FI-WARE Testbed design

FI-WARE-12-03-05

Topics addressed: First official release of the FI-WARE Testbed design

Editor: Stefano De Panfilis

e-mail: stefano.depanfilis@eng.it

Change History

Release	Major changes description	Date	Editor
0.0	First draft released for internal Testbed Chapter review	13-feb-2012	Thorsten Sandfuchs, Paolo Zampognaro, Stefano De Panfilis
0.1	Review based on: new info provided about Generic Enablers Implementations, and on comments from Telefonica	21-feb-2012	Stefano De Panfilis Paolo Zampognaro Miguel Carrillo
1.0	Integrated details about IoT Communication GE in IoT and S3C GE in I2ND.	24-feb-2012	Stefano De Panfilis Paolo Zampognaro Miguel Carrillo Hans Einsiedler Marika Stålnake Pierangelo Garino

TABLE OF CONTENT

1 Executive Summary

1.1 About this document

This document provides a high-level description of the FI-WARE Testbed (testbed in the following) and specifies the architecture of the testbed matching its components with the deployment requirements coming from the various FI-WARE Technology Chapters. The components are also described in terms of their actual implementation technology. In the above context, this document focuses on the infrastructure needed to set-up the testbed. Such infrastructure is composed by a central data-center, local infrastructures and stand-alone devices (e.g. local routers, android devices). The physical machines will be located in a dedicated data-center facility. The testbed infrastructure will take direct advantage of such machines or can be virtualised on top of them according to specific needs.

This document is a living document and solely FI-WARE Testbed Version 1, i.e. it describes the status of the testbed at the moment the document is published. Further testbed evolutions, based on changes at this point in time unforeseeable, might occur and will be duly reflected in the next releases of this document. The most updated version of the document can be found at:

- http://www.fi-ware.eu

1.2 Intended audience

This document is mainly intended for:

- FI Application designers and implementers, in particular those coming from the FI PPP Use Case projects who want to get what are the functionalities that the FI-WARE testbed is going to provide and which are the technologies on top of which FI Applications will be built
- Organizations interested in joining the FI-WARE Open Innovation Lab Community
- Organisations who wants to set-up their own FI-WARE Core Platform Instance.

1.3 Context

The FI-WARE Testbed implements the FI-WARE platform which is the Core Platform whose development is being targeted within the Future Internet PPP initiative launched by the European Commission in collaboration with the Industry.

In few words, better explained in following sections, the FI-WARE test-bed serves two functions:

1. It is the environment where to verify the developments of the FI-WARE project, to perform the required integration testing, and to validate FI-WARE as a whole before offering the testbed to the Use Case projects during phase 1 and to Use Case Trials for instantiation in phase 2 of the PPP program.
2. It will be the environment which will be used by the Use Case projects to perform specific domain experimentations.

More information about Future Internet PPP initiative can be found at:

- http://www.fi-ppp.eu
- http://ec.europa.eu/information_society/activities/foi/lead/fippp/index_en.htm

1.4 Structure of this document

The first section provides a brief discussion on the key concepts behind the FI-WARE Testbed. The two following sections present the general roadmap and the detailed plan for the deployment of the FI-WARE Testbed. These sections provide the overall rationales of the choices made. In particular the staged incremental approach for the deployment of the final FI-WARE Testbed (at month 36) is sketched given the information available at this point in time. Other two versions of this document are planned in order to provide more details for the intermediate and final version of the FI-WARE Testbed.

Then the final section describes in full details the testbed infrastructure technologies and the process adopted to identify such technologies.

This document has been built with the full and complete cooperation on the FI-WARE Testbed Team and the various Generic Enabler Implementations responsible FI-WARE project partners.

1.5 Related documents

There are no specific documents associated to this one apart from the ones being referred here or in the "References" section accompanying the description of each of the technical chapters on which the FI-WARE Reference Architecture is structured. However there are several documents related to this, they are

> "FI-WARE Vision" document (available at https://forge.fi-ware.eu/docman/view.php/7/331/FI-WARE+High-Level+Description+v1.1+11-08-31.pdf)

Future releases of this document will be strictly related to:

> "FI-WARE Architecture" document
>
> "FI-WARE Technical Roadmap" document
>
> "FI-WARE FI-WARE GE Open Specifications" series of documents
>
> Series of documents associated to each software release according to the FI-WARE project time schedule.

2 The FI-WARE Testbed: overall description

2.1 What is the FI-WARE Testbed

The FI-WARE project will generate a FI-WARE Instance, referred to as ―FI-WARE Testbed – defined as:

> ***"FI-WARE Testbed**: A concrete FI-WARE Instance operated by partners of the FI-WARE project that is offered to Use Case projects within the FI-PPP Program, enabling them to test their proof-of-concept prototypes. The FI-WARE Testbed is also offered to third parties to test their Future Internet Applications although support to them is provided on best-effort basis."*

The FI-WARE Testbed is aimed to be complete, in the sense that it will comprise reference implementations of the Generic Enablers defined in the FI-WARE Architecture. The testbed will not necessarily be centralised, but will be under central control and be accessible from a dedicated website.

However, the FI-WARE Testbed is not a FI-WARE Instance in production (i.e., serving trials involving real users). Therefore, there is no certification or production environment linked to it.

Given the very heterogeneous nature of FI-WARE Generic Enabler Implementations, the testbed aggregates in a flexible manner all the technologies the various Generic Enablers require. The testbed as such will not reside in a single location although its operation will be centrally controlled and monitored. To this extent the testbed will include software, hardware, and networking elements that will allow the testing of all the components.

2.2 Objectives of the FI-WARE Testbed

The testbed is the FI-WARE tangible asset that is responsible for making possible all relevant experimentation activities around FI-WARE and for making FI-WARE's technical implementation a reality.

The testbed is of critical importance as it allows the various Future Internet stakeholders to understand not only the viability of the solutions proposed, but also the very new nature of FI Applications. Such applications are built on specific instances of the FI-WARE platform which integrate, according to specific approaches, all the technological dimensions of Future Internet (e.g. IoS, IoT, IoC, Future Networks, and Security concerns).

Ultimately the testbed is devoted to allow the Use Cases projects to perform the validation of the FI-WARE technologies. To this extent, a significant involvement of the people involved in the Use Cases projects is expected; without them any significant validation, other than the technical integration test, is not possible. Moreover, a close coordination with the Capacity Building and Infrastructure Support project will be mandatory in consolidating details about existing and emerging advanced infrastructures that can be used in the context of the FI-WARE Testbed.

It is expected to open up the testbed to third parties as an Open Innovation Lab from the second release of the testbed onwards, and to help, through the test-bed site, Programme Facilitation and Support in engaging SMEs and maximising their involvement in phases 2 and 3 of FI-PPP Work Programme.

2.3 Operation of the FI-WARE Testbed

The testbed main basic infrastructure will be hosted by a centralised dedicated data-center hosted under industrial principles of operations. At the point in time of writing his document a negotiation is in place between the FI-WARE Consortium and the possible hosting company which is not member the FI-WARE Consortium. However if such negotiation takes longer with respect to project time-schedule Engineering Ingegneria Informatica will put its data-center at the project's full disposal.

The data-center facility will be solely responsible for running the hw and networking infrastructure, whereby the responsibility of the FI-WARE Testbed Team is to configure it according to the findings in this document.

In addition, the FI-WARE partners will provide support to Use Case projects for the deployment of applications (e.g., the conceptual prototypes) on top of the FI-WARE testbed. Tests run by the Use Case projects, coordinated with tests defined by the FI-WARE project, will help to validate Generic Enabler Open Specifications, the reference implementations of FI-WARE Generic Enablers developed within the FI-WARE project, as well as the conceptual prototypes developed by Use Case projects.

It is expected that the testbed will take benefit, as needed, by external facilities such as those provided by FIRE. **FIRE** is an initiative at the EU level that intends to create a multidisciplinary research environment for investigating and experimentally validating highly innovative ideas for new networking and service paradigms. FIRE promotes the concept of experimentally-driven research, combining academic research with the wide-scale testing and experimentation that is required for industry. FIRE works to create a European Experimental Facility, which is constructed by gradually connecting and federating existing and upcoming testbeds for Future Internet technologies.

The following picture exemplifies the overall topology of the testbed.

