

interiot

INTEROPERABILITY
OF HETEROGENEOUS
IOT PLATFORMS.

D2.3

Requirements and business analysis

January 2017

INTER-IoT

INTER-IoT aim is to design, implement and test a framework that will allow interoperability among different Internet of Things (IoT) platforms.

Most current existing IoT developments are based on “closed-loop” concepts, focusing on a specific purpose and being isolated from the rest of the world. Integration between heterogeneous elements is usually done at device or network level, and is just limited to data gathering. Our belief is that a multi-layered approach integrating different IoT devices, networks, platforms, services and applications will allow a global continuum of data, infrastructures and services that will enable different IoT scenarios. As well, reuse and integration of existing and future IoT systems will be facilitated, creating a defacto global ecosystem of interoperable IoT platforms.

In the absence of global IoT standards, the INTER-IoT results will allow any company to design and develop new IoT devices or services, leveraging on the existing ecosystem, and bring them to market quickly.

INTER-IoT has been financed by the Horizon 2020 initiative of the European Commission, contract 687283.

Requirements and business analysis

Version: Final

Security: Public

January 31. 2017

The INTER-IoT project has been financed by the Horizon 2020 initiative of the European Commission, contract 687283



Disclaimer

This document contains material, which is the copyright of certain INTER-IoT consortium parties, and may not be reproduced or copied without permission.

The information contained in this document is the proprietary confidential information of the INTER-IoT consortium (including the Commission Services) and may not be disclosed except in accordance with the consortium agreement.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the project consortium as a whole nor a certain party of the consortium warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, and accepts no liability for loss or damage suffered by any person using this information.

The information in this document is subject to change without notice.

Executive Summary

The aim of the Requirements documentation is to provide an analysis of the requirements needed for the design and implementation of the different products of INTER-IoT¹, defined in the proposal: INTER-LAYER, INTER-FW, INTER-METH, INTER-LogP, and INTER-Health.

In order to extract the fundamental requirements of the system, we have conducted interviews with several stakeholders such as end users, product providers, suppliers and developers. Based on such gathered information, the requirements were defined to describe how the system works and what should do. The requirements were classified into two main types: functional and non-functional.

The definition of requirements is an important step in any project, due several benefits that they can provide to the project. The first one is to reduce the development effort, since the definition of rigorous requirements before the design can reduce later redesign, recoding, and retesting. Moreover, the requirements can be considered as an agreement between the customers and the suppliers about the product (to be developed), for instance, facilitating the business model and marketing. Furthermore, a detailed description of the requirements can accurately estimate costs and time planning. Finally, the requirements can set the evaluation and validation criteria to obtain a quality product.

WP2 as a whole and specifically this task has been developed using the VOLERE² methodology which is an excellent methodology to extract conclusions and provide results following a systematic approach. The methodology is explained in the deliverable, in order to provide the required background to understand the work developed in WP2 that will be completed in the following deliverable. The deliverable is completed by an annex with the templates that contains all the features of the requirements. The annex is included so that the deliverable is self-contained; moreover, the JIRA³ tool is being extensively used in order to support the VOLERE methodology, and the information is available for internal use by the consortium.

This deliverable is the result of the activity carried out in T2.3. It also uses the results obtained in T2.1, the state of the art of IoT platforms, systems and devices, and the interviews with the stakeholders. The results of this deliverable will be used in T2.4 for the use cases and in the design of INTER-LAYER, INTER-FW and INTER-METH in WP3, WP4 and WP5, respectively.

The current version of the deliverable is a revised document from the one released in September 2016, after the comments received from the expert reviewers in the technical review of the project held in Vienna (Austria) in October 2016.

¹ <http://www.inter-iot.eu>

² <http://www.volere.co.uk/>.

³ <https://es.atlassian.com/software/jira>

List of Authors

Organisation	Authors	Main organisations' contributions
VPF	Miguel Llop, Pablo Giménez, Alexandre Sánchez, M ^a Luisa Escamilla, Eduardo Olmeda	Initial version Methodology INTER-LogP and General requirements Conclusion Quality review
UPV	Carlos E. Palau, Benjamín Molina, Eneko Olivares, Regel González-Usach, Andreu Belsa	INTER-LAYER requirements INTER-FW requirements
UNICAL	Giancarlo Fortino, Wilma Russo, Gianluca Aloï, Pasquale Pace, Raffaele Gravina	INTER-METH requirements INTER-Health requirements Review
PRO	Miguel Montesinos, Christophe Joubert, Amelia del Rey, Miguel A. Llorente	INTER-FW requirements
TU/e	George Exarchakos, Antonio Liotta	INTER-LAYER requirements
XLAB	Mariano Cecowski, Robert Plestenjak	INTER-FW requirements INTER-METH requirements
SRIPAS	Katarzyna Wasielewska-Michniewska, Paweł Szmeja, Wiesław Pawłowski	INTER-METH requirements INTER-LAYER requirements
RINICOM	Garik Markarian Eric Carlson	INTER-Health requirements
ASL TO 5	Margherita Gulino, Claudio Maggi, Angelina Della Torre, Domenica Pata, Monica Minutolo, Anna Aldrighetti, Bartolomeo Avataneo	INTER-Health requirements
TI	Carlo Aldera, Alberto Delpiano, Fabio D'Ercoli, Giovanna Larini	INTER-Health requirements
NEWAYS	Ron Schram, Roel Vossen, Johan Schabbink, Frans Gevers	INTER-LAYER requirements Review
AFT	Moncef Semichi, Sarah Koneke, Kimvy Bui	INTER-LogP requirements
NOATUM	Francisco Blanquer	INTER-LogP requirements
ABC	Alessandro Bassi, Jitka Slechtova	INTER-FW requirements INTER-METH requirements

Change control datasheet

Version	Changes	Chapters	Pages
0.1	Creation, index and introduction	All	18
0.2	Methodology	2	32
0.3	INTER-LAYER requirements	3	56
0.4	INTER-FW requirements	3	71
0.5	INTER-METH requirements	3	82
0.6	INTER-LogP requirements	3	93
0.7	INTER-Health requirements	3	106
0.8	General requirements	3	113
0.9	Conclusions	4	115
1.0	Final version	All	116
1.1	Internal review 1	All	116
1.2	Internal review 2	All	116
1.3	Final version to upload	All	116
2.1	Review requirements	All	125
2.2	Internal review	All	125
2.3	Final version	All	125
2.4	Submitted Version with requested changes after the technical review	All	124

Contents

Executive Summary	3
List of Authors	4
Change control datasheet	5
Contents	6
List of Figures	8
Acronyms	9
1 Introduction	10
1.1 Internet of Things	10
1.2 IoT interoperability	11
1.3 Scope of the INTER-IoT project	13
1.4 Scope of the document	16
2 Methodology	17
2.1 Requirements	18
2.1.1 Characteristics of requirements	18
2.1.2 Types of Requirements	19
2.1.3 Requirements Harmonisation process steps	25
2.1.4 Priority of requirements	27
2.2 JIRA Repository	28
3 Requirement analysis	31
3.1 INTER-LAYER requirements	31
3.1.1 Non-functional requirements	31
3.1.2 Functional requirements	41
3.1.3 Requirements by type	50
3.1.4 Analysis	53
3.2 INTER-FW requirements	57
3.2.1 Non-functional requirements	57
3.2.2 Functional requirements	62
3.2.3 Requirements by type	68
3.2.4 Analysis	70
3.3 INTER-METH requirements	74
3.3.1 Non-functional requirements	74
3.3.2 Functional requirements	78
3.3.3 Requirements by type	81
3.3.4 Analysis	83

3.4	INTER-LogP requirements.....	87
3.4.1	Non-functional requirements.....	87
3.4.2	Functional requirements	88
3.4.3	Requirements by type	91
3.4.4	Analysis	92
3.5	INTER-Health requirements.....	96
3.5.1	Non-functional requirements.....	96
3.5.2	Functional requirements	99
3.5.3	Requirements by type	106
3.5.4	Analysis	107
3.6	General requirements	111
3.6.1	Non-functional requirements.....	111
3.6.2	Requirements by type	115
3.6.3	Analysis	116
4	Conclusions	119

List of Figures

Figure 1: IoT Installed Base and Revenues in EU 28 2013-2020	11
Figure 2: INTER-IoT approach abstract schema	14
Figure 3: Abstract schema of the INTER-FW	14
Figure 4: Kinds of non-functional requirements	21
Figure 5: Requirements capture methodology.....	25
Figure 6: JIRA home page of the INTER-IoT project	29
Figure 7: View issues	29
Figure 8: Create new issue	30
Figure 9: INTER-LAYER requirements by type	53
Figure 10: INTER-LAYER requirements by category	54
Figure 11: INTER- LAYER requirements by priority	54
Figure 12: INTER- LAYER requirements by priority and category	55
Figure 13: INTER-LAYER requirements by MoSCoW priority	55
Figure 14: INTER- LAYER requirements by source	56
Figure 15: INTER-FW requirements by type	70
Figure 16: INTER-FW requirements by category.....	71
Figure 17: INTER-FW requirements by priority	71
Figure 18: INTER-FW requirements by priority and category	72
Figure 19: INTER-FW requirements by MoSCoW priority	72
Figure 20: INTER-FW requirements by source	73
Figure 21: INTER-METH requirements by type.....	83
Figure 22: INTER-METH requirements by category	84
Figure 23: INTER-METH requirements by priority	84
Figure 24: INTER-METH requirements by priority and category.....	85
Figure 25: INTER- METH requirements by MoSCoW priority.....	85
Figure 26: INTER-METH requirements by source	86
Figure 27: INTER-LogP requirements by type.....	92
Figure 28: INTER-LogP requirements by category.....	93
Figure 29: INTER-LogP requirements by priority.....	93
Figure 30: INTER-LogP requirements by priority and category	94
Figure 31: INTER-LogP requirements by MoSCoW priority.....	95
Figure 32: INTER-LogP requirements by source.....	95
Figure 33: INTER-Health requirements by type.....	107
Figure 34: INTER- Health requirements by category	108
Figure 35: INTER- Health requirements by priority	108
Figure 36: INTER-Health requirements by MoSCoW priority.....	109
Figure 37: INTER- Health requirements by source	109
Figure 38: General requirements by type	116
Figure 39: General requirements by category	117
Figure 40: General requirements by priority	117
Figure 41: General requirements by MoSCoW priority	118
Figure 42: General requirements by source	118

Acronyms

AIOTI	Alliance for Internet of Things Innovation
API	Application Programming Interface
BIP	Best Ideas and Projects
CAN	Controller Area Network
CRUD	Create, Read, Update and Delete
DSL	Digital Subscriber Lines
EC	European Commission
GOIoT	Generic Ontology for IoT Platforms
IEEE	Institute of Electrical and Electronics Engineers
IERC	European Research Cluster on the Internet of Things
INTER-LAYER	INTER-IoT Layer integration tools
INTER-FW	INTER-IoT Interoperable IoT Framework
INTER-METH	INTER-IoT Engineering Methodology
INTER-LogP	INTER-IoT Platform for Transport and Logistics
INTER-Health	INTER-IoT Platform for Health monitoring
INTER-META-ARCH	INTER-IoT Architectural meta-model for IoT interoperable platforms
INTER-META-DATA	INTER-IoT Metadata-model for IoT interoperable semantics
INTER-API	INTER-IoT Programming library
INTER-CASE	INTER-IoT Computer Aided Software Engineering tool for integration
INCOSE	The International Council on Systems Engineering
IoT	Internet of Things
IOT-A	Internet of Things - Architecture
ISO	International Organization for Standardization
ITU	International Communications Union
LTE	Long-Term Evolution networks
M2M	Machine to Machine
MAC	Media Access Control address
OGC	Open Geospatial Consortium
RFID	Radio Frequency IDentification
SAREF	Smart Appliances REFerence
SDO	Standard Development Organisation
SDR	Software Defined Radio
SPEM	Software and Systems Process Engineering Meta-model
SDN	Software Defined Networking
SSN	Semantic Sensor Network
W3C	World Wide Web Consortium

1 Introduction

1.1 Internet of Things

The connection of intelligent devices, equipped with a growing number of electronic sensors and/or actuators, via the Internet, is known as the 'Internet of Things' (IoT). With the IoT, every physical and virtual object can be connected to other objects and to the Internet, creating a fabric of connectivity between things and between humans and things. The IoT is now widely recognised as the next step of disruptive digital innovation.

The International Communications Union (ITU) and the European Research Cluster on the Internet of Things (IERC) provide the following definition: IoT is a dynamic global network infrastructure, with self-configuring capabilities based on standard and interoperable communication protocols, where physical and virtual "things" have identities, physical attributes and virtual personalities and use intelligent interfaces. All of them seamlessly integrated into the information network.

The design of the Internet and specifically the extension of the Internet to the IoT, rely on the convergence of the infrastructure with software and services. A common practice is required to think/design cross solutions between software and infrastructure in order to provide integrated solutions for some of the complex problems in the current and future systems. In the IoT environment this convergence is evident, and the continuous evolution generates more and more smart connected objects and platforms that are embedded with sensors and their respective associated services, in some cases considering virtualization.

IoT is the network or overlay associations between smart connected objects (physical and virtual), that are able to exchange information by using an agreed method (including protocols) and a data schema. IoT deployments are increasing, the same applies to standards, alliances and interest for homogenization. All of this is giving a strong push to the IoT domain to be considered as one of the most promising emerging technologies. As an example, Gartner (one of the world's leading information technology research and advisory company) estimates the number of web-connected devices will reach 25 billion by 2020. In other words, more devices, appliances, cars, artefacts, and accessories will be connected and will communicate with each other, and with other objects, thus bringing amplified connectivity and better supply chain visibility. The applications of the IoT are numerous i.e. every object could be transformed into a smart object that sends several valuable information to other devices. As an example, in the port industry IoT could be applied to shipping containers, the equipment that handles them, the trucks that carry them and, even, the ships that move them around the globe.

According to the European Commission (EC) the IoT represents the next step towards the digitisation of our society and economy, where objects and people are interconnected through communication networks, and report about their status and/or the surrounding environment. Furthermore, IoT can also benefit the European economy generating economic growth and employment; according to a recent European Commission study revenues in the EU28 will increase from more than €307 billion in 2013 to more than €1,181 billion in 2020 (as shown in Figure 1).

