

Clara Pezuela – Atos
Málaga, 15 December 2016

Open APIs
for Open
Minds

Why should I trust in FIWARE?

Quality Assurance in FIWARE

Table of contents

- The motivation: why QA is needed in FIWARE?
- The purpose: what does it mean Quality Assurance in FIWARE?
- The team: who was doing the QA?
- The methodology: how was the testing process?
- The tests: what was tested and which were the obtained results?
- The assessment: what the conclusions of performed tests?
- The labelling: how to rapidly check the GEs quality?
- The publication: how can I find the QA work and results?
- The future: how QA task will evolve?
- Why should I trust in FIWARE?
- Q&A (10')

The motivation



- FIWARE is rapidly moving from experimental to production environments in which the platform must scale up in reliable and real workload conditions
 - FIWARE GEs must work at an adequate quality, reliability and performance level
- Support FIWARE users with high-quality support for installation, configuration and operation of FIWARE technology
 - Improving the FIWARE user experience in general
- Practical approach with focus on improving quality and transparency
 - Light and agile methodology very operative

The purpose (I)



- **To test FIWARE GEs**, addressing functional and non-functional testing

The purpose (II)



- **To test FIWARE GEs**, addressing functional testing
- **Curation of GEs documentation** (functional testing):
 - to check the completeness, consistency, soundness and usability of documentation for specification, implementation and installation of the Ges
 - to test the training efficiency by analyzing the Academy courses
- **Verification of the GE specification** (functional testing):
 - developing the appropriate test cases to assess if the GEs implementation corresponds to what is defined in the specification.
 - validating the APIs
 - integration tests of common bundles

The purpose (III)



- To test **FIWARE GEs**, addressing non-functional testing
- **Assessment of performance, stability and scalability** (non-functional testing):
 - Defining and executing test scenarios to find the limits of a GE
 - Comparing with reference values of usual execution
 - Replicating environments of intensive workload

The QA team

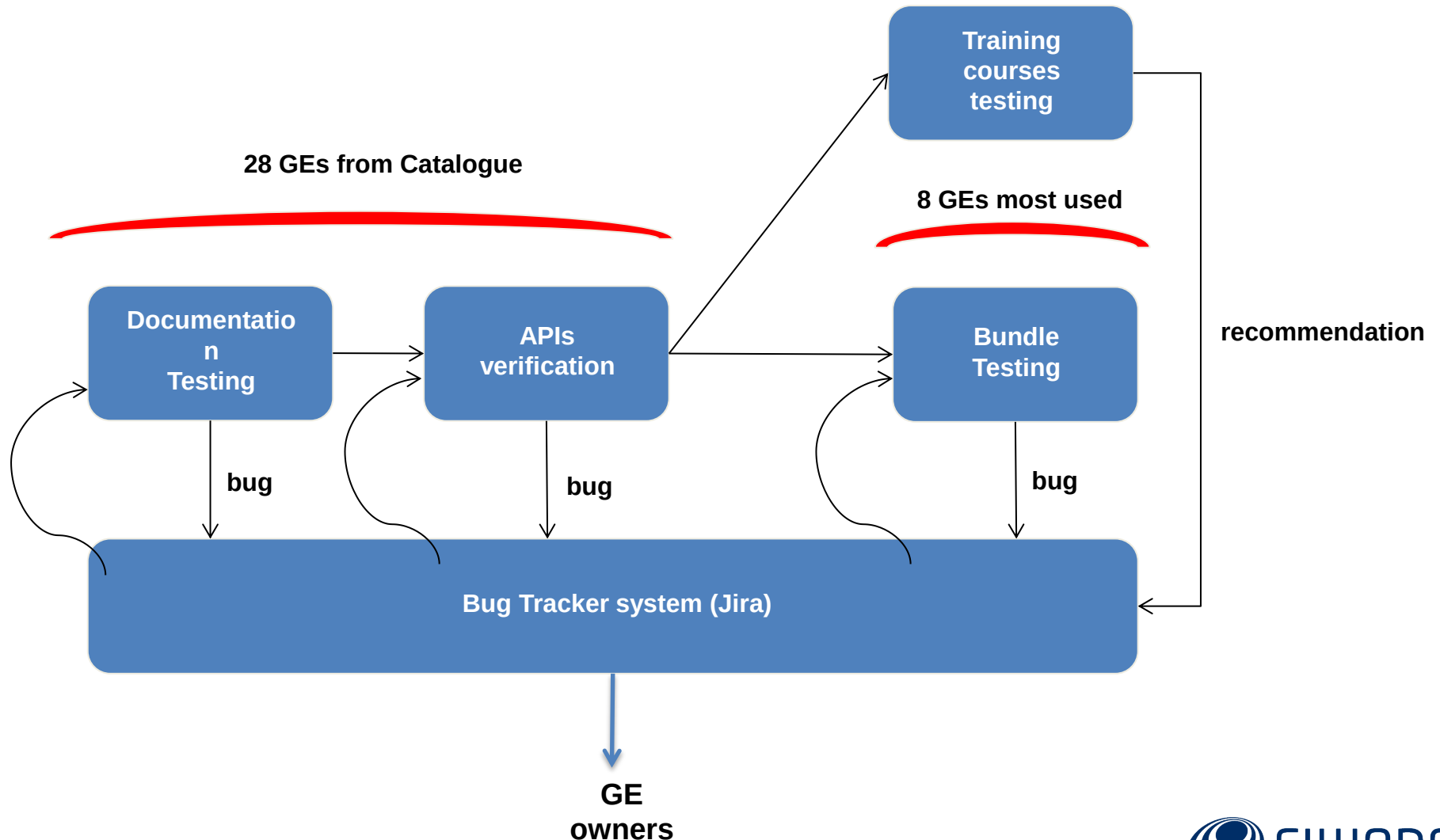
- Professional units of testing in Atos and Engineering
- Research groups specialized in testing methods (EGM, Fraunhofer, Grassroot)