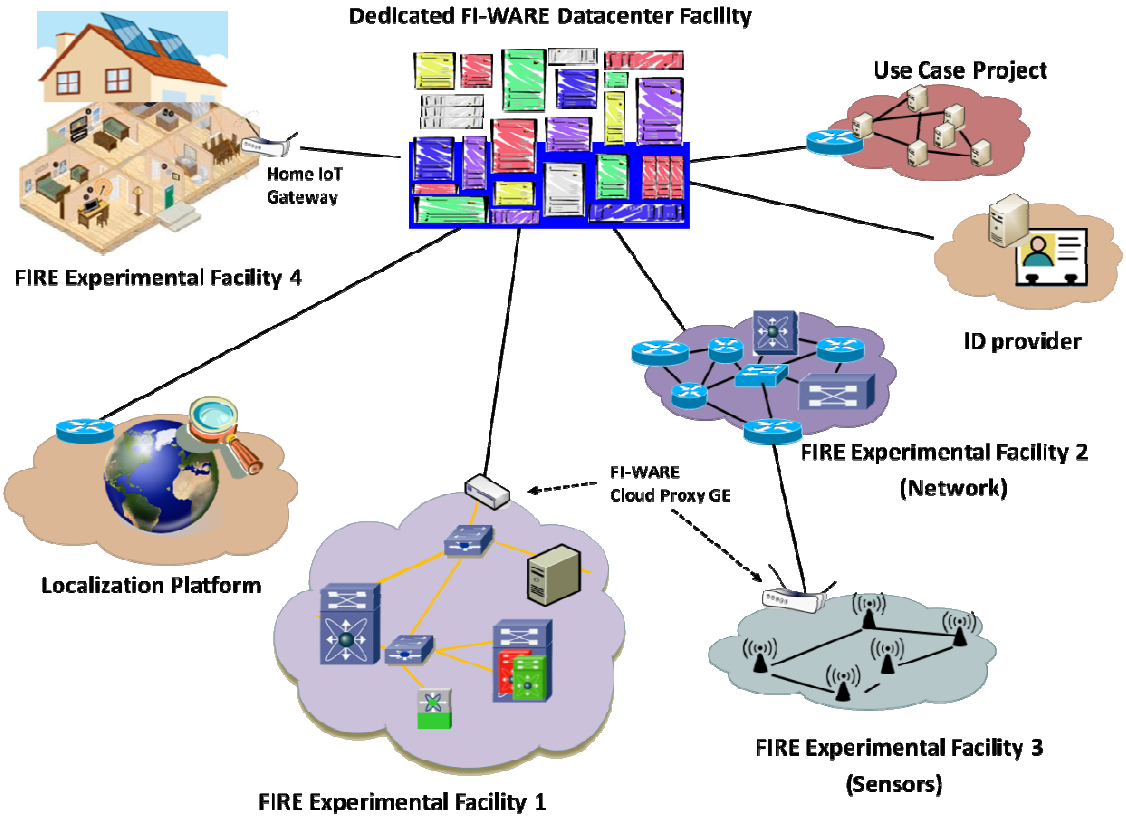
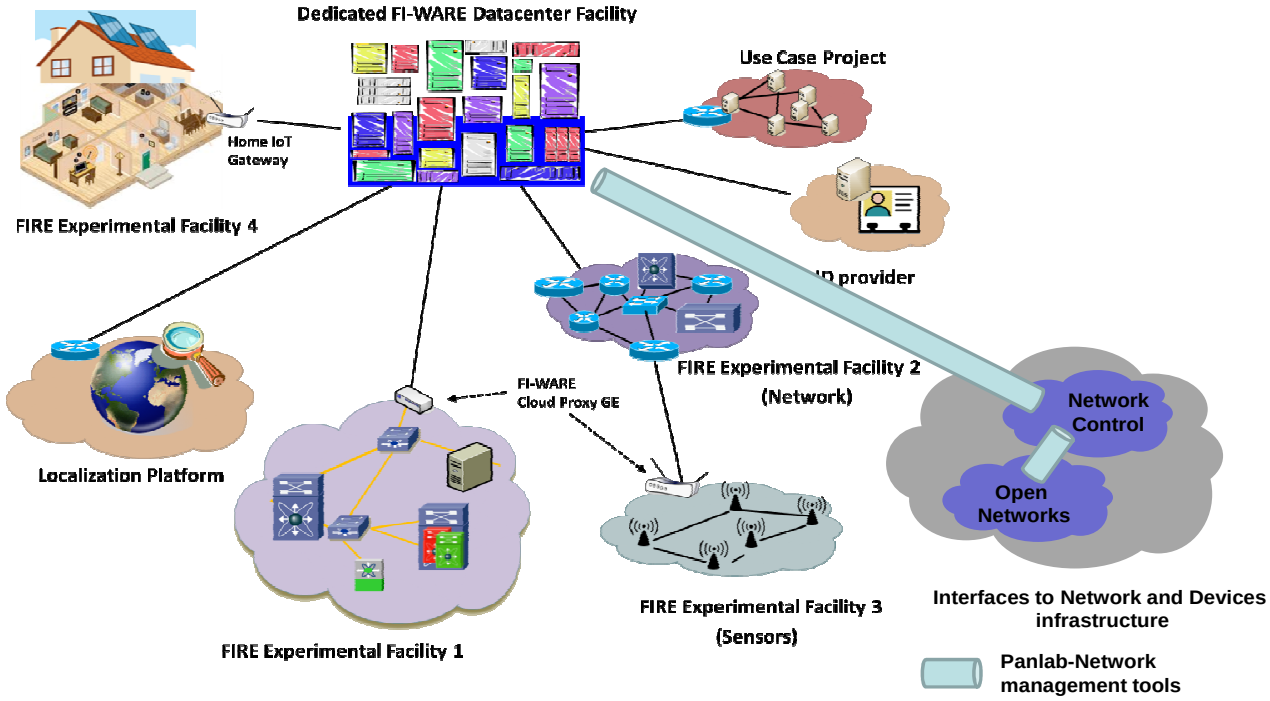


Fig 1- FI-WARE overall topology Regarding the FI-WARE internal network infrastructure development, it is planned to use the PanlabII network management tools for interconnection. The EU-project PanlabII (Pan European Laboratory Infrastructure Implementation, http://www.panlab.net/) offers the possibility to manage the infrastructure locally and to connect other locations, which are far away, very efficient into an overall and common testbed. The interconnection can be implemented via the GEANT network (http://www.geant.net/).

Such an infrastructure and interconnection will provide the means to test the functionality of those Generic Enablers which are physically bound to a remote infrastructure; these mostly include the implementation of the networking Generic Enablers in the Interface to Networks and Devices chapter, which are described in

next Sections of this document. The other locations will have restricted management access to the remote infrastructure but they will have full testing capabilities over the I2ND networking GE implementations. Therefore the physical transfer of the infrastructure can be postponed until the last stage of testing. And the costs for business trips to the overall common testbed can be held small.



3 Overall FI-WARE Test-bed roadmap

FI-WARE will adopt an agile and phase-based approach to support relevant experimentation, design and setup of the FI-WARE Testbed for the complete FI-WARE project.

The following table relates major testbed milestone with major milestones of the overall project timeframe and respective phases (In the following of this document whenever a project milestone is indicated please note that M0 is 1st May 2011).

Month	Phase	Testbed Milestone	Related and major FI-WARE Milestone
M9		Testbed Design for testbed V1 due at M15	1st release FI-WARE Architecture + Technical Roadmap
M10		(FI-WARE) Testbed Integration Plan	
M12			1st FI-WARE release, including: • FI-WARE v1 GE Specifications • Installation and Administration Guide • SW release • User & Programmers Guide • 3rd party innovation enablement
M13-M14	**First Integration phase**		
M15		1st release of the FI-WARE Testbed (available to the Use Case projects for early conceptual prototypes)	
M15		(FI-WARE) Testbed Integration Report	
M15-M23	**Validation phase**		
M18			1st UC conceptual prototypes finished
M18		Report on Validation Process including Validation with Use Case projects	

Month	Phase	Testbed Milestone	Related and major FI-WARE Milestone
M21		Testbed Design for testbed V2	2nd release FI-WARE Architecture + Technical Roadmap
M22		2nd (FI-WARE) Testbed Integration Plan	
M24			2nd FI-WARE release, including: • FI-WARE v2 GE Specifications (include APIs) • Installation And Administration Guide • SW release • User & Programmers Guide • 3rd party innovation enablement
M24			2nd UC conceptual prototypes finished
M24		Report on Validation Process including Validation with Use Case projects	
M25-M26	**Second Integration phase**		
M27		2nd release of the FI-WARE Testbed (available to test developments by UC Trials)	
M27		2nd (FI-WARE) Testbed Integration Report	
M27-M32	**Integration and validation phase**		
M33			3rd release FI-WARE Architecture + Technical Roadmap including: • FI-WARE v3 GE Specifications (include APIs) • Installation and

Month	Phase	Testbed Milestone	Related and major FI-WARE Milestone
			Administration Guide • SW release • User & Programmers. Guide 3rd party innovation enablement
M33		2nd Report on Validation Process including Validation with Use Case projects	
M36		3rd release of the FI-WARE Testbed (available to the Use Case projects for early conceptual prototypes)	FI-WARE v3 to be made available to UC Trials.
M36		Documentation and training material related to FI-WARE Instances Operation.	

4 Roadmap for Internal Testbed setup

.

The following milestones, summarised from the previous overall table, will lead the testbed setup and availability of integration infrastructure to the other FI-WARE Chapters.

Month	Milestone	Related FI-WARE Input to WP10
M6-M10	Testbed Design Phase	
M9	Testbed Design for testbed V1 due at M15	1st release FI-WARE Architecture + Technical Roadmap
M10	(FI-WARE) Testbed Integration Plan	Integration plan for WP3-WP8 Generic Enablers and their integration plans
M12		Input from use cases and their requirements
M13-M15	Testbed Integration Phase	
M12	Testbed hw and sw basic infrastructure (i.e. the data-center facility with configured machines) up and running for the integration phase	FI-WARE v1 GE committed system and hardware requirements
M15	Start Testbed Validation Phase	Established connection to use case projects and joint validation plan
M15	1st release of the FI-WARE Testbed (available to the Use Case projects for early conceptual prototypes)	Support on operation by every partner, providing Generic Enablers in the first version of the testbed

4.1 Main phases for Internal Testbed setup

Aligned with the milestones outlined in the previous sections the testbed work package operates according to different phases, namely design, integration and validation phase. The following sections will briefly introduce each phase and the main tasks outlined in these phases. The outlined tasks and responsibilities are relevant for the complete project life time and might be changed and reviewed according to the feedback received during execution of this plan.

4.1.1 Testbed Design Phase

During the design phase the requirements from Generic Enablers (GE) providers is gathered and analyzed. The deliverable will be this document in three different versions (Testbed design D.10.1.a-c) which incorporate information on system availability and sizing information in order to setup and operate the testbed.

In order to be successfully integrated and deployed in the testbed each Generic Enabler Implementation (GEI) needs to be complemented with information on availability, maturity, hardware and system requirements and potential dependencies and related integration plans with other Generic Enablers. The

testbed work package analyses the integration plans coming from Generic Enablers and compiles the integration plan (M10).