IoT is an emerging area that not only requires development of infrastructure but also deployment of new services capable of supporting multiple, scalable and interoperable applications. The focus is today associated with cloud deployments, virtualizations and the elimination of silos avoiding the existence of application domain specific developments, AIOTI and EC are pressing in this line. IoT

has evolved from sensor networks and wireless sensor networks to a most clear description and definition referring to objects and the virtual representations of these objects on the Internet and associated infrastructures. It defines how the physical things and virtual objects will be connected through the Internet and their interaction, and how they communicate with other systems and platforms, in order to expose their capabilities and functionalities in terms of services and accessibility through open APIs and frameworks. IoT is not only linking connected devices by the Internet; it is also web-enabled data exchange in order to enable systems with more capacities to become smart and accessible, creating webs of objects and allowing integration of data, services and components.

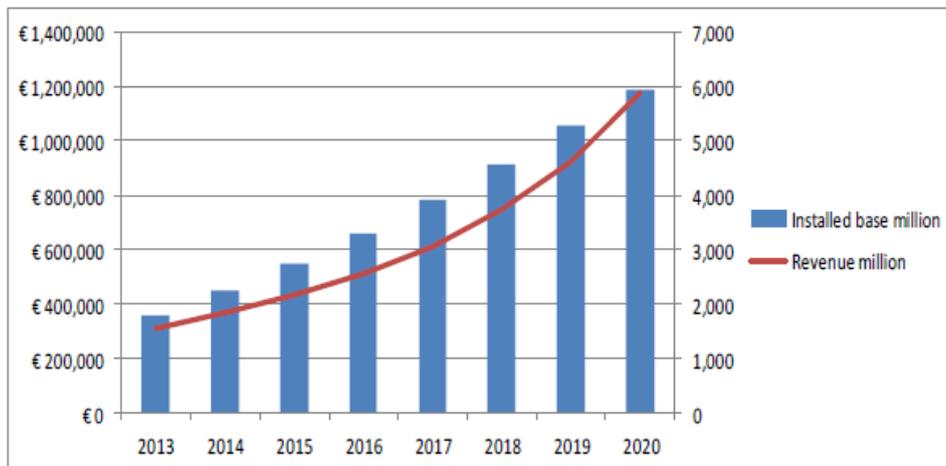


Figure 1: IoT Installed Base and Revenues in EU 28 2013-2020

There are several challenges associated with IoT and its evolution, but one major issue is related with interoperability. IoT is mainly supported by continuous progress in wireless sensor and actuator networks and by manufacturing low cost and energy efficient hardware for sensor and device communications. However, heterogeneity of underlying devices and communication technologies and interoperability in different layers, from communication and seamless integration of devices to interoperability of data generated by the IoT resources, is a challenge for expanding generic IoT solutions to a global scale, with the further aim of avoiding silos and provide solutions that are application domain agnostic, like those proposed in INTER-IoT.

1.2 IoT interoperability

Many projects have dealt and/or are dealing with the development of IoT architectures in diversified application domains. However, the conceptual realization of IoT is far from achieving a full deployment of converged IoT services and technology. The widespread of vertically-oriented closed systems, architectures and application areas has generated a fragmentation that needs to be overcome. The lack of interoperability causes major technological and business issues such as impossibility to plug non-interoperable IoT devices into heterogeneous IoT platforms, impossibility to develop IoT applications exploiting multiple platforms in homogeneous and/or cross domains, slowness of IoT technology introduction at a large-scale, discouragement in adopting IoT technology, increase of costs, scarce reusability of technical solutions and user dissatisfaction. Current research in IoT is focused on providing integrated solutions and primarily on the feature that enables convergence or what is called as Interoperability.

Interoperability is a property referring to the ability of systems and organizations to work together. The overall challenge of achieving interoperability of heterogeneous IoT platforms is to deliver an IoT extended into a web of platforms for connected devices and objects. They will support smart environments, businesses, services and people with dynamic and adaptive configuration capabilities. Interoperability of heterogeneous IoT platforms will be the way to achieve the potential benefits derived from a scenario where everything is linked; interoperability between several heterogeneous platforms is of utmost importance.

Interoperability can be generalized as the feature for providing seamless exchange of information to, for example, customize services automatically or simply exchanging information in a way that other systems can use it for improving performance, enabling and creating services, controlling operations and processing information. This type of scenarios requires increased interoperability in service management operations. The INTER-IoT project, aware of this fact, aims to provide an interoperable open IoT framework (with associated engineering tools and methodology) for seamless integration of heterogeneous IoT platforms available in the same or different application domains.

INTER-IoT will provide all the building blocks needed to achieve interoperability, including a framework, methodology and associated APIs and tool-boxes. Ensuring that interoperability will be kept as the different products and architectures may evolve in the market. The benefits of INTER-IoT will be:

- At the **device level**, seamless inclusion of novel IoT devices and their interoperation with already existing, even heterogeneous ones. This will allow fast growth of smart objects ecosystems.
- At the **networking level**, seamless support for smart objects mobility and information routing. This will allow design and implementation of fully connected ecosystems.
- At the **middleware level**, seamless service discovery and management system for smart objects and their basic services. This will allow global exploitation of smart objects in large (even extreme) scale (multi-platform) IoT systems.
- At the **application service level**, reuse and exchange (import/export) of heterogeneous services between different IoT platforms.
- At the **data and semantics level**, common interpretation of data and information based on global shared ontology in order to achieve semantic interoperability.
- At the **integrated IoT platform level**, rapid prototyping of cross-platform IoT applications.
- At the **business level**, faster introduction of IoT technology and applications across multiple application domains.

By using the aforementioned approach, IoT platform heterogeneity will be turned from a crucial problem to a great advantage, as there will be no need to wait for a unique standard for an interoperable IoT. Instead, interoperable IoT, even on a very large scale, will be created through a bottom-up approach.

1.3 Scope of the INTER-IoT project

INTER-IoT project aims at the design, implementation and experimentation of an open cross-layer framework, an associated methodology and tools to enable voluntary interoperability among heterogeneous Internet of Things (IoT) platforms. The proposal will allow effective and efficient development of adaptive, smart IoT applications and services, atop different heterogeneous IoT platforms, spanning single and/or multiple application domains. The project and associated approach has been defined to be use case-driven. And it will be implemented and tested in three realistic large-scale pilots:

- Port of Valencia transportation and logistics involving heterogeneous platforms with ~400 smart objects.
- An Italian National Health Center for mobile health involving ~200 patients, equipped with body sensor networks with wearable sensors and mobile smart devices.
- A cross-domain pilot involving IoT platforms from both application domains will be deployed and tested in the premises of the Port of Valencia.

Furthermore, the project will analyse usability of the provided solutions from the perspective of IoT platform creators, IoT platform owners, IoT application programmers and users investigating business perspectives and creating new business models. The most important benefits expected for third parties are related with the new features and components that will be released by the consortium: Methodologies, tools, protocols and APIs that will be released as open items available to develop new applications and services. The variety and cross availability of the results could be used to build and integrate services and platforms at different layers according to the needs of the stakeholders and developers. The availability of more and new data will stimulate the creation of new opportunities and products, always in the scope of open interoperability.

Open interoperability relies on the promise of enabling vendors and developers to interact and interoperate, without interfering with anyone's ability to compete by delivering a superior product and experience. In the absence of global IoT standards, the INTER-IoT project will support and make it easy for any company to design IoT devices, smart objects, or services and get them to market quickly, and create new IoT interoperable ecosystems.

The INTER-IoT approach is general-purpose and may be applied to any application domain and across domains in which there is a need to interconnect IoT systems already deployed or add new ones. INTER-IoT will be based on three main building blocks:

- Methods and tools for providing interoperability among and across each layers of IoT platforms (INTER-LAYER);
- Global framework (INTER-FW) for programming and managing interoperable IoT platforms; and
- Engineering Methodology (INTER-METH) based on CASE tool for IoT platforms integration/interconnection.

The project results will be specifically tested in the two independent application domains that will lead to two independent products, namely: INTER-LogP and INTER-Health. Thus, as an outcome of the project, INTER-IoT will provide these five products that could be introduced in the market for a wider implementation and exploitation. The market analysis and stakeholders will be based in the existence of these five products, and the interest generated by the stakeholders.

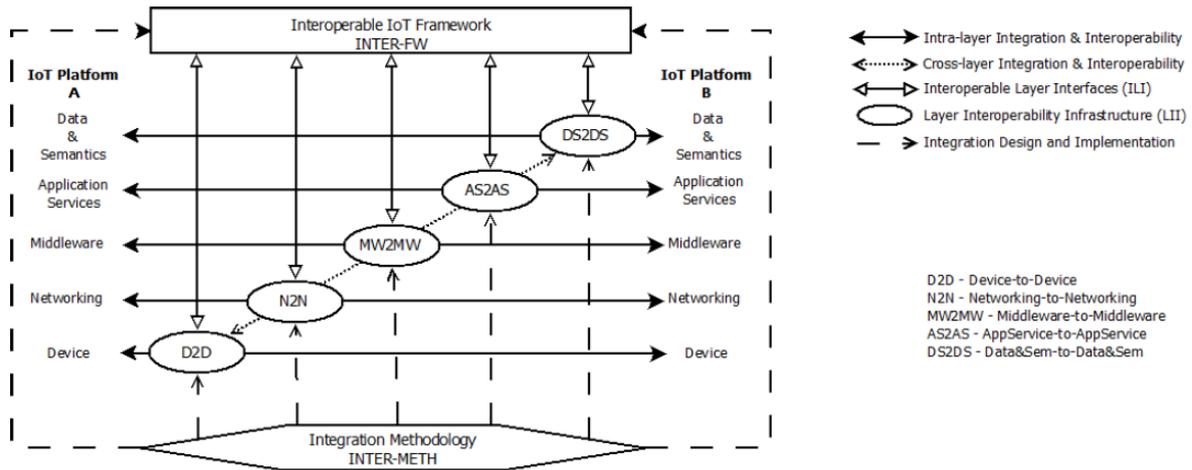


Figure 2: INTER-IoT approach abstract schema

INTER-LAYER

INTER- IoT uses a layer-oriented approach to fully exploit specific functionalities of each layer (device, networking, middleware, application services, data & semantics) (see Figure 2). Although the development of a layer-oriented approach is a research challenge, as compared to a global approach, it has a higher potential to deliver tight bidirectional integration among heterogeneous IoT platforms, notably guaranteeing independence, thus providing higher performance, modularity and reliability and, what is extremely important, more control on functional and non-functional requirements. In addition, the data and semantics level provides a global shared ontology and methods in order to achieve IoT platform semantic interoperability.

INTER-LAYER includes the design of device-to-device interaction based on multiprotocol/access mechanisms, the design of software defined interoperable modules for mobility and routing, the development of an open service discovery and management framework for smart objects, the design and implementation of smart IoT application service gateway and virtualization and the definition of a common ontology for IoT platform semantic interoperability.

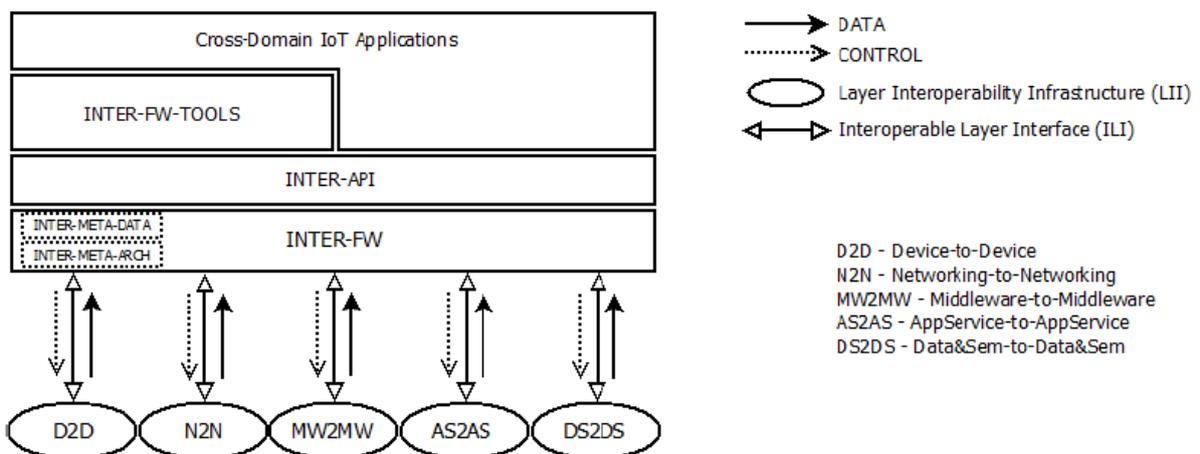


Figure 3: Abstract schema of the INTER-FW

INTER-FW

The Interoperability IoT Framework (INTER-FW) aims at providing global and open platform-level interoperability among heterogeneous IoT platforms coupled through specifically developed Layer Interoperability Infrastructures (LIIs) and Interoperability Layer Interfaces (ILIs). INTER-FW will rely on an architectural meta-model for IoT interoperable platforms, on a metadata-model for IoT interoperable semantics and it will provide a programming API and tools allowing global-level management of the integrated IoT platforms.

Figure 3 shows the abstract schema of the INTER-FW. INTER-FW will advance the state-of-the-art by providing a general and effective method for inter-platform interoperability, addressing at a global level: real-timeless, reliability, security, privacy and trust. In particular, INTER-FW will thoroughly address privacy and security-related risks and challenges resulting from the use of IoT devices.

INTER-METH

The engineering methodology INTER-METH aims at defining a systematic methodology supporting the integration process of heterogeneous IoT platforms to obtain interoperability among them and allowing implementation and deployment of IoT applications at the top of them. It is widely recognized that using an engineering methodology is fundamental in any engineering application domain (e.g. software engineering, co-design hardware/software, civil engineering, etc.). The manual and non-systematic application of complex techniques, methods and frameworks would very likely lead to an increase of the degree of errors during integration. INTER-METH includes a Computer Aided Software Engineering tool for integration (INTER-CASE).

INTER-IoT considers two application domains: transportation and logistics in a port environment and m-health. Around these two application domains, three use cases will be built and packaged as products of the project:

1. INTER-LogP for “Smart Port Transportation for Containers and Goods”;
2. INTER-Health for “Decentralized and Mobile Monitoring of Assisted Livings” and
3. INTER-DOMAIN in which IoT platforms from both application domains plus some additional ones will be integrated.