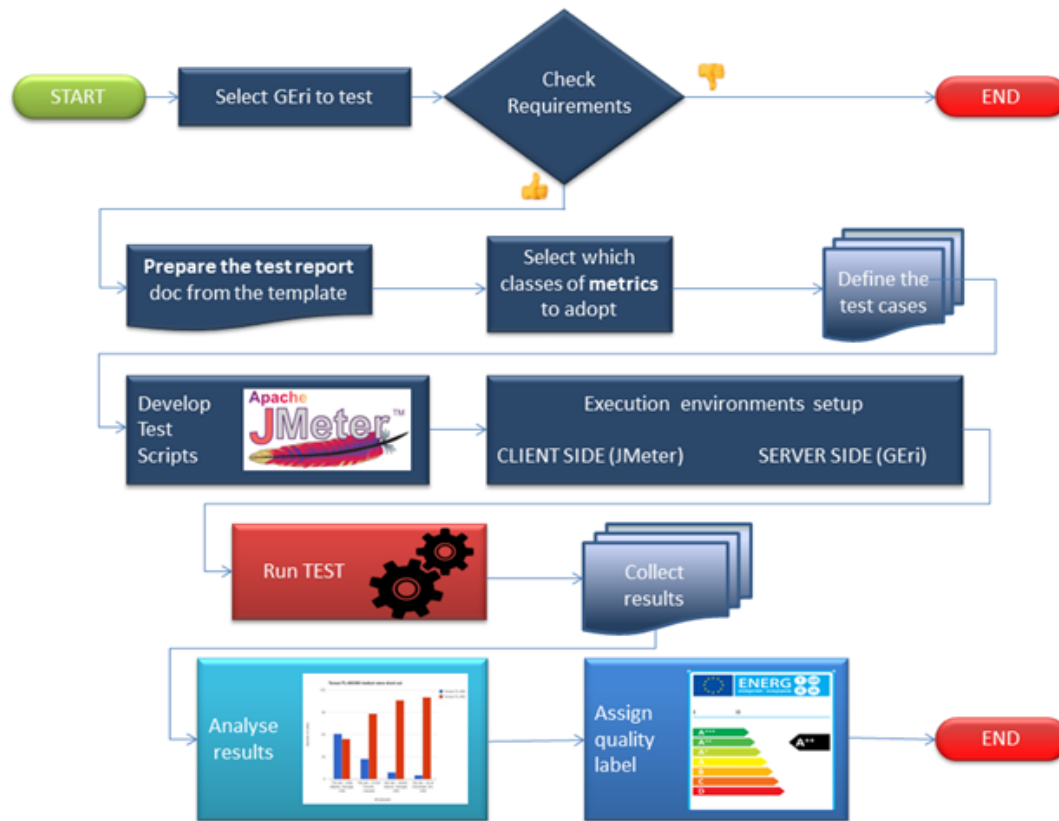
- Clara Pezuela, Atos, task coordinator
 - Miguel A. Ramirez, Atos, non-functional testing
- Carlos Lucena, Atos, non-functional testing
 - Riccardo Zanetti, ENG, non-functional testing
- Andrea La Porta, ENG, functional testing
 - Annamaria Cappa, ENG, functional testing
- Carmen Mac Williams, GAR, research testing
 - Franck Le Gall, EGM, research on testing
- Peter Murynshkin, Fraunhofer, research on testing