4.1.2 Testbed Integration Phase

According to the various FI-WARE stakeholders the testbed integration phase includes the following tasks:

Testbed Team:

- Supervision of integration process, executed by implementers
- Sign-off deployment of GEI and GE-functionality for each testbed version
- Gather usage requirements and usage scenarios from use case projects for validation phase
- Design of validation plan together with use case projects
- Plan provisioning of testbed system landscape
- Define service level agreements and terms of use of the testbed

Generic Enabler Providers (GEP):

- Propose GEI to be integrated in specific testbed version
- Implementation of integration with other GEI, according to identified dependencies and the integration plan
- Packaging of software artifacts for deployment
- Provisioning of testbed hardware (if appropriate for the GE) to the Testbed hosting platform – this is applicable only for those GE implementations which will be offered only as services, and are planned to run on private local infrastructure
- Deployment and unit test of GE-software within the dedicated testbed hosting platform.
- Configure and minimal integration
- Provide input for possible validation criteria for validation of the GE by others

Use Case projects

- Define validation requirements and scenarios
- Define validation KPIs together with Testbed Team
- Plan validation execution

4.1.3 Testbed Validation Phase

The Testbed Validation Phase will allow use case projects and other consumers of the Testbed to validate their defined scenarios and gather validation KPIs. Deliverable D.10.5.a-c will incorporate respective validation results.

4.2 Testbed Release Planning

The Testbed will release at least three versions to be used by the respective consumers of the Testbed within the overall project time frame respectively at M15, M27, and M36.

Releases are limited to the defined conditions which influence the content and the scope of the releases. Relevant dimensions are: scope and availability of Generic Enabler implementations, operational aspects of

Generic Enabler implementations, availability of GEs on private or public instances, service level agreements.

Version 1 (M15) will incorporate contributions from most of the Generic Enabler and compile a first preliminary version of the Testbed. Main usage scenario is built bottom-up from the available Generic Enabler. It is composed of a set of "guided tours" in order to allow use case projects to explore and understand the capabilities of the FIWARE platform and the Generic Enabler provided on the platform. Furthermost the integration between GE can be explored and tested within predefined validation scenarios by UC projects and GE providers jointly with the Testbed team. During Testbed V1 operation main focus is on feedback gathering and requirement engineering in order to design and enhance the platform according to future use case trails. A detailed second integration plan will be derived for Testbed Version 1 and new validation scenarios will be defined. It is not expected to have UC specific enablers deployed on this Testbed version.

Version 2 (M27) of the Testbed will incorporate most of the functional features provided by the GE, first feedback and requirements will be already integrated in the software, depending on the backlog and commitment of the GE providers. In this version the Testbed should support predefined use case trials, previously being signed off by the Testbed team. Additional support for scalability and higher availability should be supported in this release of the Testbed in order to validate and harden the platform for broader usage. Validation scenarios defined and signed-off together with the Testbed Team and the UC projects can be deployed on the Testbed and validation scenarios should be executable. An initial set of UC specific enablers could be integrated in this Testbed version. This is the first version of the testbed which will be the core of the FI-PPP Open Labs.

Version 3 (M36) provides a fully functional Testbed to prepare and support the design of further industrial FIWARE instances, later being used within the Use Case Trials projects for their ongoing work. For Testbed version 3 scalability and availability will be increased and general availability to a larger audience and field trials should be targeted.

Use Case projects should be allowed to use this Testbed version to integrate and run guided tours on their scenarios. Furthermore this Testbed version will help to proof functionality of the use case trials.

4.3 Testbed System Landscape and Service Level Agreements

The system architecture and hardware requirements, both static and dynamic, follow the testbed release planning and will incorporate a staged approach depending on: a) use case requirements, b) possible deployment and availability plans and c) system/hosting availability. Feedback from validation phase and requirements from the Use Case projects will further define the scope and the hardware requirements for broader field tests and upcoming releases.

Testbed version 1 (M15) will be deployed on a defined set of hardware and provide access for the basic integration work, executed by the Generic Enabler providers.

Currently the following usage assumptions apply to Testbed Version 1:

- No production-like use is permitted on the Testbed
- Noncommercial use only is allowed
- No deployment of UC specific enablers is allowed
- Although the basic infrastructure on top of which the Testbed will be deployed is governed under industrial operational standards (e.g. 8h/57d availability, out of bank holidays, full recovery mode), the usage of the Testbed should be always under the control of the Testbed Team. Final SLAs will depend upon further negotiations with the infrastructure provider. Future versions of this document will incorporate any updates on this matter.

5 Testbed overall Architecture and Design

5.1 Introduction

The objective of this deliverable's section is to identify one or more possible testbed infrastructure configurations.

Such result was obtained by following some steps. In the first one (section 5.2) it is highlighted and stated, what it is considered as testbed infrastructure components and what instruments were adopted to gather requirements about them. As second (section 5.3) a deep analysis is performed in order to extract an appropriate schema about the resources required from different chapters and a synthetic specification document, based on such schema, is filled. Finally, by exploiting the information contained in the specification document, a possible testbed infrastructure configuration is proposed (section 5.4).

5.2 Definitions

In order to specify the testbed (v1.0) architecture, it was chosen, preliminarily, to describe the FI-WARE conceptual architecture (see Fig 2) based on five distinct layers: FI applications, FI Core Platform – Application Layer (i.e. GEs reference instances), SW Infrastructure, FI Core Platform – Virtualisation Layer (not implemented in FI-WARE Testbed Version 1) and HW Infrastructure.

Generic Enablers provide open interfaces for the development of FI Applications belonging to several usage areas while the infrastructure layer offers all needed resources for the deployment of GEs themselves.

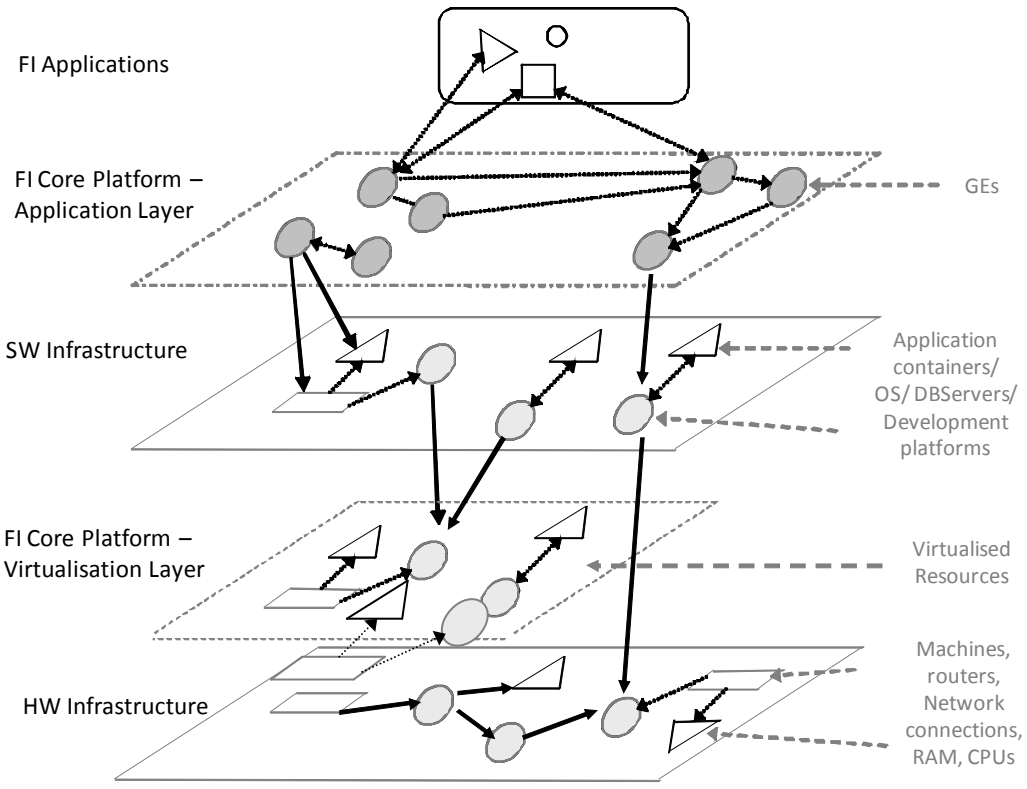


Fig 2- FI-WARE conceptual architecture

These preliminary remarks were mandatory in order to clarify the definition reported at the very beginning of this document (sect 1.1) qualifying the nature of the FI-WARE Testbed.

This document specifies the architecture of the testbed matching its components with the deployment requirements coming from the various FI-WARE Technology Chapters. The components are also described in terms of their actual implementation technology.

In fact, on the base of the conceptual architecture introduced above, it became clear that the *components* of the test-bed architecture, are the hardware and software resources owned by the infrastructure layer in order to offer an appropriate run-time environment to the core platform (i.e. all GEs coming from the various FI-WARE technological Chapters).

The following step, then, was to identify exactly the nature of such resources. To address this objective it was proposed to all FI-WARE Technology Chapters to indicate their specific requirements by filling a questionnaire put on the FI-WARE project wiki (see Fig 3). This step was needed because at the point in time of writing this document no formal releases were due by the various chapters, so intermediate bilateral workshop, on the base of the questionnaire were organised to gather the needed info. The first question is the most relevant since it is aimed at gathering GEs hw-sw requirements, while, the others, address more general needs for the further development of the testbed.