INTER-LogP

INTER-LogP use case illustrates the need to achieve seamlessly interoperability of different heterogeneous IoT platforms, oriented to port transport and logistics. The considered application domain identifies several physical transport entities (trucks, containers, semi-trailers, cranes, tractors and other container handling machines) owned by different companies. The possibility to capture in real time sensor-based data coming from these physical moving assets and connecting them to transport and logistic infrastructures, is an opportunity to drive optimal real-time execution as well as automation of transport and logistics operations. The capture and sharing of real time sensor-based data across different organisations is today a big challenge as there is not any solution in the market able to attend this need and overcoming the complexity of implementing IoT solutions connecting different sensors, systems and products. Sensor-based technology is already being pushed by the transportation and logistics industry. However, what it is lacking is the ability to effectively capture and share the data relative to the movement of vehicles and goods and convert it into actionable insights capable of driving improvements across the supply chain. The lack of use of IoT oriented platforms and their interoperability is today a main obstacle.

For example, almost any person, truck, machine and equipment have been outfitted or it is relatively easy to do so with GPS devices and other sensors to capture information such as location, speed and idle time. With this information, companies have been able to compile and assess several indicators like delivery times, fuel consumption or emissions. However, these companies are not able to design and establish connections with platforms managed by other operators in the supply, logistics and transport chains. The global and interconnected nature of today's supply chains needs a greater collaboration among supply chain partners. The interoperability of heterogeneous IoT platforms can provide a framework for real-time multidirectional information sharing to help in creating true supply chain collaboration.

INTER-Health

INTER-Health scenario for Decentralized and Mobile Monitoring of Assisted Livings' Lifestyle aims at developing an integrated IoT system for monitoring humans' lifestyle in a decentralized way and in mobility, to prevent health issues mainly resulting from food and physical activity disorders. By exploiting the integrated system - INTER-Health - the patient's monitoring process can be decentralized from the healthcare centre to the monitored subjects' homes, and supported in mobility by using on-body physical activity monitors.

The INTER-DOMAIN solution has not yet been considered as an initial product to be offered since its requirements and domain is still unknown until the open call takes place. Only when a couple of third party entities with the clear goal of fostering the adoption of INTER-IoT developments are selected, the INTER-DOMAIN could be considered as a product to be offered to the market.

1.4 Scope of the document

This deliverable provides the INTER-IoT requirements, which support the design and the implementation of the different products identified in the project.

In this task the partners have identified all the requirements necessary to begin the development of the different components that form INTER-IoT. In this document we have included the requirements identified until the delivery date of this task, but we use JIRA to complete and update the requirements throughout the project.

For the selection of the requirements, different criteria have been considered. An important input is the needs provided by the stakeholders in the interviews, as they will be the final users of the project results. Additionally, we have been reviewing the most important requirements of other IoT systems and projects. Finally, we have taken into account the wide experience of the partners.

After the first deliverable review, we have reviewed and redefined all the requirements to be more focused on the real development of the different components, since we have a deeper knowledge of the use cases defined. In addition, a prioritization of the requirements has been carried out, which allows establishing a first phase with the main functionalities of the system.

The document is divided into four sections. The first is an introduction of what is Internet of Things, the purpose of the INTER-IoT project and the description of this document. The methodology for defining the requirements and their maintenance is described in the second section. In the third section, all the requirements, grouped by the products defined in the project, are listed and analyzed. Finally, in section four, an overall conclusion is provided.

2 Methodology

The methodology that has been used as a reference for most of the tasks involved in Work Package 2 (WP2) is Volere.

Volere has been used by thousands organizations around the world in order to discover, define, communicate and manage all the necessary requirements for any type of system development (e.g. software, hardware, commodities, services, organizational, etc.). Volere can be applied in almost all kinds of development environments, with any other development methods or with most requirements tools and modelling techniques. To produce accurate and unambiguous requirements, the Volere methodology uses techniques that are based on experience from worldwide business analysis projects, and are continually improved.

The Volere methodology provides several templates to deal with the different techniques and activities that it includes. In a quick view, the Volere Requirement Process⁴ suggests a methodology that can be summarised as follows:

1. Define the Purpose of the Project (Proposal)
2. Stakeholders Identification and Analysis (T2.1)
3. Business Use Cases (T2.4)
4. Scenarios (T2.4)
5. Writing the Requirements: functional requirements and non-functional requirements (T2.3. The methodology is described in section 2.1.2 of this document).
6. Validation of requirements: completeness, relevance, testability, coherency, traceability, and several other qualities before such requirements are used by developers (T2.3. The methodology is described in the section 2.1.3 of this document).
7. Communicating the Requirements (Internally, this activity is described in the section 2.2 of this document. In addition, a set of dissemination activities are carried out about the identified requirements in the WP8).
8. Requirements Completeness (WP3, WP4, WP5 and WP6). The following tasks will examine in depth the requirements collected in this task T2.3 for continuous improvement.

Thus in WP3 (Layer Interoperability), the INTER-LAYER requirements will be reviewed and improved, the INTER-FW requirements will be reviewed in the WP4 (Interoperability Framework API), the WP5 (Inter Methodology from Analysis to Deployment) will deal with INTER-METH requirements. And finally, the INTER-LogP and INTER-Health requirements will be improved during the WP6 (Integration and Pilot Deployment).

The INTER-IoT Project consortium as a whole considered that choosing this methodology could help us to achieve our goals and the ICT30 objectives. Applying Volere for the requirement discovery process is essential to ensure that we are solving the real problem. Also to make our products more attractive and more appropriate if they are to be noticed, bought, used and valued in the whole European territory. In addition, the INTER-IoT partners consider that, to be excellent and successful

⁴ “Volere Requirements: How to Get Started “<http://www.volere.co.uk/pdf%20files/VolereGettingStarted.pdf>

in the development, it is imperative to go deeply into how we understand our customer organizations, and how we find better solutions by discovering and communicating a better understanding of the problem.

All requirements described in this document have been identified during the first phase of the project. As the project will progress, these requirements might be subject to improvements. If new requirements appear, they will be added. That is the reason why it was decided to use JIRA as a repository where having updated risks, stakeholders, products, scenarios, requirements and use cases.

2.1 Requirements

Various definitions exist of what a requirement is. In this study, we agreed to use the definitions of ISO and INCOSE:

“A requirement is Statement that identifies a product (includes product, service, or enterprise) or process operational, functional, or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for product or process acceptability.” (ISO/IEC 2007)⁵

“A requirement is a statement that identifies a system, product or process characteristic or constraint, which is unambiguous, clear, unique, consistent, stand-alone (not grouped), and verifiable, and is deemed necessary for stakeholder acceptability.” (INCOSE 2010)⁶

2.1.1 Characteristics of requirements

The characteristics of good requirements are variously stated by different writers.

There are several characteristics of requirements that are used to aid in the development of a solution and to verify the implementation of these requirements into the solution (ISO 2011, Sections 5.2.5 and 5.2.6), namely:

- *Necessary*

The requirement defines an essential capability, characteristic, constraint, and/or quality factor. If it is not included in the set of requirements, a deficiency in capability or characteristic will exist, which cannot be fulfilled by implementing other requirements.

- *Appropriate*

The specific intent and amount of detail of the requirement is appropriate to the level of the entity to which it refers (level of abstraction). This includes avoiding unnecessary constraints on the architecture or design to help ensure implementation independence to the extent possible.

- *Unambiguous*

⁵ ISO/IEC. 2007. Systems and Software Engineering -- Recommended Practice for Architectural Description of Software-Intensive Systems. Geneva, Switzerland: International Organization for Standards (ISO)/International Electrotechnical Commission (IEC), ISO/IEC 42010:2007.

⁶ INCOSE. 2010. Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. Version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.1: 362.

The requirement is concisely stated. It expresses objective facts, not subjective opinions. It is subject to one and only one interpretation.

- *Complete*

The requirement sufficiently describes the necessary capability, characteristic, constraint, or quality factor to meet the entity need without needing other information to understand the requirement.

- *Singular*

The requirement should state a single capability, characteristic, constraint, or quality factor.

- *Feasible*

The requirement can be realized within entity constraints (e.g., cost, schedule, technical, legal, or regulatory) with acceptable risk.

- *Verifiable*

The requirement is structured and worded in such a way that it is possible to verify its accomplishment, as well as the degree of customer's satisfaction regarding its realization.

- *Correct*

The requirement must be an accurate representation of the entity need from which it was transformed.

- *Consistent*

The requirement does not contradict any other requirement and is fully consistent with all authoritative external documentation.

- *Comprehensible*

The set of requirements must be written such that it is clear as to what is expected by the entity and its relation to the system of which it is a part.

2.1.2 Types of Requirements

The Volere methodology⁷ categorises requirements into several groups:

- Functional requirements are the fundamental subject matter of the system and are measured by concrete means like; data values, decision-making logic and algorithms.
- Non-functional requirements are the behavioural properties that the specified functions must have, such as performance, usability, etc. Non-functional requirements can be assigned to a specific measurement. The methodology also includes a rich catalogue of non-functional requirements to be taken into account which will be reviewed afterwards.
- Project constraints identify how the eventual product must fit into the world. For example, the product might have to interface with or use some existing hardware, software or business practice, or it might have to fit within a defined budget or be ready by a defined date.
- Project drivers are the business-related forces. For example, the purpose of the product is a project driver, as are all of the stakeholders - each for different reasons.

⁷Volere Requirements Specification Template https://www.st.cs.uni-saarland.de/edu/se/2009/slides/volere_specification_template_v6.pdf

- Project issues define the conditions under which the project will be done. We include these in the requirements specification to present a coherent picture of all the factors that contribute to the success or failure of the project.

In this document, we focus on the first two described groups: functional and non-functional requirements. The other groups will be looked at WP1 throughout the life of the project.

2.1.2.1 Functional requirements

As indicated in the previous point, the Volere methodology defines functional requirements as the fundamental subject matter of the system: an action that the product must be able to take, something that the product must do.

Volere Methodology classifies them in the following two groups:

- Functional Requirements: To specify the details for each individual functional requirement, that must be supported by the system.
- Data Requirements: A specification of the essential subject matter/business/objects/entities/classes, which are germane to the system.

These requirements clarify the system's subject matter and thereby trigger requirements that have not yet been thought of.

The functional requirements are highly dependent on products defined in INTER-IoT: INTER-LAYER, INTER-FW, INTER-METH, INTER-LogP and INTER-Health.

2.1.2.2 Non-functional requirements

Regarding non-functional requirements, the Volere methodology indicates that they are the behavioural properties that the specified functions must have, such as performance, usability, etc. Non-functional requirements describe how the system works and properties that the final product needs to have.

The Volere Methodology includes a catalog of non-functional requirements classified in eight groups, namely:

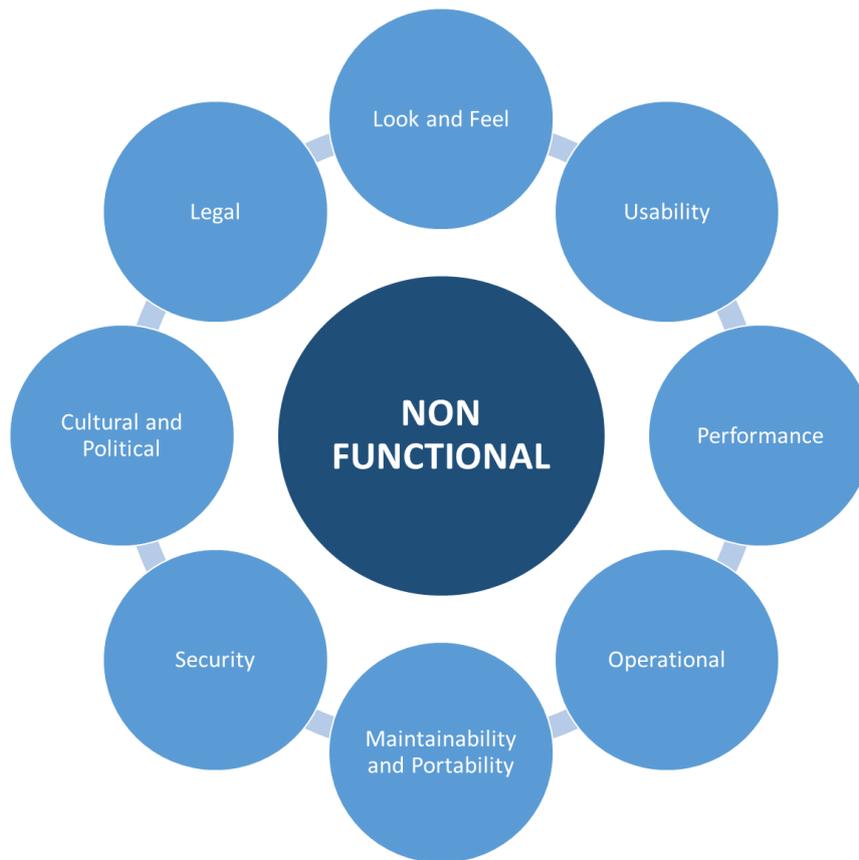


Figure 4: Kinds of non-functional requirements

- Look and Feel Requirements:
 - Interface: to ensure the appearance of the product. There are requirements relating to the interface, such as corporate branding, style, colours to be used, degree of interaction and so on.
These requirements capture the needs for interface to ensure that the appearance of the product conforms to the organization's expectations.
 - Style of the product: a description of salient features of the product that are related to the way a potential customer will see the product.
These requirements will guide the designers to produce a product as it is envisioned by your client. They seek to determine precisely how the product shall appear to its intended consumer.
- Usability
 - Ease of use: describes your client's aspirations for how easy it will be, for the intended users of the product, to operate it. The product's usability is derived from the abilities of the expected users of the product and the complexity of its functionality. It is necessary to ensure that it has been considered the usability requirements from the perspective of all the different types of users.
 - Personalization and internalization requirements: describes the way in which the product can be altered or configured to take into account the user's personal preferences. The

personalization requirements should cover such things as languages, currencies (including the symbols and decimal conventions), personal configuration options, etc.

- Ease of learning: how easy it should be to learn to use the product. To quantify the amount of time that your client feels is allowable before a user can successfully use the product. This will range from zero time for products intended for placement in the public domain to a considerable time for complex, highly technical products.