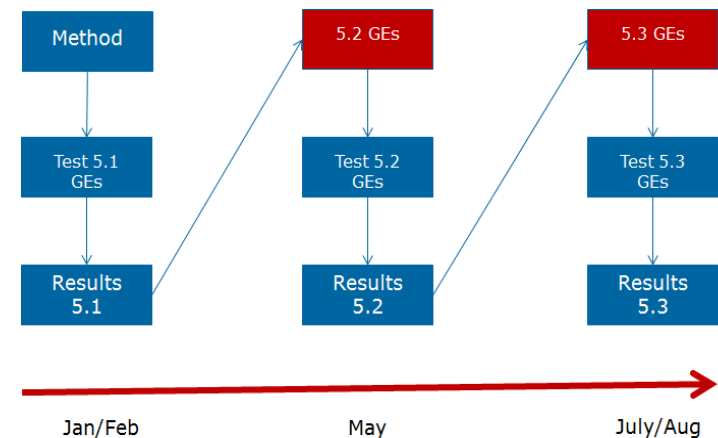
The methodology: for functional testing



The methodology: for stress testing

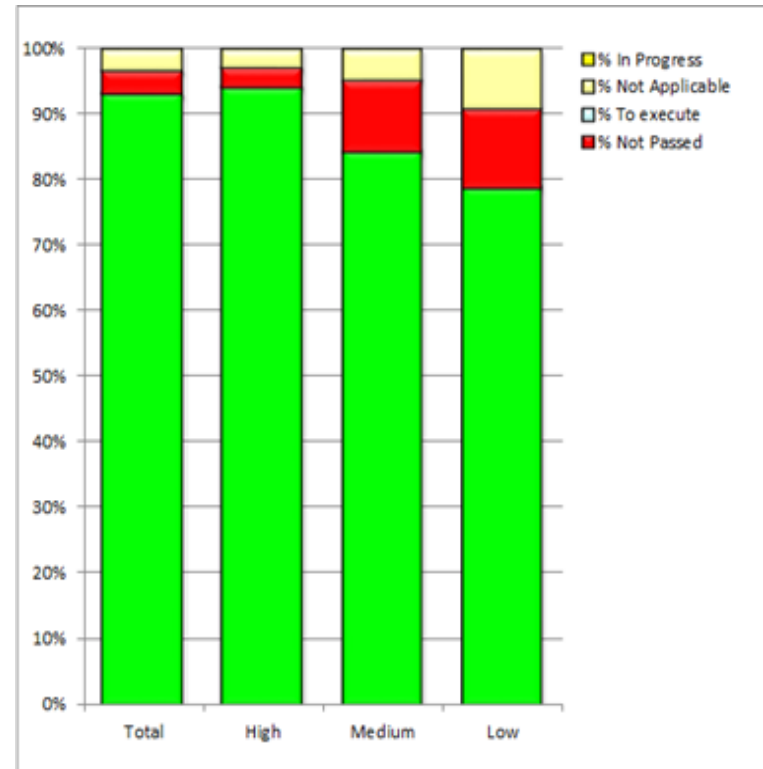


Identity Management – **KeyRock**
 Authorization PDP – **AuthZForce**
 PEP Proxy – **Wilma**
 Context Broker - **Orion**
 Complex Event Processing (CEP) – **Proton**
 Stream Oriented – **Kurento**
 Backend Device Management – **IDAS**
 IoT Broker – **Aeron**
 IoT Data Edge Consolidation – **Cepheus**
 Policy Manager - **Bosun**