In fact, the second and third questions provide information about the releases plan in terms of both functionalities exposed by each single GE and potential relationships among the GEs themselves. This information is particularly relevant for the setup of the testbed integration plan foreseen at month 10. The fourth question has been introduced, instead, in order to understand the requirements affecting the distribution degree of the testbed. Such aspects, in fact, impact the testbed architecture definition certainly from a technical point of view, since they can influence the testbed topology (e.g. centralized configuration versus star configuration) but, also, from a management perspective since a distributed approach implies a greater effort in order to guarantee security and reliability. In addition it was asked to each chapter to report on already established contacts with Use Cases projects. The answers convey, from one side, information about the concrete needs for the adoption by the Use Case projects, which is the most critical point for the whole FI PPP initiative, and from the other one, concrete aspects about the setup of a more efficient integration plan.

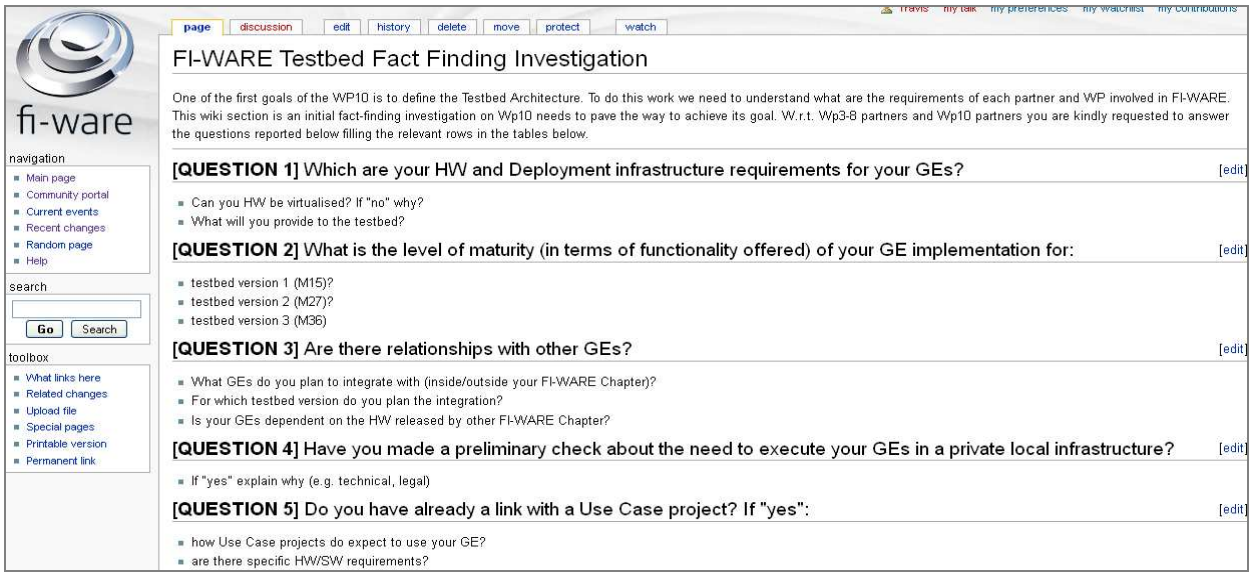


Fig 3 - Testbed questionnaire

5.3 Overall descriptions

In this section the answers to the question one, provided by the GEs owners per each chapter, are analysed in order to extract a common taxonomy to identify the software and hardware resources (i.e. the testbed infrastructure component) they need. Each category will be located with respect to the following four predefined levels: hardware, machine, operative system, applications and specific required categories instances will be highlighted as well. In order to make more understandable the process also a graphical representation of the new categories and instances of this emerging ontology is provided at the end of each step.

5.3.1 Application and Services Ecosystem and Delivery Framework

Generic Enabler	Organisation	Q1	Q2
Composition Editor	DT	Can be virtualized, LinuxVM, J2EE (Tomcat 6.X), MySQL, BPEL Engine ActiveVOS	First version M15
	EAB	Virtualization: yes; needs J2EE	Preliminary version M15
	UPM	It can be virtualized, Django 1.3, Python 2.5, 2.6 or 2.7, Apache, MySQL/PostgreSQL/Sqlite3	Preliminary version ready for M15; M27 Second Version; M36 Final version
	Atos	It can be virtualized, J2EE (Tomcat 6.X), Postqresql 8.3.X, BPMN 2.0 Engine (Activiti.org), RDF/S Repository (Sesame), Oryx Editor	First version M15 (Preliminary Version, Design time), M27 Second Version, Design and runtime. M36 Final version (design and runtime)
Composition Execution	DT	Can be virtualized, Linux VM, J2EE (Tomcat 6.X), MySQL, BPEL Engine ActiveVOS	First version M15
	EAB	Virtualization: yes; needs J2EE	Preliminary version M15
	UPM	It can be virtualized, Django 1.3, Python 2.5, 2.6 or 2.7, Apache, MySQL/PostgreSQL/Sqlite3	Preliminary version ready for M15; M27 Second Version; M36 Final version
Marketplace	SAP	can be virtualized J2EE, Linux VM	First version M15
Mediation	ATOS	It can be virtualized, J2EE (Tomcat 6.X), RDF/S Repository (Sesame)	First version M15 (Preliminary Version, Design time), M27 Second Version, Design and runtime, M36 Final version (design and runtime)
	THALES	Can be virtualized; Java/JDK	First preliminary prototype ready for V1/M15; M27 Second Version; M36 Final version

	Telecom Italia	can be virtualized, Linux VM or binary distribution (zip file) requirements: X86_64 (physical or virtual), Linux Ubuntu 10.4 LTS+, Sun Java 1.6.x+	First version M15
Registry	SAP	Can be virtualized J2EE, Linux VM	First version M15
Repository	SAP	Can be virtualized J2EE, Linux VM	First version M15
Revenue Settlement & Sharing System	TID	No, it will be installed in an internal testbed due to IPR concerns. Connectivity must be enabled between TID's internal testbed and FI-WARE's testbed.	M15: First preliminary prototype ; M27: Second Version; M36: Final version
SLA Management	ENG	The specification is technology independent	No implementation is provided
Store	EAB	virtualization: no; the preliminary version will be offered as SaaS, needs: Sun Solaris; J2EE – Jetty	
USDL Service Description	SAP	Not relevant, no deployable software component	Not relevant, no deployable software component

By analysing the terms contained inside the answers to the first question the following leaf resources *categories* were extracted: Python Language Interpreter, J2EE Application Server, WS-BPEL Orchestration Engine, BPMN Engine, Process Editor, Development Platform, SQL DB Server, RDF/S Repository, Web Development Framework.

Also the direct super categories (i.e. Programming Language Interpreter, Application Server, Workflow Engine) were reported (see Fig 3) since they can be useful in the classification process involving the other chapters. It is also important to note that while some chapter has expressed the need of a leaf category (e.g. J2EE Application Server) some other has indicated the need of a specific “individual” of such category (red nodes in Fig.3); for this reason the structure of the testbed architecture specification document (see section 5.3.8) was organized to report both the information types.

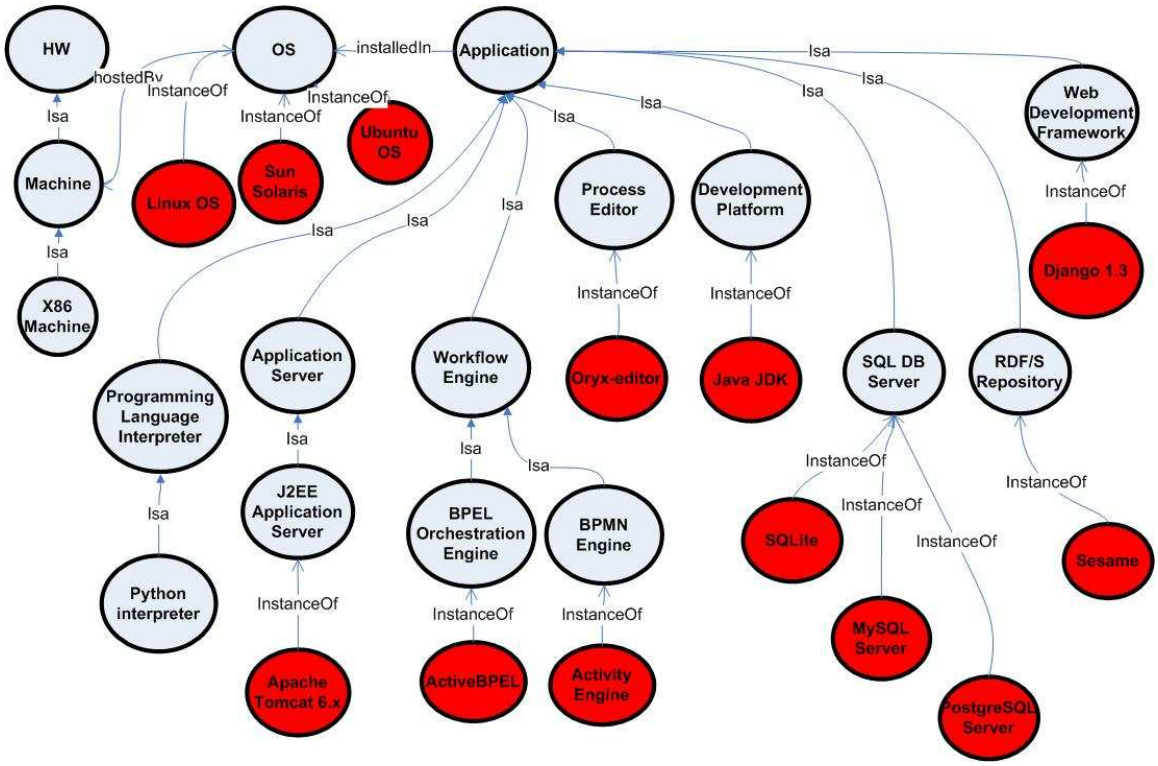


Fig 4 - Apps resource categories

5.3.2 Cloud Hosting

Generic Enabler	Organisation	Q1	Q2
IaaS DataCenter Resource Management	IBM, Intel	x X86 with Ubuntu, + dedicated network switches. Amount of machines and storage depends on UCs req.	fully functional for v1, with additional functions added to following versions
IaaS Service Management	TID	1 VM: HDD - 10 GB maybe more if we want to manage image there, RAM - 4 GB, vCPU - 2 vCPUs, Network - no special requirements, OS: CentOS 5.1 EAP	fully functional for v1, with additional functions added to following versions
Object Storage	Intel	min 5 networked machines. Amount of storage depends on UCs req.	fully functional for v1, with additional functions added to following versions
PaaS Management	TID	3 VM: 1 VM: HDD - 10 GB maybe more if we want to manage image there, RAM - 4 GB, vCPU - 2 vCPUs, Network - no special requirements, OS: CentOS 5.1 EAP 1 VM: HDD - 10 GB, RAM - 512Mb, vCPU -	Not for V1