This requirement will guide designers in how users will learn the product. For example, the designers may build elaborate interactive help facilities into the product, or the product may be packaged with a tutorial.

- Accessibility requirements: how easy it should be for people with common disabilities to access the product. (i.e. sight, physical disablement, hearing, cognitive, or others).

- Performance Requirements

- Speed and latency requirements: Specifies the amount of time available to complete specified tasks. These often refer to response times. They can also refer to the product's ability to fit into the intended environment.

Some products, usually real-time products, must be able to perform some of their functionality in a given time slot.

- Safety critical requirements: Quantification of perceived risk of possible damage to people, property and environment. To understand and highlight the potential damage that could occur when using the product within the expected operational environment.

If it has been building safety critical systems then the relevant safety critical standards are already well specified.

- Precision requirements: Quantification of the desired accuracy of the results produced by the product. To set the client and user expectations for the precision of the product.

- Reliability and Availability requirements: quantifies the necessary reliability of the product. This is usually expressed as the allowable time between failures, or the total allowable failure rate. It also quantifies the expected availability of the product.

- Robustness requirements: specifies the ability of the product to continue working under abnormal circumstances.

- Capacity requirements: specifies the volumes that the product must be able to deal with and the numbers of data stored by the product.

- Scalability or extensibility requirements: specifies the expected increases in size that the product must be able to handle. As business grow (or are expected to grow), software products must increase their capacities to cope with the new volumes.

- Operational Requirements

- Expected physical environment. To highlight conditions that might need special requirements, preparations or training. These requirements ensure that the product is fit to be used in its intended environment. It should also be taken into consideration that there are users with disabilities other than the commonly-described, such as for low-visibility and poorly lit environments.

- Expected technological environment. To identify all the components of the new system so that the acquisition, installation and testing can be effectively managed. It may be that the operating environment is complex, and becomes a subject of requirements study itself. Special considerations should also be given if the product is to be embedded in a device.
- Partner applications. Requirements for interfacing to other applications often remain undiscovered until implementation time.
- Production requirements. Any requirements needed to make the product distributable or saleable. It is also appropriate to describe here the operations to be performed to have a software product successfully installed.

Some products have special needs to turn them into a saleable, or usable product. You might consider that the product has to be protected such that only paid-up customers can access it. This might be implemented as a dongle, a daily keyword, a check that no other copy of the product is running on the network at the same time.
- Maintainability and Support requirements
 - How easy must be to maintain this product. A quantification of the time necessary to make specified changes to the product.

There may be special requirements for maintainability, such as whether this product must be maintained by its end-users, or developers who are not the original developers. This has an effect on the way that the product is developed, and there may be additional requirements for documentation or training.
 - Special conditions that apply to the maintenance of this product. To make everyone aware of how often it is intended to produce new releases of the product.
 - Supportability. This specifies the level of support that the product requires. This is often done using a help desk. If there are to be people who provide support for the product, this will be a part of the product and there will be requirements for that support. You might also build support into the product itself, in which case this is the place to write those requirements.
 - Portability requirements. Description of other platforms or environments to which the product must be ported.
- Security requirements
 - Access requirements. Specification of who has authorized access to the product, and under what circumstances that access is granted, and to what parts of the product access is allowed.
 - Integrity requirements. Specification of the required integrity of databases and other files, and of the product itself. To specify what the product will do to ensure its integrity in the case of an unwanted happening such as an attack from the outside or an unintentional misuse by an authorized user.
 - Privacy requirements. Specification of what the product has to do to insure the privacy of individuals that it stores information about. The product must also ensure that all laws about privacy of individual's data are observed.
 - Audit requirements. Specification of what the product has to do (usually retain records) to permit the required audit checks.

- Immunity requirements. The requirements for what the product has to do to protect itself from infection by unauthorized or undesirable software programs, such as viruses, worms, Trojan horses and others.
- Cultural and Political Requirements.
 - Requirements that are specific to the sociological and political factors that affect the acceptability of the product. If you are developing a product for foreign markets, then these requirements are particularly relevant.
- Legal Requirements
 - The system falls under the jurisdiction of any law. A statement specifying the legal requirements for this system.
 - Some standards with which we must comply. A statement specifying applicable standards and referencing detailed standards descriptions.

2.1.2.3 INTER-IoT requirements

In the case of INTER-IoT, we have decided to simplify the group number and to specialize for some particular cases. Specifically, we have collected the types of requirements reported in Table 1.

Table 1: Requirement types association

INTER-IoT REQUIREMENTS	VOLERE REQUIREMENTS
FUNCTIONAL	
Functionality	Functional
Semantics	Data
Data model	
NON-FUNCTIONAL	
Look and Feel	Look and Feel
Usability Commercial Application	Usability
Performance Architecture	Performance
Operational Communications Interoperability API Middleware Virtualization	Operational Maintainability
Security	Security

Privacy	
Legality	Legal/standards
QoS	
Methodology	

2.1.3 Requirements Harmonisation process steps

We have used a 5-step iterative process for identifying, capturing, defining, analysing, and reconciling requirements (see Figure 5).

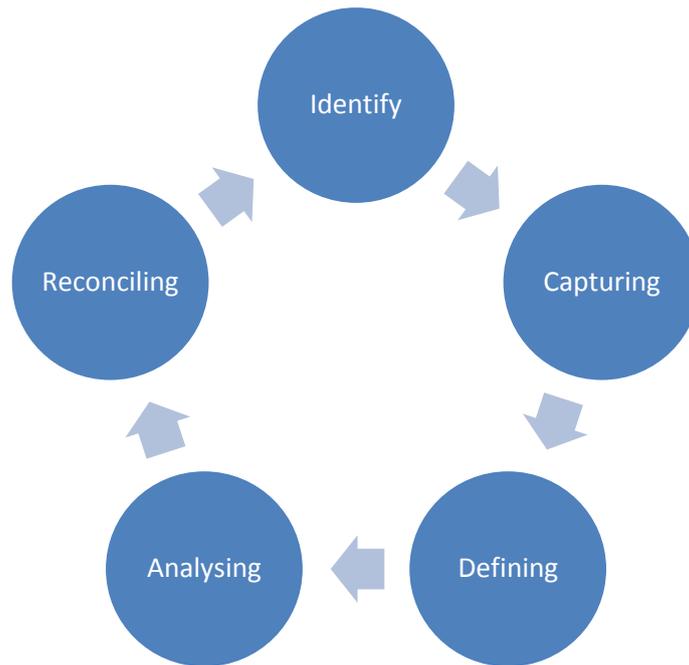


Figure 5: Requirements capture methodology

The requirement harmonisation process is defined as follows:

- Identify sources of requirements
 These were the sources of information to collect requirements such as previous research projects, our own knowledge, stakeholders, regulation, standards, etc.
- Requirement Capturing
 This step generated an inventory of identified requirements by product, including requirement name and brief description.
- Defining
 This step produced a detailed requirement specification following the requirement template and taking into account the characteristics of good requirements specification (see section 2.1.1).
 A requirement template has been prepared with the main information needed in order to be collected from the requirements identified (see Table 2).

Table 2: Requirement template

Requirements			
Requirement's Name: <i>Name of the identified requirement</i>		Identifier: #1	
Category: <i>Functional, Non-functional, or Design constraints</i>	Type: <i>Security, Privacy, QoS, Semantics, Resilience, Interoperability, Data model, Communications, Commercial, Operational ...</i>	Priority: <i>Must, May or Nice to have</i> MoSCoW Priority: <i>Must have, Should have, Could have or Won't have</i>	Status: <i>Approved or Out of scope</i>
Product: <i>INTER-LAYER, INTER-FW, INTER-METH, INTER-LogP, INTER-Health</i>	Affected Layer: <i>In the INTER-Layer product</i>	Scenario: <i>Involved scenario (in case of business pilot)</i>	
Rationale: <i>Reason of involvement</i>			
Requirement Description: <i>Brief description of the requirement</i>			
Acceptance criteria: Conditions that requirement must satisfy to be accepted			
Source: <i>EU project, One-M2M, IOT-A, Partner's expertise, ...</i>	Identified by: <i>Partner who has identified the requirement</i>	Registration Date: <i>Date of registration</i> <i>Date of update</i>	

- Analysing

This phase consisted in assessing the requirements obtained. For the analysis and assessment of requirements we created five task forces as reflected in Table 3 (one per product).

Table 3: Task forces by product

Task forces				
INTER-LAYER	INTER-FW	INTER-METH	INTER-LogP	INTER-Health
UPV	PRO	UNICAL	FVP	TI
TUE	UPV	SRIPAS	PRO	RINI
NEWAYS	XLAB	XLAB	AFT	UNICAL
SRIPAS	ABC	ABC	NOATUM	ASLTO5

These task forces produced the following improvements in the requirements:

- Improving the quality of the description
 - Correcting and homogenizing the relevant classifications
 - Grouping similar requirements
 - Validating the requirements
 - Detecting new requirements not identified in other sources of information
- Reconciling
 - This was the final step in which there were an agreement to incorporate the requirement into the list.

One important point in this process is that after completing the first identification phase, the requirements harmonisation process steps have to be repeated as additional requirements are identified⁸.

2.1.4 Priority of requirements

When implementing the functionality of a system it is important to prioritize the requirements to first develop the essential parts and remove the less significant ones if necessary due to lack of time or resources.

In INTER-IoT, the requirements have two priority types. On the one hand are the initial needs of stakeholders and the final users of the products. On the other hand, it is necessary to prioritize what is essential for the operation of the product for the development. The prioritization technique that has been used as a reference to prioritizing the requirements is MoSCoW.

MoSCoW was developed by Dai Clegg of Oracle UK in 1994 and it gained popularity in the DSDM methodology (Dynamic Software Development Method). The MoSCoW method is a prioritization technique used in management, business analysis, project management, and software development to reach a common understanding with stakeholders on the importance they place on the delivery of each requirement - also known as MoSCoW prioritization or MoSCoW analysis.

MoSCoW is a fairly simple way to sort features into priority order – a way to help teams quickly understand from the customer's view what is essential for launching product and what is not.

The term MoSCoW itself is an acronym derived from the first letter of each of four prioritization categories (Must have, Should have, Could have, and Won't have).

The categories are typically understood as:

- Must have

Requirements labelled as MUST have to be included in the current delivery timebox in order for it to be a success. If even one MUST requirement is not included, the project delivery should be

⁸ Reusing Requirements: Taking Advantage of What you Know <http://www.volere.co.uk/pdf%20files/ReusingRequirements.pdf>

considered a failure. It is good to have clarity on this before a project begins, as this is the minimum scope for the product to be useful.

MUST can also be considered an acronym for the Minimum Usable SubseT.

- Should have

SHOULD have requirements are also critical to the success of the project, but are not necessary for delivery in the current delivery timebox. SHOULD requirements are as important as MUST, although SHOULD requirements are often not as time-critical or there may be another way to satisfy the requirement so that it can be held back until a future phase.

Therefore, it could be considered SHOULD are features that are not critical to launch, but are considered to be important and of a high value to the user.

- Could have

Requirements labeled as COULD are desirable but not necessary, and could improve user experience or customer satisfaction for little development cost. These will typically be included if time and resources permit.

- Won't have

Requirements labeled as WON'T have been agreed by stakeholders as the least-critical, lowest-payback items, or not appropriate at that time. As a result, WON'T requirements are not planned into the development schedule for the delivery timebox.

WON'T requirements are either dropped or reconsidered for inclusion in later phases or projects. This, however, doesn't make them any less important. Alternately described as "Would like to have" in the future.

All requirements are important, but they are prioritized to deliver the greatest and most immediate business benefits early. Developers will initially try to deliver all the Must have, Should have and Could have requirements but the Should and Could requirements will be the first to be removed if the delivery timescale looks threatened.

Thus, this ranking helps everyone (stakeholders, project manager, designer, developers) understand the most important requirements, in what order to develop them, and what not to deliver if there is pressure on resources.

2.2 JIRA Repository

JIRA is a commercial software for issue tracking in software development manufactured by Atlassian. This commercial software can be licensed for running on-premises or as a hosted application. JIRA provides bug tracking, issue tracking, and project management functions. The main features of JIRA for agile software development are to plan development iterations, generate iteration reports and bug tracking functionality.

Because stakeholder's needs, products, scenarios or requirements are elements that can evolve throughout the project, it is necessary to have a tool that allows keeping them updated and accessible among all stakeholders at any time.

The project repository keeps updated and easily accessible the details of stakeholders, market analysis identified products, scenarios, requirements and use cases after the submission of the respective deliverables at the date of delivery.

JIRA implementation

The access URL for the project repository is jira.inter-iot.eu. Each partner of the project has its own credential to access, and there is an extra credential to provide access to external reviewers when required. Figure 6 illustrates Inter-IoT Project on JIRA home page.

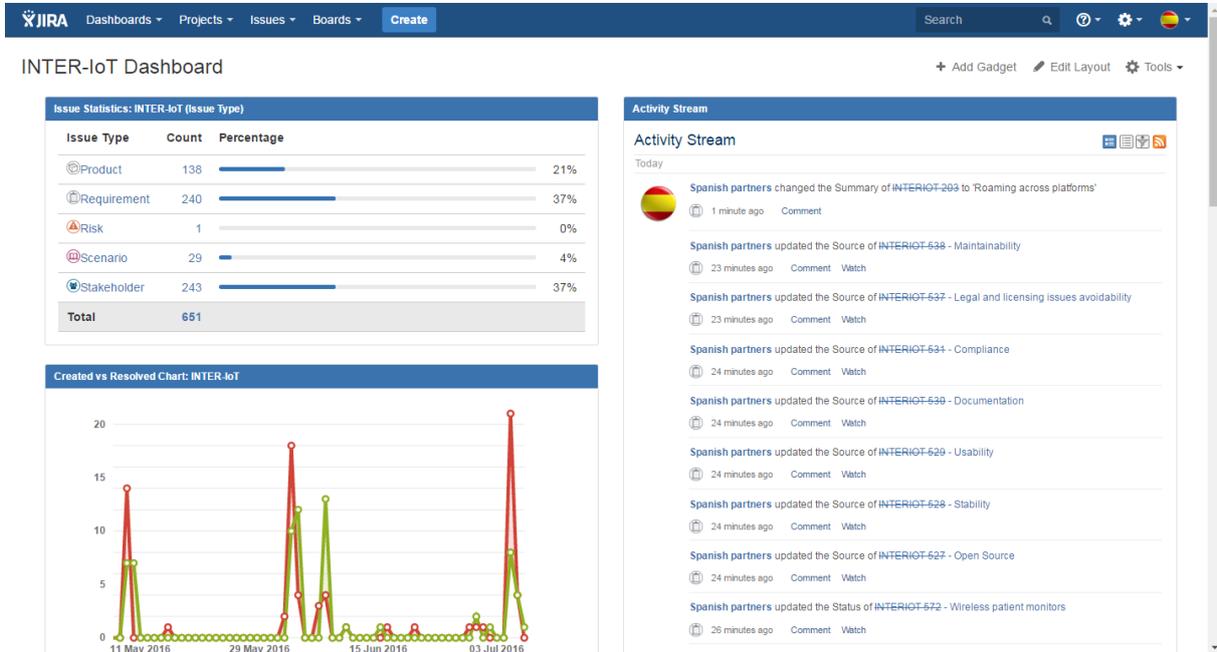


Figure 6: JIRA home page of the INTER-IoT project

Once in the application the user can access all stored information and can filter by type of issue (e.g. stakeholder, product, scenario, requirement, or use case) or by any field or metadata of the form (as shown in Figure 7).