The results: curation of documentation and verification

- Live dashboard collects and maintains the assessment information
- Near **95% of the high priority GEs** has passed **successfully** the documentation and verification tests.
- The **medium and low priority GEs** are around **80% of success** but they are working on solving the issues.
- In **average**, more than **90%** of GEs have passed the tests



The results: training courses

- 32 courses were tested
- 6 were **Good** (18%)
- 8 were **To improve** (25%)
- 18 were **Sufficient** (56%)

Chapter	Course ID	Status	Jira Ref.	Score
Security				
Identity Management	79	Completed	SEC-973	Good
PEP Proxy	131	Completed	SEC-974	Good
Authorization PDP	144	Completed	SEC-975	To Improve
Access Control (OAUTH-API-AZ)	57	Completed	SEC-985	To Improve
Applications/Services Ecosystem and Delivery Framework				
Apps and Services Overview	52	Completed		Sufficient
DataVisualization	141	Completed	APP-1180	Sufficient
Application Mashup	53	Completed	APP-1181	Good
Marketplace	21	Completed	APP-1182	Sufficient
Repository	127	Completed	APP-1183	Sufficient
Revenue Settlement and Sharing System	117	Completed	APP-1184	To Improve
Store	104	Completed	APP-1185	Sufficient
Data/Context Management				
Context Broker	132	Completed	DATA-1655	Good
Context Broker (2)	44	Completed	DATA-1677	Sufficient
Context Broker (5)	149	Completed		Sufficient
Big Data	69	Completed	DATA-1656	Sufficient
Complex Event Processing	58	Completed	DATA-1674	Good
Stream Oriented	62	Completed	DATA-1675	Good
Short Term Historic Open Data Repository (CKAN)	145	Completed	DATA-1676	To Improve
Interface to Networks and Devices (I2ND)				
Network Information And Control (OFNIC)	72	Completed	MIND-628	To Improve
Advanced Middleware (Kiara)	140	Completed	MIND-627	To Improve
Internet of Things (IoT) Services Enablement				
Backend Device Management (IDAS)	128	Completed	IOT-897	Sufficient
IoT Broker	33	Completed	IOT-898	Sufficient
IoT Data Edge Consolidation	36	Completed	IOT-899	Sufficient
IoT Discovery	40	Completed	IOT-900	Sufficient
Advanced WebUI				
3D-UI-XML3D	97	Completed	WEB-1265	To Improve
Cloud Rendering	92	Completed	WEB-1266	Sufficient
GIS Data Provider	88	Completed	WEB-1267	To Improve
Interface Designer	91	Completed	WEB-1268	Sufficient
POI Data Provider	96	Completed	WEB-1269	Sufficient
Synchronization	111	Completed	WEB-1270	Sufficient
Virtual Characters	112	Completed	WEB-1271	Sufficient
Cloud Hosting				
Policy Manager	119	Completed	CLD-1405	Sufficient

Data obtained in September 2016

The results: stress testing (GEs)

GE name	Reliability (errors rate)	Performance in stress condition (num requests, response time, bit rate)	Stability (crashes)
IDAS	0%	140 updates /second, 200 threads	No crash
IoT Broker	44% for SubscribeContext	30 requests/s generated by 16 concurrent threads with an average response time of 468 ms	Crashed
AuthZForce	0%	AV RT around 11 ms; 4376 requests per second	No crash
KeyRock	0 %	Authorisation max load with AV RT<1s= 220 requests/s Authentication max load with AV RT<1s=22 requests/s	No crash
Wilma	0%	up to 839 requests/sec	No crash
Orion	0%	5160 attribute updates/sec	Crashed
Proton	0,39%	500 requests/sec; 950 creating definitions/sec	Crashed
Bosun	3,5%	160 simultaneous threads , 26 HTTP responses/sec for Cloto and 30 for Facts	Crashed
Kurento	0%	Good for less than 50 simultaneuos users and low quality video	No crash
Cepheus	42% for Broker Test	100 requests/sec	Crashed

The results: stress testing (bundles)