		1 vCPUs, Network - High, OS: Debian Squeeze 1 VM: HDD - 20 GB, RAM - 512Mb, vCPU - 1 vCPUs, Network - no special requirements, OS: Debian Squeeze	
Resource Monitoring	TID	1 VM: HDD - 10GB, RAM - 512 Mb, vCPU - 1 vCPU, Network - UDP connection with deployed VMs, OS - Fedora 14	Not for V1

By analysing the terms contained inside the answers to the first question the following leaf resources *categories* were extracted: Network switch. As for the previous case it was also reported (see Fig 4) their direct super categories (i.e. Network Component) and specific required individuals (e.g. Debian Squeeze). Furthermore some machine attributes were identified as the RAM quantity and HD capacity and also information about the required machines number is extracted as well. All these information were reported in the testbed specification document.

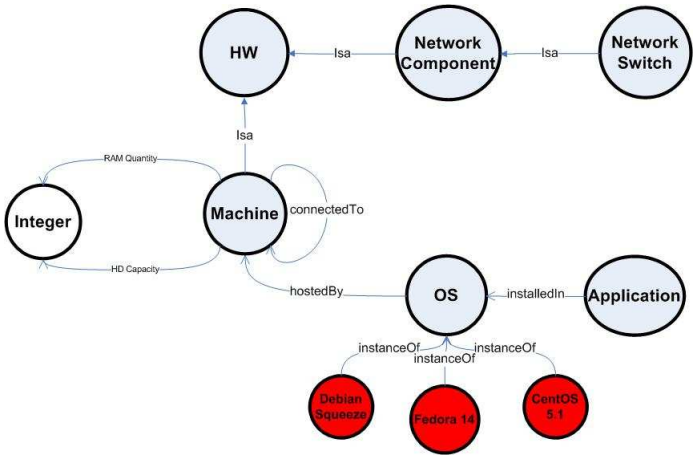


Fig 5 - Cloud hosting resource categories

5.3.3 Internet of Things Service Enablement

Generic Enabler	Organisation	Q1	Q2
IoT Communications	Orange	2 VMs with Windows XP or Linux. Required Softwares : Fosstrak Platform, MySQL, Apache Tomcat. We may need Eclipse IDE, Apache Maven and Rifidi Emulator for developments.	Preliminary version for M15 with additional functions added to folllowing versions.
IoT Communications	Telecom Italia	Stub: No HW requirements for this module. As SW is concerned this is a java application developed with JDK 1.6 and uses a DB MySQL release 5 or higher Skeleton: It is recommended a CPU >= 450 MHz for this module. As SW, this	Both the stub and skeleton are going to be used in a Telecom Italia field trial, named Energy@Home. So they will be tested to work together.

		module has been tested with JDK 1.6 and Equinox OSGI release 3.6. Both of the modules can be virtualized.	
IoT Communications	Ericsson	The main software platform on the gateway is OSGi which is a Java-based platform that allows loading/unloading software components called bundles. The minimum HW requirements for the OSGi platform are: • 16 MB flash for boot image (could work on 8M as well) • 16 MB flash for file system • 64 MB RAM (can work on 32 MB as well) • > 300 Mhz CPU The minimum software requirements are: • Linux 2.4 or higher • Java CDC 1.1.2/Java SE 1.4.2 or higher • OSGi 4.1 or higher, preferably Knoplerfish distribution Virtualization is not applicable. The gateway has to run locally in the IoT architecture of each Use Case project, meaning outside of backend and most likely on dedicated HW. OSGi bundles as binaries will be provided to the testbed.	Version 1: OSGi code base and additional bundles have pre-product grade, bundles targeting specific and separately defined capabilities stable prototype level. Additional functions will be added to following versions
IoT Data Handling	Orange	We need 6 VMs: 3 Ubuntu Server, 3 Windows 7 or XP. This requirement won't exceed 10 VMs. For each Vm: 1 proc, 1 or 2Go RAM depending on OS requirement(Windows or Linux), 10go HD. We have some hardware that needs to be virtualized: Android phones can be virtualized thanks to Google's Android emulator. RFID readers can be virtualized with RiFiDi, a dedicated emulator. We will have to build our own emulators for simple sensors. We will provide the testbed with Ubuntu server 32, Java, Tomcat, MySQL, Android Emulator, Petals ESB.	Version 1:pre-alpha maturity, with only small functioning parts. Version 2:functionnal, but still away from production quality. Version 3: fully functional and debugged.
IoT Process Automation	NEC	GE are Components based on Java + OSGi. No special hardware requirements; required amount of RAM not tested yet, but most likely no excessive requirements. - Geographical Index Structure will support distribution among several	- testbed V1 (M15): Most functionality, beta version - testbed V2 (M27): Expected full functionality + stable

		virtual machines employing the JGROUPS middleware (number of machines depends on the requirements of the use case) - Exposure Interface used both on Gateway and Backend We do not see any problem regarding virtualization.	
IoT Resources Management	TID	MongoDB, JEE, Subversion	Minimal version
Centralised Backend/Gateway without Exposure GE	NSN	Centos 6.2 64 with JDK 1.6, MySQL, Tomcat and CouchDB computer with at least 4GB ram. Can be virtualised using VirtualBox. Also testbed on the public Amazon EC2 instance.	v1: partial functionality, v2: full functionality prototype level, v3: full functionality

By analysing the terms contained inside the answers to the first question the following leaf resources *categories* were extracted: IDE, Software Project/Document Management Tool, Android Emulator, RFID Emulator. As for the previous cases we also their direct super categories (e.g. Emulator) and specific required individuals (e.g. Google's Android Emulator) were reported (see Fig 5).

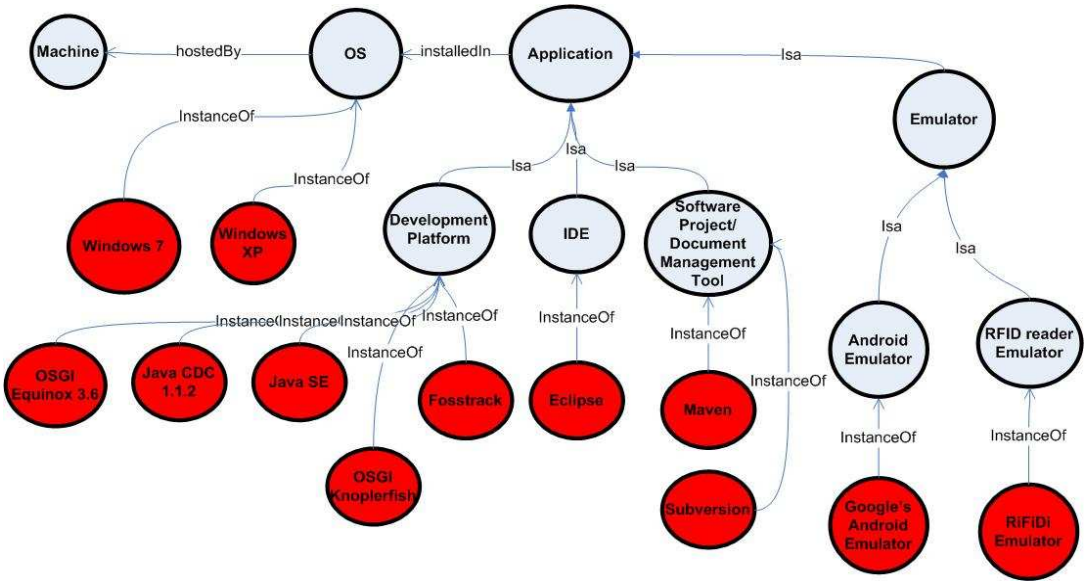


Fig 6 - IoT resource categories

5.3.4 Data/Context Management Services

Generic Enabler	Organisation	Q1	Q2

Big Data Analysis	TID	1 x X86-64 Ubuntu 11.04 or CentOS 6.0 Linux VM with 4GB of RAM/10GB of disk. Performance usage requires 3 or more machines of the same spec	Partially functional with the SAMSON platform with MongoDB for v1 with the integration with HDFS coming later on
Complex Event Processing	IBM-IL	1 x Linux with 4GB of RAM/10GB of disk. Performance usage requires 1 or more machines of the same spec	Version1: full CEP functionality with adapter framework in place
Localization Platform	Thales	It depends of the capacity requirements in term of request/second. A simple PC can offer enough services for a trial. We will provide a remote connection of our infrastructure, for the test bed.	available with first set of release for testbed version1, 2nd set of release for testbed version 2
Meta-data Pre-processing	Siemens	1x VM (Win XP or higher with .NET 4.0 or higher), 4 GB RAM	Version 1: -; version 2: full functionality of core features (e.g., metadata handling, metadata transformation)
Multimedia Analysis	Siemens	1x VM (Win XP or higher), 4 GB RAM	Version 1: change detection; version 2: object detection, foreground/background detection
Publish/Subscribe Broker	Telecom Italia	(in a minimum configuration, could be virtualized): 1 CentOS5 server with 2 CPUs; 2 GB RAM; 50 GB free hard disk space on used partition, 1Gb Ethernet. Better if this centralized GE is installed in a HA/FT configuration. JBoss AS, Java SE SDK, MySQL	Context data handling in v1; Mainly partial NGSI10 support for v1. NGSI9 support is minimum for context/data providers registration only in v1; Data and more NGSI9 support for following releases
Query Broker	Siemens	1x VM (Windows or Linux) JRE 1.6 or higher, 4 GB RAM	Version 1: full functionality of core features (i.e., federated database access)
Semantic Annotation	Telecom Italia	(in minimum configuration; could be virtualized): 1 Ubuntu or CentOS server with 2 CPUs; 16 GB RAM; 1 TB free hard disk space on used partition, 1Gb Ethernet. Better if this centralized GE is installed in a HA/FT configuration. Apache, PHP, MySQL, SPARQL/Virtuoso, Freeling, DBpedia	Key-word extraction and DBpedia based semantic geo-tagging in v1. Other annotations and features in next releases dependent on required application domains
Semantic Application	Atos	It can be virtualized: Ubuntu or CentOS server with 2 CPUs, 12	Basic ontology management capabilities accessible through REST APIs. RDF