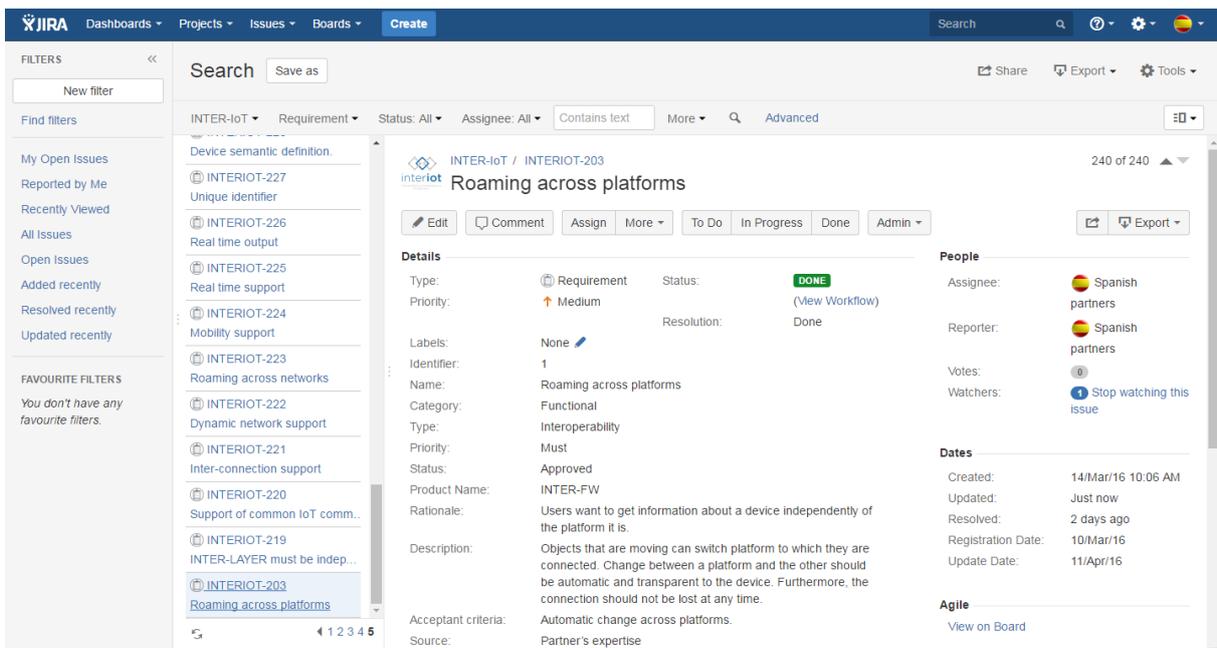


Figure 7: View issues

To create new issues, the user can execute the *create button* option at the top menu as it can be seen in Figure 8. The user can also select the type of issue (Stakeholders, Products, Requirements or Use cases). The templates used for filling the different issues are personalized according to the designed above, following the methodology.

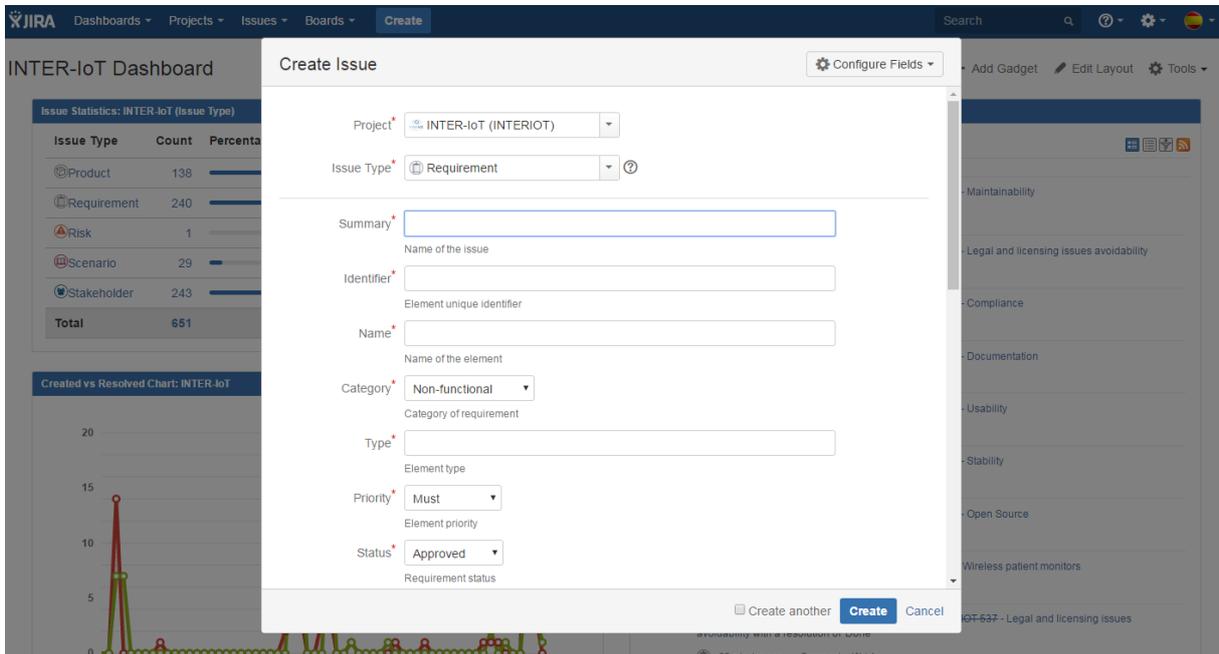


Figure 8: Create new issue

3 Requirement analysis

In this section the requirements identified at this point of the project are listed. They are grouped by the five products defined in INTER-IoT, plus some generic project requirements.

The set of requirements define how should work the different products from the needs of end users, suppliers and developers. A rigorous assessment of requirements before design allows a clear idea of what you want to implement and reduces delays due to design flaws. It also allows estimating more accurately the costs and the risks.

For each section we can find the requirements classified by some of its features. The complete template with all the features of each requirement is in the Annex. There is also an analysis of the requirements in each section with some conclusions that can be drawn.

3.1 INTER-LAYER requirements

For a clear design and implementation of INTER-LAYER, we must identify a collection of functional and non-functional requirements that describe precisely all aspects and functionalities of the product.

3.1.1 Non-functional requirements

Within non-functional requirements we can determine, for our product, the ones related with the description of how INTER-LAYER works, or its operating characteristics. These are requirements that specify the standard, technology, function, usability or appearance that the product will provide. They elaborate the generic performance features of the system in the different covered areas.

Addressability and reachability [11]
INTER-LAYER must addressable and reachable. Processes should be able to communicate among each other, typically through IP addresses or some sort of URL including an ID of the process (or device).
Acceptance criteria: The system must be able to provide end-to-end communication between devices or, when not possible, allow reachability through smart gateways (either at device level or in the cloud) able to mediate among IoT processes.
MoSCoW Priority: Must have

Real time support [20]
INTER-LAYER must support real time data. Real time data transfer and processing must be supported by INTER-IoT to allow real-time sensing.
Acceptance criteria:

Real-time sensors must be supported by the system, that is, the receiving and sent of information from/to the devices in less than a certain set period time (delay) acceptable by the user or the .system. If it is a critical system the real-time must be hard, if it is not so critical, could be soft.

MoSCoW Priority: Must have

Communication restrictions and limitations [29]

INTER-LAYER must comply the communication regulations.

The communication must respect the protocol's restrictions and limitations, for instance if a communication protocol like LoRa is used, the gateway must not force communication over the line.

Acceptance criteria:

Used protocols, transmissions and/or transceivers. Implements all the restrictions described in the official specification of the protocols, and, if applicable, the legislation and law issues of this ones.

MoSCoW Priority: Must have

SDN capabilities [229]

INTER-LAYER must support SDN capabilities.

Inclusion of a SDN module when necessary, which could be used or not, to implement network functionalities in a more centrally located and easier manner. Thus network administration is managed through abstraction of higher-level functionalities.

Acceptance criteria:

Centralized management of network functionalities expediting the control and administration of them.

MoSCoW Priority: Must have

Support of semantic modelling in the middleware layer [235]

Middleware must support semantic modelling.

Semantic interoperation defines the rules for understanding the meaning of the content of information, and creates a domain specific information model, known as semantic model.

Acceptance criteria:

Allow semantic interoperation in the middleware layer.

MoSCoW Priority: Must have

Support of main Internet of Things platforms [236]

INTER-LAYER must support existing IoT platforms.

INTER-LAYER requires connectors to different IoT platforms (Fiware, OpenIoT, OM2M, Sofia2...) to access their services like discovery, access, tasking, location, etc.

Acceptance criteria:

The platform should ensure the connection to Fiware, OM2M and WSO2.

MoSCoW Priority: Must have

Object/Device virtualization [242]

INTER-LAYER must virtualize objects.

The IoT platforms use virtual objects of its physical entities for managing data from different sources.

In order to facilitate sensor data to the other layers, the gateway should store a virtual image for each object/device that has to reflect the real time value of each sensor in that object/device. This way, multiple petitions can be coordinated and there is no overload in the access network of that sensor. The virtual representation should also handle the interaction with the actuators of each sensor.

Acceptance criteria:

Each object/device should have a one to one representation of each sensor that should be exposed. That virtual representation must be as real time as possible. Actuators can be controlled from that virtual representation.

The Haulier IoT platform generates a virtual entity for each truck. The management data associated to that truck is attached to this virtual object.

MoSCoW Priority: Must have

Scalability. Design [2]

INTER-LAYER should be scalable.

Scalability is related to the ability of systems to seamlessly cater for higher demand in computing resources of data, devices, people and applications. The system should be designed to be scalable.

Acceptance criteria:

Scalability is typically obtained via a scale-out approach and using cloud services, this primarily targets the higher layers of INTER-LAYER. INTER-IoT should, at least, introduce currently available scalability approaches in cloud services (e.g. Amazon, Azure).

MoSCoW Priority: Should have

Support of opportunistic communications to avoid data loss [7]

INTER-LAYER should support opportunistic communications.

The system shall be able to support opportunistic communications in order to guarantee data availability. When Internet connectivity is not available (i.e. due to mobility, interference, etc.) opportunistic communication technologies should be able to avoid loss of data.

Acceptance criteria:

Multimode devices, when applicable, are used to build ad-hoc networks and thus preserve connectivity or some temporal storage service (when applicable) are used to retain data that is later sent automatically into the system when network communication is available again.

MoSCoW Priority: Should have

Extensibility [13]

INTER-LAYER should be extensible.

Extensibility of all the system components has to be taken into account, from providing hardware platforms to integrate multiple sensors to middleware software, to testbed infrastructures. The M2M middleware should be capable of receiving the data from multiple types of sensors: physical or emulated.

Acceptance criteria:

INTER-Layer (and INTER-FW) should be able to easily support extensions, updates and inclusion of new modules as they are being integrated. Furthermore, as SDOs and supported protocols evolve, this fact should be reflected in an easy way to extend INTER-LAYER and INTER-FW.

MoSCoW Priority: Should have

Common IoT communication protocols must be supported. [15]

INTER-LAYER should support IoT communication protocols.

The IoT gateway must be able to use the most common communication protocols for IoT.

Acceptance criteria:

Any communication protocol can be used. The most used communication protocols on IoT, at different layers as: BLE, IEEE 802.15.5, IEEE 802.11 in physical, IPv6 and OpenFlow in network, or MQTT and CoAP in middleware must be supported by the interoperability solution. INTER-Layer must implement this protocols and successfully exchange the information between the systemsystems involved.

MoSCoW Priority: Should have

Inter-connection support [16]

INTER-LAYER should support communications with other systems.

Sensors, data, networks and platforms from other sources must be able to be coupled into middleware applications for extended analysis or use of advanced algorithms. These sensors and devices may be running via different middleware.

Acceptance criteria:

Devices, networks and IoT platforms have to be interconnected through Inter-IoT solutions. New platforms should be automatically added and accepted.