Bundle	Reliability (errors rate)	Performance in stress condition (num requests, response time)	Stability (crashes)
AuthZForce+Wilma+KeyRock	0%	99 requests/second	No crash
Orion+IDAS+Cygnus	0%	280 requests/second/ 300 Threads	No crash

The results: research activities

- Model based testing (MBT) for generating automating tests to verify the compliance of FIWARE GEs APIs with NGSI standard

GE	Executed tests	Failed test	NGSI compliance
Orion	30	3 (10%)	High
IoTBroker	30	9 (30%)	Medium-high
Cepheus	30	21 (70%)	Light

- Continuous integration testing (some trials)
- Automatic online documentation testing (Catalogue)
 - Require two sets of metrics:
 - derived from the Compliance Guide Catalogue (optimal text length, context relevant concepts,...)
 - common language metrics to ensure good style and legibility of online documentation (average sentence length, lexical complexity,...)

The assessment (I): overall work

- Hard starting...
 - Pressure in setting up the task very fast for public events
 - Business teams never worked before together
 - Heterogeneity of the GEs, different testing environments settings
 - Lack of reference values and test cases in some GEs
- But progressively improving...
 - GE owners collaboration and responsiveness increased
 - Better obtained results in consecutive releases
 - Homogeneity in reporting was increased
 - Joint labelling process for functional and non-functional testing

The assessment (II): GEs quality status

■ Documentation testing

- Most of GEs manual allows installing the components
- But the documentation is not always clear, readily and available from the links.

■ APIs testing

- The installed software package implements the APIs declared into Open Specification.
- The main failures concern the missing information on documentation.

■ Bundle Integration tests

- Simulation of a parking sensors scenario
- Using Orion, Wilma, CEP, IDAS, AuthZForce, SpagoBI, KeyRock
- 4 out of 22 tested cases failed

■ Academy courses testing

- 60% of the academy courses are sufficient
- Many tips were reported on Jira tool to help the GE owner improving the training.

The assessment (III): GEs quality status



- In scalability, all GEs behave very good except one
- In performance, 4 out of 10 GEs are providing values that could be improved in next releases
- In stability, half of the GEs are not managing properly the memory or CPU resources

The labelling (I)

- Quick at a glance mechanism to check the assessed GEs' quality
- Following the EU energy label system
- Sub-label per each tested aspect (usability, reliability, efficiency, scalability, performance and stability)
- Global label as average of all sub-labels
- Initial pilot with 10 GEs



FIWARE Catalogue

Home Enablers Bundles Tools IoT Ready

Data/Context Management

Stream-oriented - Kurento

Powerful software stack devoted to simplify the creation of complex interactive multimedia applications by exposing a rich family of APIs on top of a J2EE application server.

qa **A+**

FIWARE GEs

Data/Context Management

Publish/Subscribe Context Broker - Orion Context

Orion Context Broker is an implementation of NGS19 and NGSI10 with persist based in MongoDB.

qa **A++**

FIWARE GEs

Data/Context Management

BigData Analysis - Cosmos

Monitoring and control of the BigData Analysis GE

Complex Event Processing (CEP) - Proactive

Complex Event Processing GE

criteria	value	label
Documentation completeness	Excellent	qa A+++
Documentation soundness	Very Good	qa A++
APIs Failure Rate	0 tests failed/executed	qa A+++
Detected defects by Priority	0 average bugs priority	qa A+++
Time to respond issues	< 1 day	qa A+++
Time to fix issues	< 1 day	qa A+++
Scalability	1,02 response time/thread number	qa A+++
Performance	1900 updates/sec	qa A+++
Stability	Memory leak avoidable by configuration	qa B

The overall label is the average of all individual labels assessed by Sep 2016

Identity Management – **KeyRock**
 Authorization PDP – **AuthZForce**
 PEP Proxy – **Wilma**
 Context Broker - **Orion**
 Complex Event Processing (CEP) – **Proton**
 Stream Oriented – **Kurento**
 Backend Device Management – **IDAS**
 IoT Broker – **Aeron**
 IoT Data Edge Consolidation – **Cepheus**
 Policy Manager - **Bosun**

Labels assigned in September 2016

The labelling: meaning of values

how complete is the GE documentation?

how many GE APIs were failed over the tested ones?

how long does GE take to answer the issues?