Support		GB RAM, 1 TB free hard disk space. JDK 1.6, Tomcat Server, JBoss AS, Sesame RDF triplestore, OWLIM, SQL database.	storage, SPARQL querying and OWL 2 RL inference provided by the infrastructure
Unstructured data analysis	Atos	N/A GE integration starting from second year	N/A GE integration starting from second year

By analysing the terms contained inside the answers to the first question the following leaf resources *categories* were extracted: Fault Tolerant Hardware Configuration, Natural Language Analyzer, RDF Query Language, Information Data. In this case no new super category was introduced (see Fig. 6) while specific required individuals were reported (e.g. OWLIM).

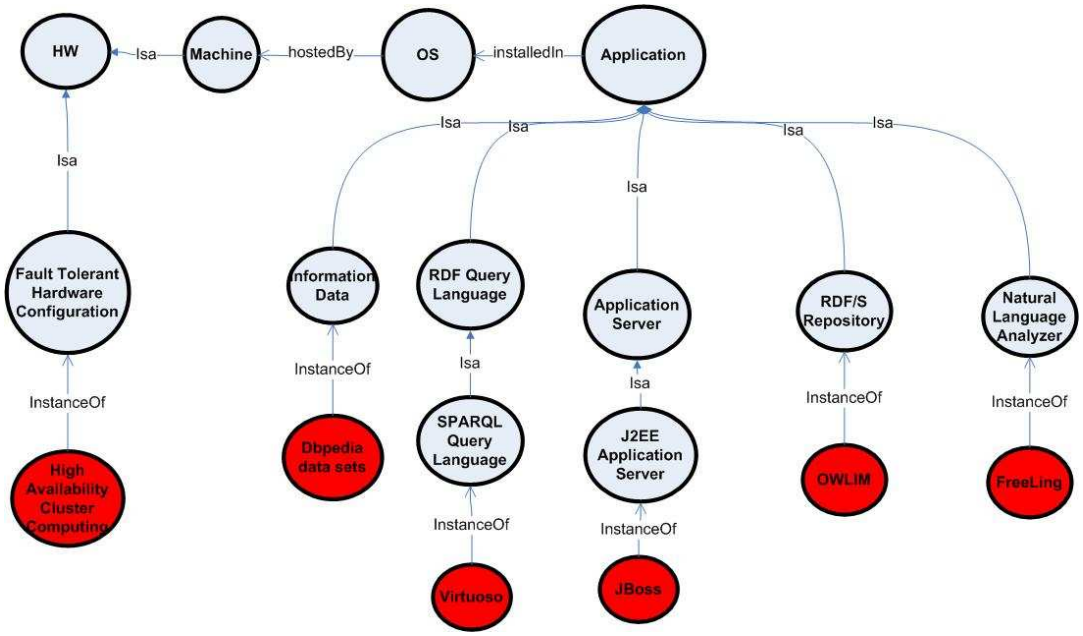


Fig 7 - Data resource categories

5.3.5 Interfaces to the Network and Devices

Generic Enabler	Organisation	Q1	Q2
Cloud Edge	Technicolor	Our GE is the "cloud proxy", sort of super gateway. The SW is 100% dependant on the HW we will provide. Our plans right now are to provide the 1st release running on a more or less standard small PC, the 2nd and 3rd releases might be more "industrial" being released on a specifically manufactured HW(tbc). The cloud	Preliminary version pc-based

		proxy is a terminal, it needs an internet connection and nothing more (xDSL).	
Connected Devices Interfaces	Intel	android device	Preliminary version
Network Information and Control (NetIC)	Alcatel-Lucent Germany	pc based (may be virtualized), needs IP link to external network information source	probably a minimal version
Network Information and Control (NetIC)	Alcatel-Lucent Italy	Java/C, on a dedicated workstation and vmware hypervisor on private IP	v1: preliminary version with synchronisation functionalities
Network Information and Control (NetIC)	NSN	preliminary pc-based, a specific hw not yet available	probably a minimal version
Service, Capability, Connectivity, and Control (S3C)	DT + Fraunhofer	Preliminary virtualized environment (6 virtual machines) deployed on a specific hardware at DT premises in conjunction with Fraunhofer	Preliminary version
Service, Capability, Connectivity, and Control (S3C)	Orange + Ericsson	Remote access or part of existing environment of the WP	Preliminary version

By analysing the terms contained inside the answers to the first question the following leaf resources *categories* were extracted: External Network, Mobile Phone, Virtualization Software. In this case no new super category was introduced (see Fig. 6) while specific required individuals were highlighted (e.g. Android).

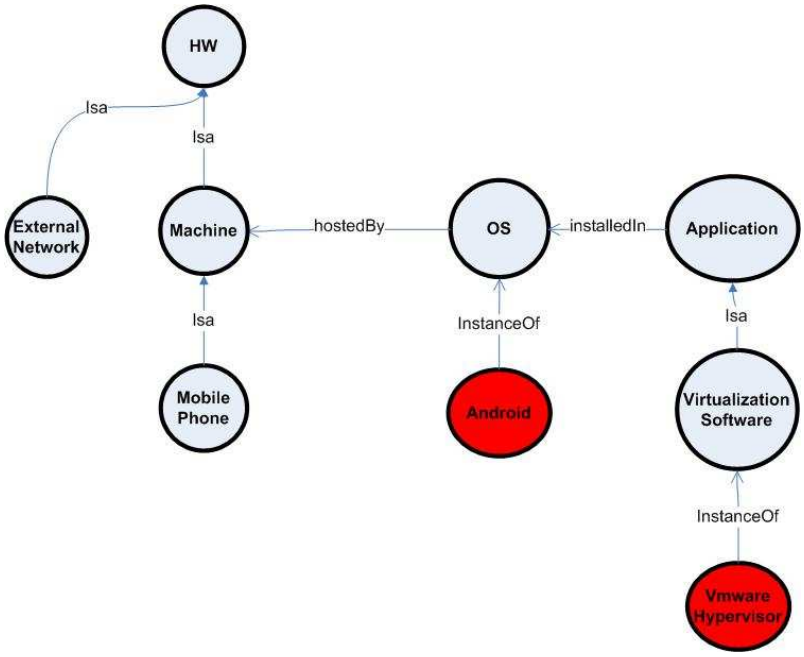


Fig 8 - Interface to Networks and Devices resource categories

The Interface to Network and Device chapter envisage an environment which has some inherent constrains. In general the various GEs will be developed and will run on specific equipment, which cannot be transferred to other places. Being aware about this, the chapter is considering to manage the planned tests and distributed testbeds by using a management tool which was developed in the framework of the PanlabII project (Pan European Laboratory Infrastructure Implementation, http://www.panlab.net/).

PanlabII was an EU funded project where Fraunhofer, Technicolor, Orange, Telefonica, and Deutsche Telekom, current FI-WARE partners, were involved.

The management team of the chapter is already in contact with the co-ordinator - Anastasius Gavras from Eurescom – to discuss the conditions under which to use such powerful tool.

5.3.6 Security, Privacy, Trust

Generic Enabler	Organisation	Q1	Q2
Context-based security and compliance	Atos	J2EE	Preliminary version for tesbed v2
Identity Management	NSN	It can be virtualized	
Identity Management	DT	No requirements as per answer to Q4	fully functional for v1, some fine tuning expected for following versions

Data Handling	SAP	JEE, MySQL - NB: a bundle shall be delivered, that will contain all dependencies. No other options will be available	Preliminary version (API incomplete), but fully functional for testbed V1 with additional functions to be added to following versions
DB Anonimisation	SAP	JEE, MySQL: a WAR file + instructions will be delivered, otherwise it could be possible to proceed as per Data Handling	Fully functional for testbed V1 with additional functions to be added to following versions
Privacy	Thales	Linux, Apache 2,Tomcat 6, Java	Probable a minimal version
Malware Detection System	INRIA	No requirements as per answer to Q4	fully functional for v1, with additional functions added to following versions
Security Monitoring	Thales	JEE, MySQL	Preliminary version for V1

By analysing the terms contained inside the answers to the first question no new leaf resources *category* was extracted.