MoSCoW Priority: Should have

Dynamic network support [17]
<p>INTER-LAYER should support dynamic network.</p> <p>A continuous change of networks and devices must be allowed. Especially sensor networks may be very dynamic in nature. New sensors may come online, while others die out. Sensors may be offline or in hibernation for long time periods. The INTER-IoT must be able to handle these constant changes in the network.</p> <p>Wireless networks like a Zigbee network with multiple simple devices must be able to be connected through the gateway/switch/hub which contains the intelligence.</p>
<p>Acceptance criteria:</p> <p>Rapid and dynamical network changes must be possible. Scaling up and down of devices and LAN networks is possible, complete networks must able to be connected to the IoT system.</p>
<p>MoSCoW Priority: Should have</p>

Real time output [21]
<p>INTER-LAYER should have real time output.</p> <p>In order for the system to operate in real time, a short delay is allowed. Ideally the delay is not noticeable.</p>
<p>Acceptance criteria:</p> <p>Real time delays should not interfere with the proper performance of the system (Less than 5 seconds).</p>
<p>MoSCoW Priority: Should have</p>

IoT Services discoverability [43]
<p>INTER-LAYER should have discoverability of IoT services.</p> <p>The IoT services are generally available without human intervention. However, this does not mean that humans (user of the IoT services) do not need to know about the existence of the IoT services surrounding the users. When the IoT services are provided to a user, it is recommended that the user can notice the presence of the IoT services and do so in a manner consistent with the relevant regulations. A collection of available services must be made available to the user to browse and search.</p>
<p>Acceptance criteria:</p> <p>The different services, made available for the pilots, shall be discoverable through the platform, and the pilots should make use of the discovery features to connect the services with the data.</p>
<p>MoSCoW Priority: Should have</p>

Precise synchronization [56]
<p>INTER-LAYER should support precise synchronization.</p> <p>A system built using INTER-IoT shall support synchronization.</p>

Acceptance criteria:

Services which depend on a precise timing require that the devices they are communicating have the same time, and allow precise synchronization among them.

MoSCoW Priority: Should have

Priority of routing and processing of critical messages upon low-priority sensor data [89]

INTER-LAYER should prioritize routing and processing critical messages.

The devices or smart objects are semi-autonomous, they have to send a lot of information to the Big Data Servers, but they only receive a few orders. It is very important give priority and security to the orders. Also, for instance, in INTER-Health any information that could trigger deterioration of health status should be transmitted with emergency priority. The alarm capability should be at multiple levels, as multiple emergency situations can occur. The alarm should be triggered by the monitoring device upon a list of predetermined emergency situations. The alarm needs to be reset manually so as to guarantee a human control over the situation. The system should allow priority to the processing and transmission of high-priority data such as data for sensitive issues, alarms, orders, etc.

Acceptance criteria:

Define priority routing situations and enable priority routing for information related to those situations. At least two levels of priority should be provided. The highest level corresponds to alarms, orders, etc. and requires security and message confirmation. The rest could be without confirmation (therefore UDP could be used).

MoSCoW Priority: Should have

Sensor service mapping among most prominent standards [221]

INTER-LAYER should have semantic interoperability for services.

The system supports semantic-level semantic interoperability for services.

Acceptance criteria:

This requirement is achieved through an API for offering sensor service interoperability among W3C SSN, ETSI SAREF and One M2M.

MoSCoW Priority: Should have

Offloading [227]

INTER-LAYER should support offloading.

Offloading is the use of complementary network technologies for delivering data. Reduces the amount of data being carried on one connection and freeing bandwidth for other uses. It is also used in situations where local reception may be poor.

The INTER-LAYER product should provide to the device the possibility of connection to more than one access network from the same or different gateways in order to discharge faster or safer the data it gathers, improving the performance, throughput and flexibility of the connection between those parts.

Acceptance criteria:

The systems allows offloading, i.e. data-load distribution over complementary networks.

MoSCoW Priority: Should have

Support Service Composition [239]

INTER-LAYER should have choreography and orchestration of services.

Support Service composition through one of its forms: Choreography and Orchestration.

Service Orchestration is a form of Service Composition in which the coordination of the involved services is performed by a central component. This component is called the orchestrator and defines the messaging schema, needed to fulfil the composite service.

Service Choreography is a form of Service Composition in which the participating services interact without being coordinated by a central component. The messaging schema between the elementary services is defined by a global point of view outside the involved services.

Acceptance criteria:

In service orchestration, a new service is created by combining several existing services in a workflow. Thus capabilities for orchestration are a must to build flexible solutions at the top of INTER-IoT. On the other side, Service choreography defines the interaction protocol between services, also can be used to build distributed solutions at the top of INTER-IoT.

MoSCoW Priority: Should have

Gateway virtualization [244]

INTER-LAYER should virtualize part of the gateway.

A real D2D gateway can be split in two parts, one that allows different access network for the object/devices; and the proper gateway functions and services that can be fully virtualized.

A physical gateway must be able to be split in two parts the physical and the virtual one.

Acceptance criteria:

The connection by the device through the access network located on the physical part and the access by the platform to the gateway services by the virtual one.

MoSCoW Priority: Should have

Efficiency of the information processing [6]

INTER-LAYER could process the information efficiently.

The IoT system and devices should optimize the processing of information with respect to a cost function, e.g. related to communication, computation, energy, etc.

Acceptance criteria:

Depending on the involved layer and use case, a different cost function is applied and must be defined and evaluated.

MoSCoW Priority: Could have

Multi-level data processing support [9]

INTER-LAYER could have multi-level processing.

Depending on the application and the security/privacy requirements, data may need to be aggregated/processed/fused at different system levels, i.e. at device level, at the gateways, at intermediate nodes (cluster heads or neighbouring nodes), etc.

For example, Compressive Sensing aggregates/fuses/encrypts data at intermediate nodes, but this may not be efficient for some types of private data. Another example is related with the requirement to transmit aggregated data of user speeds in the traffic monitoring use cases for privacy preservation. Similarly, fusion/encryption at device level may waste resources/energy.

Acceptance criteria:

Management distributed tools include all components defined in WP3 that are able to provide IoT platform layer interoperability/integration: wrappers, mediators, brokers, translators, and matchers. The distributed approaches impose, as a non-functional requirement for the system architecture, the ability to support multi-level data processing.

MoSCoW Priority: Could have

Monitoring and self-awareness of the system [57]

INTER-LAYER could have self-awareness of the system.

The system must collect evidence from its components to check that they are actually running.

Acceptance criteria:

An IoT system should have self-awareness of its elements. Systems require information of their elements status and performance. INTER-LAYER needs to ensure reliability between its components.

MoSCoW Priority: Could have

Communication with message size efficient protocols [72]

INTER-LAYER could have efficient protocols for communications.

Communication should be done using protocols that are efficient in terms of amount of exchanged information over amount of exchanged data measured in bytes.

Acceptance criteria:

Selection of communication protocols that are commonly acceptable, efficient data transmission.

MoSCoW Priority: Could have

The interaction between IoT endpoints may follow M2M concept [75]

INTER-LAYER could use the M2M concept for interaction between IoT endpoints.

<p>The interaction between IoT endpoints may follow the M2M communication concept. M2M refers to solutions that enable machines to communicate with back-end information systems and/or directly with other machines, in order to provide real-time data. M2M communication can be event-based and/or polling based (predefined time intervals). M2M applications consider the following stages: data collection, data transmission, data validation, response to available information.</p>
<p>Acceptance criteria:</p> <p>The communication between IoT platforms on devices, networking and middleware layers follows M2M approach with minimal or none human interaction.</p>
<p>MoSCoW Priority: Could have</p>

<p>Automatic and dynamic selection of communication protocol [78]</p>
<p>INTER-LAYER could select automatically and dynamically the communication protocol.</p> <p>When an object is able to send information via various communication systems, there should be a mechanism for selecting the most appropriate depending on the data to transmit, distance, coverage, location, etc.</p>
<p>Acceptance criteria:</p> <p>Protocol to automatic selection of communication in devices with more than one.</p>
<p>MoSCoW Priority: Could have</p>

<p>Support multicast communication among devices [80]</p>
<p>INTER-LAYER could support multicast communication.</p> <p>To transmit information to many devices is more efficient to use multicast communication. Sometimes using a multicast protocol for faster communication and optimization of resources may be necessary.</p>
<p>Acceptance criteria:</p> <p>Implementation of a protocol (e.g. CoAP support multicast) that allows multicast communication between devices and gateway, in the specific situation that requires it. (e.g. group of devices with similar characteristics that communicates via CoAP).</p>
<p>MoSCoW Priority: Could have</p>

<p>Minimum data fields on the protocols for the device communications [93]</p>
<p>INTER-LAYER could have a standard protocol for the device communications.</p> <p>Communication protocols must provide the minimum data fields for manage the devices and the information they provide.</p> <p>It is needed the implementation of protocols which have at least ID, security, authentication, data, etc. fields to communicate with the devices and to obtain the information they provide in a understandable format.</p>
<p>Acceptance criteria:</p>

Could be a repository where it is defined the data fields and number of fields, to store the information (meta-data) about the device, and the information (meta-data) about the data they provide.

MoSCoW Priority: Could have

Robustness, resilience and availability [95]

INTER-LAYER could be sturdy, resilience and available.

The system should guarantee availability, robustness and resilience. To ensure those capabilities, the system has to be designed to avoid flaws and attacks that cause a system failure.

However if a failure takes place, the system should be resilient to recover itself in a proper amount of time in order to ensure and maximize availability.

INTER-IoT has to implement tools and mechanisms to detect fails and alerts about reliability (i.e. It should store the information lost in buffers in order to be sent later).

Acceptance criteria:

The system guarantees availability and a fast recovery, by the ability of the different services and elements to prevent and react on attacks and failures (robustness & resilience).

MoSCoW Priority: Could have

Support smart network resource allocation in heterogeneous wireless sensor networks [204]

INTER-LAYER could support smart resource allocation in wireless networks.

The smart resource allocation (e.g. transmission channel, transmission power) can be done using both ways, deterministic (using standard deterministic algorithms), and probabilistically (using self-learning capable algorithms, e.g. reinforcement learning).

Acceptance criteria:

The system must be compatible with deterministic and probabilistic algorithms.

MoSCoW Priority: Could have

Power-awareness for communications [206]

INTER-LAYER could maximize energy in communications.

The communication protocols have to minimize the power consumption during exploitation to maximize the battery lifetime, while still being reliable and robust.

Acceptance criteria:

Use energy efficient communication protocols and avoid redundant communications.

MoSCoW Priority: Could have

Support scalable network topologies [207]

<p>INTER-LAYER could support scalable network topologies.</p> <p>To achieve an optimal topology to maximize their quality of service, the networks need to be capable of auto-reconfiguration, to auto-detect redundancy, and to find themselves their strongest and weakest elements.</p>
<p>Acceptance criteria:</p> <p>Using network science algorithms or topology-aware paradigm (as SDN), the network will keep growing and still be reliable and with a min QoS. The inclusion of new nodes in the network must not affect the performance of the existing one.</p>
<p>MoSCoW Priority: Could have</p>

<p>Support Mash-up [240]</p>
<p>INTER-LAYER could support Mash-up of services.</p> <p>Support Mash-up to create new services from existing services.</p>
<p>Acceptance criteria:</p> <p>The utilization of Mash-up tools to provide a simple way to develop applications by composing, or mashing-up, existing services in the Web.</p>
<p>MoSCoW Priority: Could have</p>

<p>Legacy gateway integration [245]</p>
<p>INTER-LAYER could support integration of legacy gateways.</p> <p>There are several gateways already in the market, they have to be considered as well in the INTER-IoT solution.</p>
<p>Acceptance criteria:</p> <p>Legacy devices could also connect to the gateway. This one must support the protocols that the legacy devices understand.</p>
<p>MoSCoW Priority: Could have</p>

3.1.2 Functional requirements

Additionally, we present the requirements that specify what INTER-LAYER should do, by means of behaviour or concrete function, as a specific facility of the system.

<p>Device semantic definition [23]</p>
<p>INTER-LAYER must have a semantic definition for devices.</p> <p>Each device can be defined (attributes, metadata, etc.) by means of a semantic ontology.</p>
<p>Acceptance criteria:</p> <p>A device is semantically described. The minimum unequivocal data fields of information are needed for semantically recognize, identify and connect with a specific device. Additional fields meta-data, no mandatory, can join the must data, in order to provide context or extra-information.</p>

MoSCoW Priority: Must have

Connectivity not based on HW identifiers [45]

INTER-LAYER must not base the communications in HW identifiers.

No single link meets all communication needs for the Industrial Internet of Things. Today multiple RF links, multiple power line links and a variety of wired solutions are needed to implement the various applications. Furthermore, transceiver development is an area of active research and investment, so the protocol stack must be able to take advantage of new technologies as they become available.

Acceptance criteria:

Pilots test the correct functioning of device identification at several levels and in a heterogeneous environment. Different Pilots make usage of different solutions, proving the flexibility of integrating heterogeneous technologies.

MoSCoW Priority: Must have

IoT Platform Semantic Mediator provides data and semantic interoperability functionality [178]

IPSM must have data semantic interoperability.

IoT Platform Semantic Mediator provides data and semantic interoperability functionality accessible with a set of interfaces. Source and target platform send messages to and consume messages from dedicated endpoints (messages in platform's semantics and format).

Precondition: IoT platforms data exchange formats and semantics and services description are explicitly documented.

Acceptance criteria:

IoT platforms send requests and receive responses with their data formats and semantics, regardless of data format and semantics of target platforms. Communication is supported with interfaces provided by semantic mediator tool.

MoSCoW Priority: Must have

IoT Platform Semantic Mediator supports platform to platform communication and communication between platforms and an external actor [179]

IPSM must have data interoperability between platforms.

Inter Platform Semantic Mediator supports platform to platform communication and communication between platforms and an external actor.

Communication between platforms may affect different layers e.g. MW2MW, AS2AS, but can also be initiated by an external actor e.g. request from external system to provide data that are gathered and stored in different IoT platforms.

Inter Platform Semantic Mediator shall provide functionality to achieve common understanding of the communication (translation of semantics between communication parties).

Acceptance criteria:

Two IoT platforms are able to communicate, and an external actor can access data from different IoT platforms in a uniform way (indicated and documented semantics and format).

MoSCoW Priority: Must have

Semantic and syntactic interoperability [180]

INTER-LAYER must provide semantic and syntactic interoperability.

The mixing and mashing of data gathered by many IoT applications adds values to the collected data as a whole and to facilitate such data exchanges, the IoT applications require common data formats and application programming interfaces (APIs) so data can be accessed and combined as needed. For achieving semantic interoperability syntactic interoperability must be enabled. It can be achieved through simple translation. Dedicated INTER-IoT components (producers, IoT Platform Semantic Mediator, consumers) can receive requests in selected data formats and semantics and translate them to the data formats and semantics of target IoT platform. Supported data interchange formats are at least: OWL, RDF, XML, and JSON.

Acceptance criteria:

Mechanisms are proposed to translate data format and semantics of exchanged message to achieve communication with common understanding on both sides. Syntactic interoperability among different protocols, W3C SSN, ETSI SAREF and One M2M as a minimum. FI-WARE and OGC SensorThing as a may.