how the response time is growing as long as number of concurrent threads is growing?

how well are memory/CPU managed?

criteria	value	label
Documentation completeness	Very good	qa A++
Documentation soundness	Very good	qa A++
APIs Failure Rate	0 tests failed/executed	qa A+++
Detected defects by Priority	0 average bugs priority	qa A+++
Time to respond issues	< 1 day	qa A+++
Time to fix issues	5,75 days	qa A+
Scalability	1,65 response time/thread number	qa B
Performance	140 Kbps * user	qa C
Stability	Memory/CPU are slightly increasing but no leak	qa A+
The overall label is the average of all individual labels assessed by Sep 2016		

how sound is the GE documentation?

how many bugs in average by priority have been detected?

how long does GE take to fix the issues?

Depend on GE type:

- how many updated attributes/second?
- how many complex events/second?
- which bit rate per user is acceptable?
- how many requests/second?

The labelling (II): meaning of labels

- Qualitative and quantitative values (ranges)
- Same metrics for all GEs, except for performance, which depends on type of app

Completeness		
Label	Value	Formula
A+++	Excellent	Each expected document is available. The information are exhaustive, easily accessible and easy to use. There are examples, comments or other utilities that improve the reading/comprehension.
A++	Very good	Each expected document is available. The information are exhaustive, easily accessible and easy to use.
A+	Good	Each expected document is available. The information are fully exhaustive.
A	Fair	Each expected document is available and en
B	Poor	Each expected document is available but the

Growing Response Time ratio/ Growing thread number ratio

Label	Value
A+++	< 1.05
A++	1.22 - 1.05
A+	1.42 - 1.21
B	1.71 - 1.41
C	2.11 - 1.71
D	2.61 - 2.11
E	> 2.61

(Bit Rate * users number) ratio

Label	Value
A+++	> 4'5 Mbps
A++	2'01 - 4'5 Mbps
A+	1'01 - 2 Mbps
A	0'501 Mbps - 1 Mbps
B	251 Kbps - 500 Kbps
C	100 Kbps - 250 Kbps
D	< 100 Kbps

Stability	
Label	Value
A+++	Memory/CPU keep stable
A++	Memory/CPU are very slightly increasing but no leak
A+	Memory/CPU are slightly increasing but no leak
A	Memory/CPU are progressively increasing but no leak
B	Memory leak avoidable by configuration
C	Memory leak not avoidable. The system crashes after a few hours.
D	High leak. System crashes before half an hour

The labelling (III): resulting labels

GE	Overall value	FUR Overall	NFR Overall
Orion	A++	A+++	A++
KeyRock	A+	A+	A+
AuthZForce	A++	A++	A+
Wilma	A++	A++	A++
Proton	A++	A++	A+
Kurento	A+	A++	B
IDAS	A	B	A+
lot Broker	A	A+	A
Cepheus	A++	A+++	A++
Bosun	A++	A++	A++

GE	Overall value (by average approach)	NFR Overall	NFR (warranty) Scalability	NFR (warranty) Performance	NFR (warranty) Stability
			Growing response	Attribute update per	Stability
Orion	A++	A++	A+++	A+++	B
KeyRock	A+	A+	A+++	B	A
AuthZForce	A++	A+	A++	A++	A
Wilma	A++	A++	A+++	A+	A+++
Proton	A++	A+	A+++	A+++	C
Kurento	A+	B	B	C	A+
IDAS	A	A+	A+++	B	A++
lot Broker	A	A	A++	B	A
Cepheus	A++	A++	A+++	A+++	B
Bosun	A++	A++	A+++	A+++	B

GE	Overall value (by average approach)	FUR Overall	FUR (utility) Usability	FUR (utility) Usability	FUR (utility) Reliability	FUR (utility) Reliability	FUR (utility) Efficiency	FUR (utility) Efficiency
			Completeness	Soundness	Failure Rate	Defects by Priority	Time to Taking charge	Time to Fix
Orion	A++	A+++	A+++	A++	A+++	A+++	A+++	A+++
KeyRock	A+	A+	A+	A+	A+++	A+	B	A++
AuthZForce	A++	A++	A++	A++	A+++	A+++	A	A
Wilma	A++	A++	A++	A++	A+++	A+++	A++	A
Proton	A++	A++	A+	A+	A+++	A+++	A++	A
Kurento	A+	A++	A++	A++	A+++	A+++	A+++	A+
IDAS	A	B	B	B	A+	A+	D	D
lot Broker	A	A+	A+	A+	A++	A	A+	C
Cepheus	A++	A+++	A+++	A++	A+++	A+++	A+++	A+++
Bosun	A++	A++	A	A+	A++	A++	A+++	A++