5.3.7 Developer Community and Tools

DevComE component	Organisation	Q1	Q2
Catalogue	Ericsson	1 VM, LAMP installation (Ubuntu 32bit server edition 10.04, MySQL)	Preliminary version for testbed v1.
Forge	Engineering	4 VCPU, 8GB Ram, 100 GB HD, CENTOS 5.x 64bits, FusionForge, MediaWiki	Fully functional for testbed v1.
IDE	Engineering	1 VM, JDK 1.6, Eclipse Indigo, MS Windows, IOS (V1). Expected compatibility with other OS for further testbed versions	Preliminary version for testbed v1.
Software Performance Cockpit (SOPECO)	SAP	Offered as DevComE plug-in	
PROSA	University Duisburg-Essen	1 VM (Windows or Linux), JDK 1.6, Apache Tomcat (optional)	Prototypical state in testbed stage 1&2, Ready in stage 3

By analysing the terms contained inside the answers to the first question no new resources *category* was extracted.

5.3.8 FI-WARE Testbed: specification document

		HW									Application																	
		Machine								Network comp.	Fault Tolerant HW Configuration	Programming Language Interpreter	Application Server	Workflow Engine		DB Server	Web Develop Framework	RDF/S Repository	Process Editor	Develop Platform	IDE	Sw Project/ Document Mng Tool	Information Data	RDF Query Language	Natural Language Analyzer	Virt. Software	Emulator	
		Type	N°	Connected	RAM quantity	HD capacity	Virtual (*)	CPU N°	OS				J2EE Application Server	BPEL Orchestration Engine	BPMN Engine									SPARQL Query Language			Android Emulator	RFID Reader Emulator
Composition Editor	DT	n.i.	1	n.i	n.i	n.i	YES	1	Linux	n.i	n.i	n.i	Tomcat 6.x	ActiveBPEL	n.i	MySQL	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Composition Editor	EAB	n.i	1	n.i	n.i	n.i	YES	1	n.i	n.i	n.i	n.i	Required	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Composition Editor	UPM	n.i	1	n.i	n.i	n.i	YES	1	n.i	n.i	n.i	Python (2.5, 2.6, or 2.7)	Tomcat 6.x	n.i	n.i	MySQL/ PostgreSQL/ Sqlite3	Django 1.3	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Composition Editor	ATOS	n.i	1	n.i	n.i	n.i	YES	1	n.i	n.i	n.i	n.i	Tomcat 6.x	n.i	Activity 2.0	PostgreSQL 8.3.x	n.i	Sesame	Oryx	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Composition Execution	DT	n.i	1	n.i	n.i	n.i	YES	1	n.i	n.i	n.i	Linux	Tomcat 6.x	ActiveBPEL	n.i	MySQL	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Composition Execution	EAB	n.i	1	n.i	n.i	n.i	YES	1	n.i	n.i	n.i	n.i	Required	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Composition Execution	UPM	n.i	1	n.i	n.i	n.i	YES	1	n.i	n.i	n.i	Python(2.5, 2.6, or 2.7)	Tomcat 6.x	n.i	n.i	MySQL/ PostgreSQL/ Sqlite3	Django 1.3	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Marketplace	SAP	n.i	1	n.i	n.i	n.i	YES	1	Linux	n.i	n.i	n.i	Required	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Mediation	ATOS	n.i	1	n.i	n.i	n.i	YES	1	n.i	n.i	n.i	n.i	Tomcat 6.x	n.i	n.i	n.i	n.i	Sesame	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Mediation	Thales	n.i	1	n.i	n.i	n.i	YES	1	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	Java JDK	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Mediation	Telecom Italia	X86_ 64	1	n.i	n.i	n.i	YES	1	Ubuntu 10.4	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	Java JDK ≥1.6.x	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Registry	SAP	n.i	1	n.i	n.i	n.i	YES	1	Linux	n.i	n.i	n.i	Required	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Repository	SAP	n.i	1	n.i	n.i	n.i	YES	1	Linux	n.i	n.i	n.i	Required	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Revenue Settlement & Sharing System	TID	No, it will be installed in an internal testbed due to IPR concerns. Connectivity must be enabled between TID's internal testbed and FI-WARE's testbed.																										
SLA Management	ENG	The specification is technology independent																										
Store	EAB	n.i	1	n.i	n.i	n.i	NO	n.i	Sun Solaris	n.i	n.i	n.i	n.i	n.i.	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
USDL Service Description	SAP	Not relevant, no deployable software component																										
IaaS DataCenter Resource Management	IBM, Intel	X86	Use Case dep.	yes	Use Case dep	Use Case dep	YES	n.i.	Ubuntu	Switches	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
IaaS Service Management	TID	n.i.	1.	No special requirem.	4GB	≥10GB	YES	2	CentOS 5.1 EAP	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
Object Storage	Intel	n.i.	5	yes	Use Case dep	Use Case dep	YES	1	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
PaaS Management	TID	n.i.	1	No special requirem.	4GB	≥10GB	YES	2	CentOS 5.1 EAP	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.

		n.i.	1	High Network Connection	512Mb	10GB	YES	1	Debian Squeeze	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
		n.i.	1	No special requirem.	512Mb	20GB	YES	1	Debian Squeeze	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
Resource Monitoring	TID	n.i.	1	Network: UDP connection with deployed VMs	512Mb	10GB	YES	1	Fedora 14	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
IoT Communications	Orange	n.i.	2	n.i.	n.i.	n.i.	YES	1	Windows xp or Linux	n.i.	n.i.	n.i.	Tomcat	n.i.	n.i.	MySQL	n.i.	n.i.	n.i.	Fosstrak Platform	Eclipse	Apache Maven	n.i.	n.i.	n.i.	n.i.	Rifidi Emulator
IoT Communications	Telecom Italia	n.i.	1	n.i.	n.i.	n.i.	YES	1 (=> 450 Mhz)	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	MySQL (=>5)	n.i.	n.i.	n.i.	JDK 1.6/OSGI 3.6	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
IoT Communications	Ericsson	n.i.	1	n.i.	16 MB flash for boot image 16 MB flash for file system 64 MB RAM	n.i.	NO: gateway intended to be run locally.	1 (> 300 Mhz)	Linux (≥ 2.4)	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	Java CDC 1.1.2 Java SE (≥1.4.2) OSGi (≥4.1) prefer. Knoplerfish dist.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
IoT data Handling	Orange	n.i.	6	n.i.	2GB	10GB	YES	1	3 Ubuntu server 3 windows 7 or XP	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	Google's Android Emulator	Rifidi Emulator
IoT Process Automation	NEC	n.i.	1	n.i.	n.i.	n.i.	YES	1	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	OSGI	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
IoT Resources Management	TID	n.i.	1	n.i.	n.i.	n.i.	YES	1	n.i.	n.i.	n.i.	n.i.	Required	n.i.	n.i.	MongoDB	n.i.	n.i.	n.i.	n.i.	n.i.	Subversion	n.i.	n.i.	n.i.	n.i.	n.i.
Centralised Backend/Gateway without Exposure GE	NSN	n.i.	1	n.i.	≥ 4GB	n.i.	YES (VirtualBox is a possibility)	1	Centos 6.2 64	n.i.	n.i.	n.i.	Tomcat	n.i.	n.i.	MySQL/ CouchDB	n.i.	n.i.	n.i.	Java JDK 1.6	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
Big Data Analysis	TID	X86-64	3	Yes (for performance reason)	4GB	10GB	YES	1	Ubuntu 11.04 or CentOS 6.0	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
Complex Event Processing	IBM-IL	n.i.	≥ 1	Yes (for performance reason)	4GB	10GB	YES	1	Linux	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
Localization Platform	Thales	It depends of the capacity requirements in terme of request/second. A simple PC can offer enough services for a trial. We will provide a remote connection of our infrastructure, for the test bed.																									
Metadata Preprocessing	Siemens	n.i.	1	n.i.	4GB	n.i.	YES	1	≥Windows xp	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	≥ Microsoft .Net 4.0	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
Multimedia Analysis	Siemens	n.i.	1	n.i.	4GB	n.i.	YES	1	≥Windows xp	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.

Publish/Subscribe Broker	Telecom Italia	n.i.	1	High Network Connection (1GB bandwidth)	2GB	50GB free space	YES	2	CentOS5 server		YES preferred HA (High Aval.) conf.	n.i.	JBoss	n.i.	n.i.	MySQL	n.i.	n.i.	n.i.	Java Se SDK	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
Query Broker	Siemens	n.i.	1	n.i.	4GB	n.i.	YES	1	Windows or Linux	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	≥ JRE 1.6	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
Semantic Annotation	Telecom Italia	n.i.	1	High Network Connection (1GB bandwidth)	16GB	1 TB free space	YES	2	Ubuntu or CentOS server	n.i.	YES preferred HA (High Aval.) conf.	PHP	Tomcat	n.i.	n.i.	MySQL	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	DBpedia dataset	Virtuoso	Freeling	n.i.	n.i.	n.i.
Semantic Application Support	Atos	n.i.	1	n.i.	12GB	1 TB free space	YES	2	Ubuntu or CentOS server	n.i.	n.i.	n.i.	Tomcat and JBoss	n.i.	n.i.	SQL DB required	n.i.	Sesame and OWLIM	n.i.	Java JDK 1.6	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
Cloud Edge	Technicolor	Our GE is the “cloud proxy”, sort of super gateway. The SW is 100% dependant on the HW we will provide. Our plans right now are to provide the 1st release running on a more or less standard small PC, the 2nd and 3rd releases might be more “industrial” being released on a specifically manufactured HW(tbc). The cloud proxy is a terminal, it needs an internet connection and nothing more (xDSL).																										
Connected Devices Interfaces	Intel	Android device																										
Network Information and Control (NetIC)	Alcatel-Lucent Germany	n.i.	1	Yes (connection to external network info source)	n.i	n.i	YES	1	n.i	n.i	n.i.	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Network Information and Control (NetIC)	Alcatel-Lucent Italy	n.i.	1	Yes (connection to a private ip equipped with vmware hypervisor)	n.i.	n.i.	YES	1	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	Java JDK	n.i	n.i	n.i	n.i	n.i	Hypervisor Vmware	n.i	n.i
Network Information and Control (NetIC)	NSN	n.i.	1	n.i.	n.i	n.i	YES	1	n.i	n.i	n.i.	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Service, Capability, Connectivity, and Control (S3C)	Jointly FOKUS (contact point), DT, Ericsson, Orange	n.i.	6	YES (network connections among them)	n.i.	n.i.	YES	1	n.i.	n.i.	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Context-based security and compliance	Atos	n.i.	1	n.i.	n.i.	n.i.	YES	1	n.i.	n.i.	n.i.	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Identity Management	NSN	The Generic Enabler will be connected to the test bed via the Internet. For the test bed we provide a link to our enabler.																										
Identity Management	DT	GE will be executed in a private environment due to business reasons.																										
Data Handling	SAP	n.i.	1	n.i.	n.i.	n.i.	YES	1	n.i.	n.i.	n.i.	n.i	Required	n.i	n.i	MySQL	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
DB Anonimisation	SAP	n.i.	1	n.i.	n.i.	n.i.	YES	1	n.i.	n.i.	n.i.	n.i	Required	n.i	n.i	MySQL	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Security Service Storage	Thales	n.i.	1	n.i.	n.i.	n.i.	YES	1	Linux	n.i.	n.i.	n.i.	Tomcat 6	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	Java JRE	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
Malware Detection	INRIA	GE will be executed in a private environment (not explained for what reason)																										

