from an administrative perspective if TID as the owner of the JIRA instance from Atlassian's takes care of that.

JIRA support will be offered in Best Effort schema. Moreover, everybody will receive the same treatment; i.e. there is not VIP treatment, at least from the perspective from the XIFI project.

The following functions will need to be performed related to JIRA administration. Responsibilities for that are assigned as indicated in Table 47:

- General JIRA Administration.
- User account management: creation, modification, permissions and elimination of JIRA users
 - Note that users will only be created for the matter of support levels 1, 2 and 3. External FI-developers will not get a JIRA account. User will be added with care, given the related license costs to be paid to Atlassian for each additional user. Once created, in practice users cannot be deleted.
- User support for JIRA specific issues.
- Tracker Management: creation, modification, administration and deletion of JIRA projects.
- Creation of issue collectors associated to each federation node, GE and FIWARE Ops component.
- Plugin and or Gadget management in order to improve the JIRA user experience.
- Preparation of reports.
- The email server for notifications (to be managed by TID).
 - Inbox for incoming messages for creation of tickets.
 - Outbox for notifications.
 - Note that this mailbox will be one of the addresses of the support list (it is NOT the mailman support list)
- The mailman support channel (fiware-lab-help@lists.fi-ware.org) will be administrated by TID. Emails sent to this list will be converted into JIRA tickets.

6.7 Reporting (JIRA statistics)

Reporting of FIWARE Lab related JIRA statistics will be done. There will be single and uniform reports distributed to users in all related projects, e.g. XIFI and FI-WARE / FI-CORE, avoiding the need of preparing individual, project specific customised reports. The responsible persons are indicated in Table 47. Details on reporting have been discussed but still need to be finally agreed, including:

- Metrics
- Periodicity (probably monthly)
- Granularity
- Format (Excel may be user-friendly – depending on what JIRA offers)

6.8 FAQ and beginners guide

Currently this topic is under discussion internally. Three technical options were identified that could be considered for implementing the FAQ pages:

- Stack Overflow [17] is a Q&A site for programmers. It is built and run by you as part of the Stack Exchange network of Q&A sites. FI-WARE had started to use it in order to give support to some of the (TID, UPM) GEIs. Example of tags in use are filab, fiware, fiware-wirecloud, fiware-cosmos, cosmos, ckan.
- Ask OpenStack is the Q&A site of the OpenStack community where you can ask and obtain answer related to any OpenStack service.
- We are evaluating the possibility to offer a Q&A site associated to WordPress in the Cloud Portal, but it should be confirmed.

7 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 so far, 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 agreements and procedures are still work in progress. Protocols and procedures have evolved since the beginning of the project and will likely continue to do so in the remainder of XIFI project, fed by experience gained from adding further (heterogeneous) nodes and from operating and maintaining a growing federation getting more complex. Also the growing number of users who rely on a secure and highly available platform and their needs in terms of support with guaranteed SLAs will provide new insight and a respective refinement of the procedures. Thus, the definitions and specifications contained in this document will further evolve. They will be kept up-to-date on the XIFI Wiki as they evolve and serve there 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 D5.4: XIFI federation extension and support
- [7] XIFI Deliverable D6.2: XIFI Showcases Demonstrators v1
- [8] XIFI Deliverable D9.2b: XIFI Office – Description and Establishment
- [9] XIFI Wiki: <http://wiki.fi-xifi.eu/>
- [10] "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.
- [11] Wikipedia contributors, "MyExperiment," Wikipedia, The Free Encyclopedia, <http://en.wikipedia.org/w/index.php?title=MyExperiment&oldid=595344437> (accessed July 23, 2014).
- [12] 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
- [13] Sally Floyd maintained the RED resource at <http://icir.org/floyd/red.html>
- [14] Checklist SLA OLA according to ITIL 2011 Service Design, http://wiki.en.it-processmaps.com/index.php/Checklist_SLA_OLA
- [15] XIFI stakeholder definitions, <https://www.fi-xifi.eu/about-xifi/stakeholders.html>
- [16] 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/>
- [17] Stackoverflow, question and answer site, <http://stackoverflow.com>
- [18] Ask OpenStack: Q&A site of the OpenStack community: <https://ask.openstack.org/en/questions/>
- [19] Glance added property NID description <http://docs.openstack.org/image-guide/content/image-metadata.html>
- [20] Guestfish: http://docs.openstack.org/image-guide/content/ch_modifying_images.html
- [21] GE / GEi owners list (maintained by FI-WARE project): <https://docs.google.com/spreadsheets/cc?key=0AqLSWp0KXaaDdEpLT0RmXzhzbXEtSTVvSE5wX3oyWXc#gid=0>

Appendix A Further details on OLA

A.1 OLA Scheme Description

The following presentation and discussion of the service categories for infrastructure operators is oriented along the ITIL SLA/OLA template [14] 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 project⁹ 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 important to rank the criticality of services and incidents which allows defining countermeasures and reaction times that respect the

⁹ 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.

A.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.

A.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.

A.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.

A.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.

A.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

A.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 B 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>
```

B.1.1 Appendix level 3

B.1.1.1 Appendix level 4

B.1.1.1.1 Appendix level 5

[end of document]