The publication

- Public document:
https://www.fiware.org/wp-content/uploads/2016/10/QA_public_document.pdf
- QA wiki page:
https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE_QA_Activities
- Brochure
- Blog post:
<https://www.fiware.org/2016/09/20/assessing-fiware-ges-quality/>

The screenshot displays the FIWARE website's 'QA Activities' page. At the top, the European Commission logo is visible. The page features a navigation bar with links for 'page', 'discussion', 'view source', and 'history'. The main content area is titled 'FIWARE QA Activities' and includes a paragraph about the Quality Assurance chapter's purpose. Below this, a 'Contents' section lists '1 Functional Testing'. A sidebar on the left contains a 'navigation' menu with 'Main page' and 'Quick Tour'. A banner below the sidebar states '10 FIWARE GEs fully tested and labelled'. The page also features a table of capabilities and a section titled 'ASSESSING FIWARE GES QUALITY' with a list of assessment activities and a progress bar at the bottom.

EUROPEAN COMMISSION
Community Research

FIWARE

page discussion view source history

FIWARE QA Activities

The Quality Assurance chapter is mainly devoted to analyze and assess the GEs are working as they are expected to work according to specifications. For this reason, the QA chapter is organized in two sub-tasks:

Contents [hide]

1 Functional Testing

1.1 Compliance, consistency and soundness of the documentation

10 FIWARE GEs fully tested and labelled

The 10 most used GEs at the core of FIWARE architecture

Capability	GE	Label
Development of Context-Aware Applications	Orion	QA A++
Handling Authorization & Access Control	KeyRock Authz Force	QA A++

ABOUT BLOG NEWS DEVELOPERS COMMUNITY FOUNDATION

ASSESSING FIWARE GES QUALITY

APIs, Developers, Experimentation, GEs, NGSI

FIWARE is rapidly moving from experimental to production environments in which the platform must scale in reliable and real workload conditions. This fact implies that all FIWARE GEs must work at an adequate quality, reliability and at performance level appropriate for these conditions. A dedicated activity has been launched in the framework of the initiative to analyze and assess the level of quality of each GE, providing diverse kind of reports and an assessment dashboard.

The quality is evaluated from different points of view:

- Curation of GEs documentation (documentation testing), both inspecting the code and the accompanying documentation (installation manuals, user guidelines, and similar). The goal of this assessment is to support FIWARE users with high-quality support for installation, configuration and operation of FIWARE technology, thereby improving the FIWARE user experience in general.
- Verification of the GE specification (functional testing), developing the appropriate test cases to assess if the GEs implementation corresponds to what is defined in the specification.
- Assessment of performance, stability and scalability of GEs in operational environments, like under excessive workload (stress testing). Test scenarios are defined and executed such that limits of a GE under test are identified, and can be compared with reference levels. The goal of this assessment is to favor the applicability of FIWARE in purely commercial scenarios.

The testing of the documentation and verification has been done for all GE not deprecated in FIWARE Catalogue (28 in total).

ie QA functional test

100% 90% 80% 70% 60% 50% 40% 30% 20% 10% 0%

GEs in Progress
GEs Not Applicable
GEs To Exclude
GEs Not Passed

The future

- Upon the continuation of presented activities...
- **Enlarge the set of tests** to be more complete and extensive to all GEs and bundles
- **Automate** as much as possible the tests and the labelling process
- **Integration** of tests with FIWARE development process



Why should I trust in FIWARE?



- **Because...**
- we are **continuously testing** all GEs
 - including documentation, APIs and performance among others
- we are **publishing all** the performed tests and obtained results
 - in a transparent and open way (GitHub, Docman)
 - guidelines for replicating the tests
- we are providing **recommendations** to the GE owners
 - to improve their functionality and behaviour

| Thank you!

<http://fiware.org>

Follow @FIWARE on Twitter