System																										
Security Monitoring	Thales	n.i.	1	n.i.	n.i.	n.i.	YES	1	n.i.	n.i.	n.i.	n.i.	Required	n.i	n.i	MySQL	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i	n.i
Catalogue	Ericsson	n.i.	1	n.i.	n.i.	n.i.	YES	1	Ubuntu 32bit server edition 10.04,	n.i.	n.i.	Php	Tomcat	n.i.	n.i.	MySQL	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
Forge	Engineering	n.i.	1	n.i.	8GB	100GB	YES	4	CENTOS 5.x 64bits	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	FusionForge + Media Wiki	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
IDE	Engineering	n.i.	1	n.i.	n.i.	n.i.	YES	1	Microsoft Windows or IOS	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	JDK 1.6	Eclipse Indigo	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.
Software Performance Cockpit (SOPECO)	SAP	Provided as DevComE plug-in																										
PROSA	Univ Duisburg-Essen	n.i.	1	n.i.	n.i.	n.i.	YES	1	Microsoft Windows or Linux	n.i.	n.i.	n.i.	Tomcat (optional)	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	JDK 1.6	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.	n.i.

Table 1 - Testbed specification table

(*) we assume to adopt always virtual machines excerpt in that cases it is explicitly requested and justified to avoid virtualization.

Legend

- GEs for which information were not provided for different reasons (e.g. integration starting from second year)
- GEs which impact on the distribution of the testbed infrastructure or on the nature of involved device.

n.i. Not indicated (this means that there is no specific requirement)

5.4 Testbed HW Configuration

A testbed configuration is the hw and network access configuration on top of the physical infrastructure as provided by the dedicated data-center. Such configuration is made through virtualisation.

A possible configuration of the testbed infrastructure can be derived starting from an analysis of the specification document described above. For instance it brings out that only three GEs have specified the required virtual machine type and that it is X86 or X86_64. Since the more specific one is the X86_64 and that the other GEs have not declared other requirements, then it is possible to assume the infrastructure will adopt mainly virtual X86_64 machines. About the required number of machines, instead, the situation is reported below:

Number of GEs	Required Machines
39	1
1	Use Case Dependent
2	6
1	5
1	3
1	2
16	n.i.

The table points out that at least 62 virtual machines (it is assumed a single machine in case of dependency from use case) must be provided so it can be concluded that the testbed core infrastructure fleet will be able to host 62 virtual machines equipped with an X86_64 architecture. However, the virtualisation layer provides enablers that could cope with individual requests for other platforms if required.

Naturally there are a lot of possible considerations about the organization of such virtual machines. For example in the case several GEs require the same operative system, the manager of the infrastructure can decide to install them in the same machine (together with the software artefacts required from each GE) by properly increasing the hardware resources (e.g. RAM, HD, etc…) as well. Anyway such policies cannot be decided in this phase since the impact of the related tuning activity on the performance of the testbed itself would depend on the workload which is not still available. For this reason it will be assumed, for the moment, to have one or more machines for a single GE, on the base of its need (see table above), but not the vice versa.

Others aspects influenced by the information gathered inside the specification document concern the distributed nature of the testbed infrastructure and the nature of the involved devices. For the first one, in fact, the following GEs have expressed the need to be executed on a private testbed.

Revenue Settlement & Sharing System	TID	It will be installed in an internal testbed due to IPR concerns. Connectivity must be enabled between TID's internal testbed and FI-WARE's testbed. Subject to review.
Localization Platform	Thales	It depends of the capacity requirements in terme of request/second. A simple PC can offer enough services for a trial. We will provide a remote connection of our infrastructure for the test bed.
Cloud Edge	Technicolor	Our GE is the “cloud proxy”, sort of super gateway. The SW is 100% dependant on the HW we will provide. Our plans right now are to provide the 1st release running on a more or less standard small PC, the 2nd and 3rd releases might be more “industrial” being released on a specifically manufactured HW(tbc). The cloud proxy is a terminal, it

		needs an internet connection and nothing more (xDSL).
Identity Management	NSN	The Generic Enabler will be connected to the test bed via the Internet. For the test bed we provide a link to this enabler.

Table 2 - GEs requiring a private testbed

About the second one, instead, the GE "Connected Devices Interfaces" has expressed the need to include an android device into the testbed.

The following picture summarise the overall physical infrastructure.

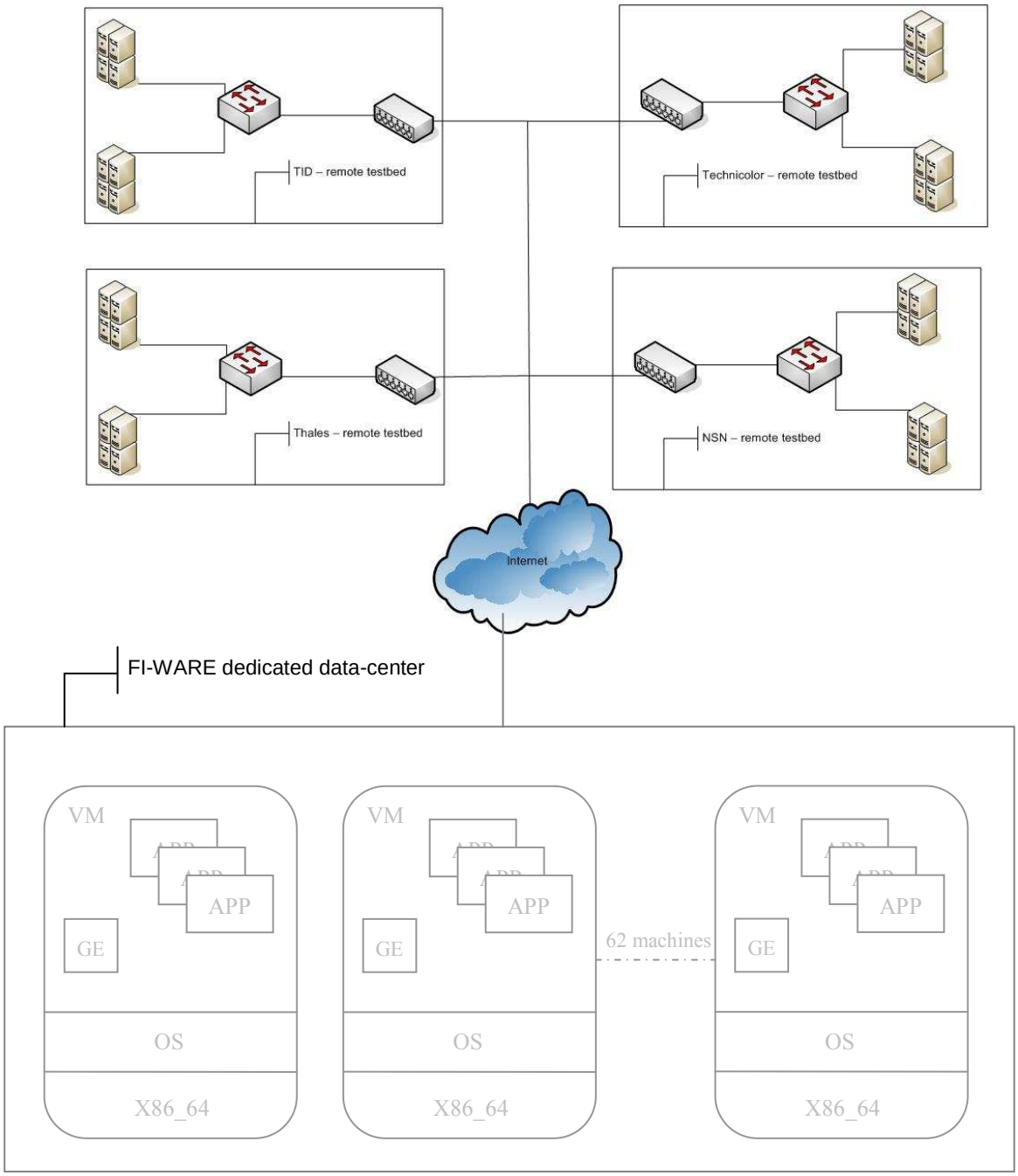


Fig 9 - Testbed infrastructure configuration