MoSCoW Priority: Must have

IoT Platform Semantic Mediator does not store sensor data [183]

IPSM must not store sensor data.

All data is stored in IoT platforms. IoT Platform Semantic Mediator supports homogeneous access to the data without storing it internally.

Acceptance criteria:

In case of communication between IoT platforms data is stored in platforms that own them and exchanged on request. IoT Platform Semantic Mediator does not store sensor data but supports mechanisms to exchange them with common understanding.

MoSCoW Priority: Must have

Monitoring and provision of subscription services between different platforms [201]

INTER-LAYER must provide subscription between platforms.

The system needs a publisher/ subscriber model to some of the offered services. For instance, notification or alert systems must have a push operation in order to act in real time.

Acceptance criteria:

Subscribe to an alarm among heterogeneous IoT platforms upon entry in (geofence) an area and, more generally, whenever the attributes of a virtual entity change according predefined

values or ranges (e.g. movement, temperature variations, infringement, delivery completion, etc.).
MoSCoW Priority: Must have

API for network services [226]
INTER-LAYER must provide an API for network services. The network layer should provide an API to guarantee that the access to the gateways and devices could be accomplished in a transparent manner from the upper layers point of view.
Acceptance criteria: INTER-LAYER must provide the specific available API for the access.
MoSCoW Priority: Must have

Network function virtualization[231]
INTER-LAYER must support network virtualization. All the components of the network (routers, switches, load balancers, firewalls, controllers, etc.) could be virtualised within a central component, for mainly managing and orchestration.
Acceptance criteria: The different functions performed by the network, normally in a physical distributed way (e.g. firewall, routing, DPI) must be virtualized and centralized in an adequate virtual controller.
MoSCoW Priority: Must have

API Middleware for interoperability between different platforms [237]
Middleware must provide an API for interoperability between platforms. An exported API by the middleware provides access to the different devices of the platforms connected to Inter-Layer.
Acceptance criteria: The correct access and utilization of the functionalities provided by the middleware layer, exposed through and public API.
MoSCoW Priority: Must have

Native support services [241]
INTER-LAYER must provide access to the native services of the IoT platforms.
Acceptance criteria: INTER-IoT will implement or use a mechanism to provide direct access for using the native services located within the IoT platforms.
MoSCoW Priority: Must have

Gateway access API [243]
<p>INTER-LAYER must provide an API for accessing the gateway.</p> <p>The API is obligatory in order to retrieve virtual object/device data, control of actuators, etc. from another layer.</p>
<p>Acceptance criteria:</p> <p>All exposed functions of the gateway are reachable from the API.</p>
<p>MoSCoW Priority: Must have</p>

Roaming across networks [18]
<p>INTER-LAYER should support roaming across networks.</p> <p>The devices and IoT must be robust for signal loss and be able to reconnect when coverage has been restored. The system must be able to recognize the device again, data dumps must be supported.</p> <p>An object changes location making a change on the platform that is connected. The change is automatic, unattended and transparent to the user.</p> <p>Some communication standards are based on 1-way communication, so both communication principles must be supported.</p>
<p>Acceptance criteria:</p> <p>The device can travel from one access network to another without losing connections. The device must be recognized again and allowed to do a logging data dump. Automatic change across platforms and any communication standard can be used.</p>
<p>MoSCoW Priority: Should have</p>

Gateway capabilities[39]
<p>Gateway should support multiple technologies interoperability.</p> <p>Multiple interfaces support: At the device layer, the gateway capabilities support devices connected through different kinds of wired or wireless technologies, such as a controller area network (CAN) bus, ZigBee, Bluetooth or Wi-Fi. At the network layer, the gateway capabilities may communicate through various technologies, such as the public switched telephone network (PSTN), second generation or third generation (2G or 3G) networks, long-term evolution networks (LTE), Ethernet or digital subscriber lines (DSL). Protocol conversion: There are two situations where gateway capabilities are needed. One situation is when communications at the device layer use different device layer protocols, e.g., ZigBee technology protocols and Bluetooth technology protocols, the other one is when communications involving both the device layer and network layer use different protocols e.g., a ZigBee technology protocol at the device layer and a 3G technology protocol at the network layer.</p>
<p>Acceptance criteria:</p> <p>The network layer must demonstrate the interconnecting capabilities between different technologies with the pilots.</p>
<p>MoSCoW Priority: Should have</p>

User device capability detection [138]
<p>INTER-LAYER should assess the devices capabilities.</p> <p>Service shall be able to check user device capability according to required device features for application. This information should be checked at the earliest stage, upon first connection attempt, so as to prevent user of becoming aware of incompatibility after having started to use the system. If the device is not fully compliant, the system sends to the device a standard message pointing to the non-compliance. If the device is totally non-compliant (i.e. making any communication between the system and device impossible), perhaps an acoustic alarm can be triggered upon connection attempt. During the connection attempt the IoT system sends its stack version number. So the system can determine if connection and compliance is possible. Also the device must send its capabilities so the system knows what kind of device is connected and what the device can do. This way also legacy devices may still be supported.</p> <p>If a device does not react at all, or does not sent meaningful messages, it can be concluded that the device is not compliant, otherwise, the level of compliancy can be determined by means of device capabilities and stack version number.</p>
<p>Acceptance criteria:</p> <p>Develop criteria for device capability and a system that checks compliance as soon as sign-in is in process (after having typed ID and password and before granting access). Error messages must be unambiguous and simple. A list of device capability criteria and a list of error messages should be available to users.</p>
<p>MoSCoW Priority: Should have</p>

Cacheable Data [153]
<p>INTER-LAYER should store incidents data.</p> <p>Devices and systems may be able to cache data so information is stored for each incident or appointment that is available in a central server. Inter-IoT should be able to handle real-time data or bulk transfer in the case of cached data from a temporarily disconnected system.</p>
<p>Acceptance criteria:</p> <p>All incident information captured on the IoT platform, if available, is stored on the server.</p>
<p>MoSCoW Priority: Should have</p>

Support the connectivity of a physical entity to several access networks [170]
<p>INTER-LAYER should support connectivity between physical entity and access networks, All objects must be connected through any communications technology (WiFi, GSM, Satellite, etc.), or even more than one. So that it is accessed regularly to their information and you can send orders in response to the data processing.</p>
<p>Acceptance criteria:</p> <p>A physical entity has more than one technology of access network to connect with the gateway (e.g. satellite and 3G), so the connectivity must be possible in all cases.</p>
<p>MoSCoW Priority: Should have</p>

Ontology mapping among most prominent standards [220]
<p>INTER-LAYER should provide ontology mapping among standards.</p> <p>The data & semantics layer allows ontology mapping between SSN, SAREF and OneM2M, at least.</p>
<p>Acceptance criteria:</p> <p>This requirement is achieved through semantic data interoperability among W3C SSN, ETSI SAREF and oneM2M.</p>
<p>MoSCoW Priority: Should have</p>

Special considerations in the semantic ontology to objects with low resources [225]
<p>INTER-LAYER should consider semantic ontology in low resources objects.</p> <p>Many sensors and smart devices have low resources (low battery, connectivity, etc.). Semantic ontologies were not designed taking into consideration the special characteristics and resource limitations of this type of devices, and there are some difficulties in the use of semantic annotations with them.</p>
<p>Acceptance criteria:</p> <p>GOIoTP is designed taking into consideration the difficulties in the semantic representation and updating of data from smart objects with low resources, trying to solve this drawback. By this means, GOIoTP becomes an ontology that takes into consideration a relevant problem, and therefore, optimal in this aspect for IoT.</p>
<p>MoSCoW Priority: Should have</p>

Fault tolerance [232]
<p>INTER-LAYER should have tolerance towards faults.</p> <p>Network must be designed to support procedures to maintain connection persistence in the event of a failure and the ability to recover from such failures without disruption of customer traffic. Having reporting of faults, redundancy and recovery capabilities.</p>
<p>Acceptance criteria:</p> <p>The network layer must have mechanisms to ensure reliable communication among network units.</p>
<p>MoSCoW Priority: Should have</p>

Provide connectors to middleware standards [234]
<p>INTER-LAYER should provide connectors to middleware standards.</p> <p>INTER-IoT should be able to provide connectors to middleware standards (One-M2M, SSN W3C and ETSI). Various platforms use different standards, however ETSI M2M, One-M2M and SSN W3C are starting to emerge as interesting standards to consider, therefore the INTER-LAYER system should support these standards.</p>

Acceptance criteria:

Provide a connector to One-M2M, SSN W3C and ETSI.

MoSCoW Priority: Should have

A common data model compatible with all platform-specific models is shared [255]

INTER-LAYER should have a shared model compatible with commercial platforms.
The system has a common data model for internal operations.

Acceptance criteria:

There is a common data model for INTER-IoT internal operations.

MoSCoW Priority: Should have

Manage a sensor or actuator [283]

INTER-LAYER should be able to manage actuators.

In addition to receiving information from sensors and actuators, it is necessary to be able to send configuration changes or specific actions.

Acceptance criteria:

Development of a method to send an action to a specific actuator.

MoSCoW Priority: Should have

Remote device control [26]

INTER-LAYER could be able to control devices remotely.

Manufacturer apps must be supported and allowed to take control over their devices.

Acceptance criteria:

A manufacturer app remains functional once the system is connected to IoT.
Compatible devices must be able to be connected and controlled together.

MoSCoW Priority: Could have

Location of sensor and measurement is included in semantic models [53]

INTER-LAYER semantic models could include sensor's location and measurement.
The location/position has to be an element to be considered in order to identify the devices.
This location needs to include:

- Sensor location
- Measurement location (where is the feature that is being measured)

The location must be expressed under a standardized way (OGC related).

Acceptance criteria:

Location is available as a feature for data and services, if it is available on the device/sensor.

MoSCoW Priority: Could have

Provide services to detect and predict devices' events in heterogeneous wireless networks [205]

INTER-LAYER could predict events in wireless networks.

The event and prediction detection system may work decentralized (it runs on the devices themselves), centralized (it runs in the cloud), collaborative (more devices are working together to detect events) or in a hybrid manner (combination of the above). Furthermore, the system may use simple detection techniques (e.g. thresholding) or more advanced algorithms (e.g. machine learning).

Acceptance criteria:

The IoT system must allow event and prediction detection systems to run in several ways (decentralized/centralized/collaborative/hybrid).

MoSCoW Priority: Could have

MPTCP support [228]

INTER-LAYER could support MPTCP.

The network layer supports MPTCP extension for TCP to enable a simultaneous use of several IP-addresses/interfaces, establishing connection using multiple paths to maximize resource usage and increase connection resilience, performance, efficiency, robustness and redundancy.

Acceptance criteria:

INTER-LAYER could support the use of MPTCP connections.

MoSCoW Priority: Could have

6LoWPAN protocol support [230]

INTER-LAYER could support 6LoWPAN protocol.

Communication should include the capability of use protocols as 6LoWPAN or other communication protocols to identify and connect wireless devices. These protocols are made for devices with limited power, memory and processing resources as IoT sensors.

Acceptance criteria:

The data flows can travel among the network being addressed by the 6LoWPAN protocol.

MoSCoW Priority: Could have

Flow control and network information tracking [233]

INTER-LAYER could provide flow control and network information tracking.

The network must provide mechanism for notifications, or administrations reports, monitoring information from the data plane, such as threshold exceed alerts or details of

performance. Also a mechanism for flow control to avoid message loss, high queuing delays or general degradation of the network performance.
Acceptance criteria: The network layer should provide flow control techniques and periodic reports to ensure the performance of the network.
MoSCoW Priority: Could have

Publish data stream into a platform through the middleware [281]
Middleware could provide a mechanism to publish data stream in a platform. A user may be able to publish data stream into a platform for a specific (usually previously registered) sensor, so that the platform is aware of the continuous data updates. This means that, as it's done for the subscription, the API allows the user to establish a continuous channel to publish sensor data into the connected platform.
Acceptance criteria: The implementation of tools to configure and perform the publishing of data about a sensor into a connected IoT platform.
MoSCoW Priority: Could have

Map publish/subscription between platforms in the middleware [282]
Middleware could provide publication and subscription between platforms. The user can configure to map publish/subscription between platforms, by performing a specific subscription from a source IoT platform and mapping it to be directly published in the target IoT platform.
Acceptance criteria: An API allows to set up a subscription from a source IoT platform to the target IoT platform.
MoSCoW Priority: Could have

3.1.3 Requirements by type

The entire assortment of requirements can be presented divided by their type, as each one of them is focused on describing characteristics involved in the different fields covered by the product. Additionally, we could have a more general overview of which types are well addressed and which have a lack of specifications.

Application

- Support Service Composition [239]
- Support Mash-up [240]
- Native support services [241]

Architecture

- Scalability. Design [2]

- Efficiency of the information processing [6]
- Multi-level data processing support [9]

Communications

- Support of opportunistic communications to avoid data loss [7]
- Common IoT communication protocols must be supported. [15]
- Dynamic network support [17]
- Roaming across networks [18]
- Gateway capabilities [39]
- Connectivity not based on HW identifiers [45]
- Communication with message size efficient protocols [72]
- Automatic and dynamic selection of communication protocol [78]
- Support multicast communication among devices [80]
- Cacheable Data [153]
- Support the connectivity of a physical entity to several access networks [170]
- Power-awareness for communications [206]
- Offloading [227]
- MPTCP support [228]
- SDN capabilities [229]
- 6LoWPAN protocol support [230]
- Network function virtualization [231]
- Flow control and network information tracking [233]

Data model

- Location of sensor and measurement is included in semantic models [53]

Functionality

- Addressability and reachability [11]
- Real time support [20]
- Real time output [21]
- Device semantic definition [23]
- Remote device control [26]
- IoT Services discoverability [43]
- Priority of routing and processing of critical messages upon low-priority sensor data [89]
- IoT Platform Semantic Mediator supports platform to platform communication and communication between platforms and an external actor [179]
- IoT Platform Semantic Mediator does not store sensor data [183]
- Publish data stream into a platform through the middleware [281]
- Map publish/subscription between platforms [282]

Interface

- Gateway access API [243]

Interoperability

- Extensibility [13]
- Inter-connection support [16]
- Precise synchronization [56]
- Minimum data fields on the protocols for the device communications [93]
- User device capability detection [138]
- Sensor service mapping among most prominent standards [221]
- API for network services [226]
- Legacy gateway integration [245]
- A common data model compatible with all platform-specific models is shared [255]
- Manage a sensor or actuator [283]

Legality

- Communication restrictions and limitations [29]

Middleware

- Provide connectors to middleware standards [234]
- Support of main Internet of Things platforms [236]
- API Middleware for interoperability between different platforms [237]

Operational

- Monitoring and self-awareness of the system [57]
- The interaction between IoT endpoints may follow M2M concept [75]
- IoT Platform Semantic Mediator provides data and semantic interoperability functionality [178]
- Monitoring and provision of subscription services between different platforms [201]

QoS

- Support smart network resource allocation in heterogeneous wireless sensor networks [204]
- Provide services to detect and predict devices' events in heterogeneous wireless networks [205]
- Shall support scalable network topologies [207]

Security

- Robustness, resilience and availability [95]
- Fault tolerance [232]

Semantics

- Semantic and syntactic interoperability [180]
- Ontology mapping among most prominent standards [220]
- Special considerations in the semantic ontology to objects with low resources [225]
- Support of semantic modelling in the middleware layer [235]

Virtualization

- Object/Device virtualization [242]
- Gateway virtualization [244]

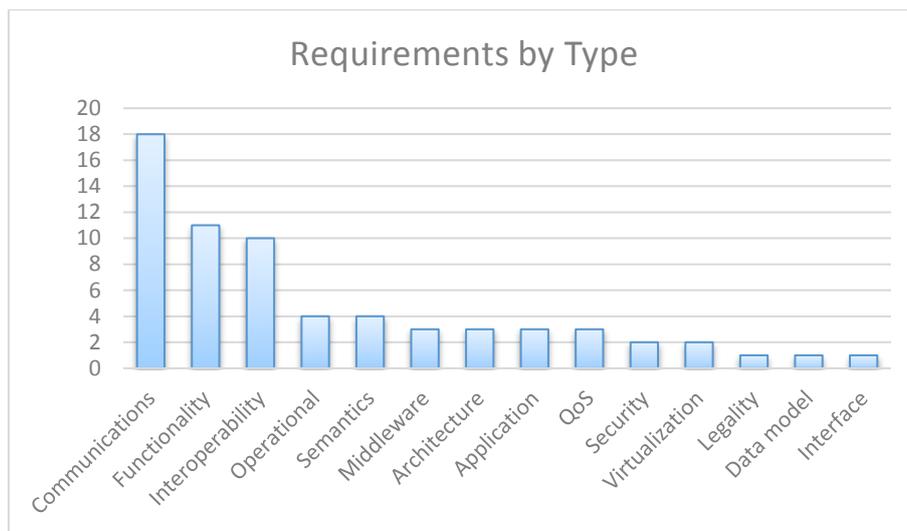


Figure 9: INTER-LAYER requirements by type

The chart in Figure 9 describes the apportionment of the requirements by type. We can notice that the majority of identified requirements are related with communications, as we are involved in an interoperability project, it is logical to pay high attention in how each element communicates with each other, and together with the interoperability requirements they cover more than a third part of the scenario. Additionally, functionality and operational requirements are relevant, being those related with how the system properly works.

3.1.4 Analysis

In the following section we describe in detail the results provided by the aggregation of information by categories, priority and source.

Category

Following the analysis of requirements, we separate them, as we have done before, by category.

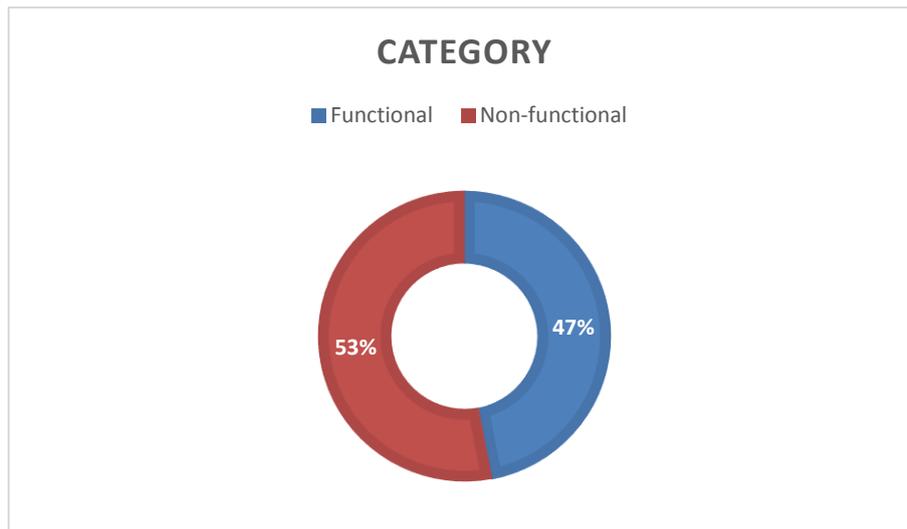


Figure 10: INTER-LAYER requirements by category

The graph in Figure 10 shows that more than a half of the requirements are non-functional, leaving the functional ones in a more reduced load.

Priority

Relying on the level of importance or need each requirement has within the product; we can classify them by its priority as follows.

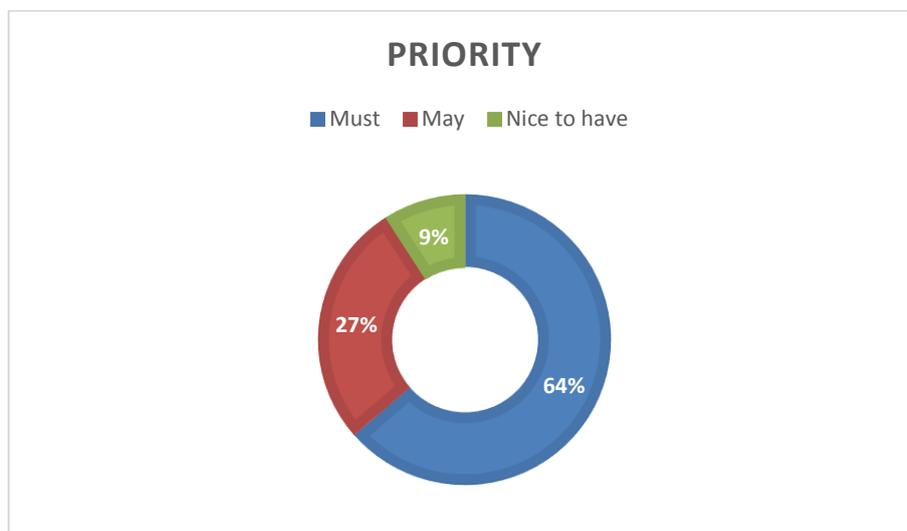


Figure 11: INTER- LAYER requirements by priority

According to the Figure 11, more than a half of the requirements are absolutely necessary and should be accomplished, and just 18 of them "May" be implemented and 6 are "Nice to have".

Priority by category

Considering the previous dissection, we can sub-divide the requirements by category and priority and obtain a well-delineated overview of the situation.

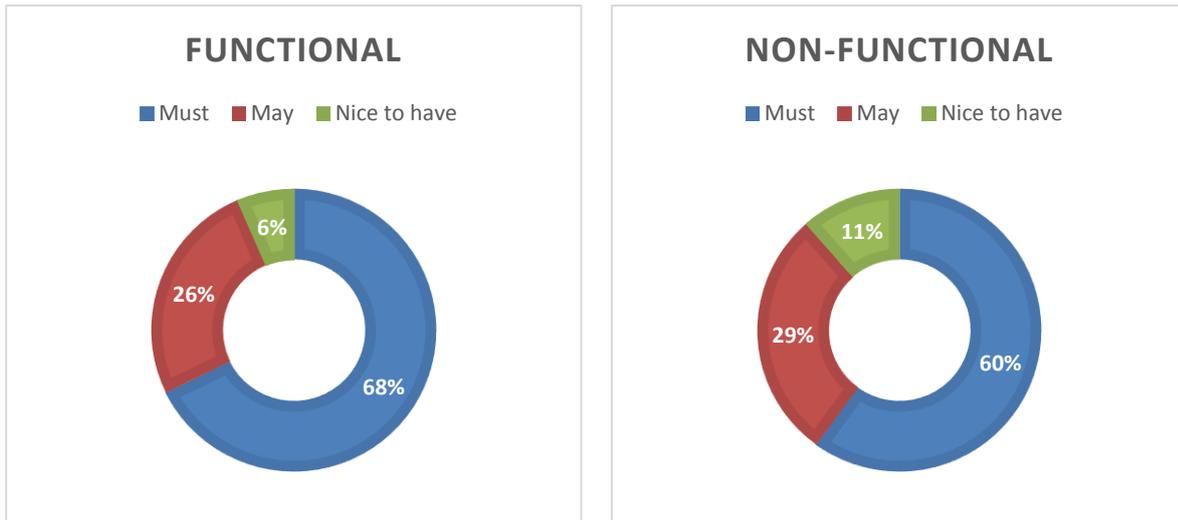


Figure 12: INTER- LAYER requirements by priority and category

The Figure 12 describes the priority of the requirements if they are functional or non-functional differently. In both cases the majority of them are a “Must” to implement in the development of the product, 21 in each case. For the first case, a third are “May” or “Nice to have” and for the second case, being the number of non-functional requirement slightly superior, still 10 may and 4 nice to have requirement are identified.

MoSCoW priority

In the Figure 13 is described the priority of the requirements following the Moscow methodology. Unlike the Figure 11 the “Must Have” requirements are about 30% of them. These are the core functionality of INTER-LAYER and the first part to be developed. Thus, “Should Have” requirements are about 40% and “Could Have” around a third.

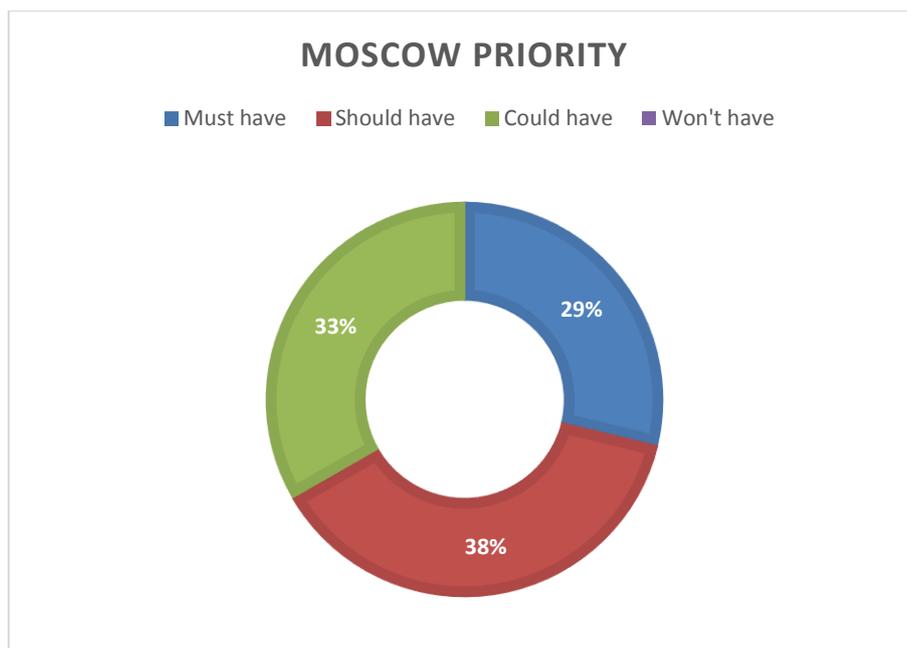


Figure 13: INTER-LAYER requirements by MoSCoW priority

Sources

Finally, each requirement has been provided by a source of information or experience. We can use this label to classify them and show where each of the requirements comes from.

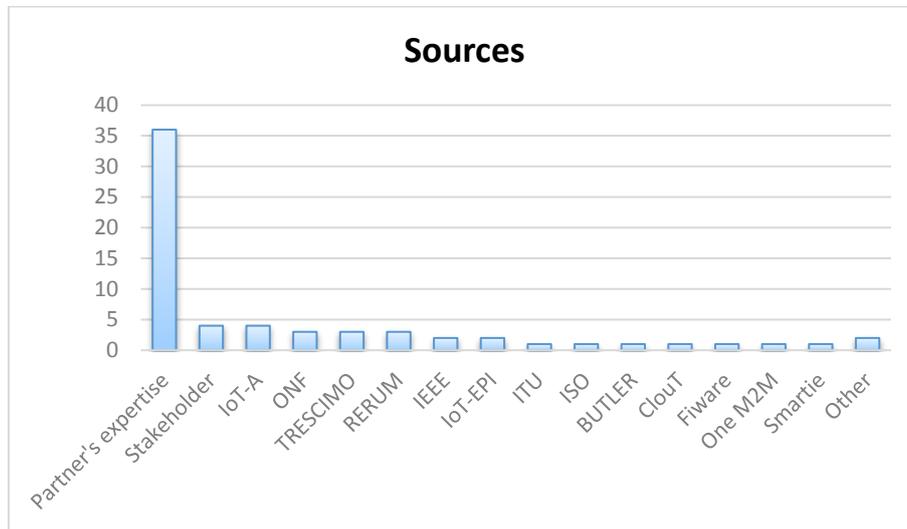


Figure 14: INTER- LAYER requirements by source

Is strongly notable, in the Figure 14, the Partner's expertise as a main source of information to derivate the requirements of the product, since almost half of them comes from this origin. To a lesser extent we can find important IoT projects as IoT-A as a source of information, along with organizations like ITU or IEEE. Still few requirements have been identified by other products, projects and organizations.

3.2 INTER-FW requirements

This product is a direct consumer of the INTER-LAYER product and lies on top of it within the INTER-IoT system. The objectives of this product are 1) to provide top-level interfaces for all the layer interoperability modules, 2) to develop common components and subsystem to help the platforms integration and third-parties application development and 3) to provide a common architectural reference with a common data model to make the INTER-IoT system flexible and adaptable.

The requirements gathered in this section are a preliminary list of needs for a common gluing platform that makes all the INTER-LAYER submodules operate within a common framework with a uniform interface. Although the concept of INTER-FW is key to help data consumer and producers to interact in a common scenario, some requirements are still immature because of the early stage of the development on the other software components of INTER-IoT. A continuous revision of these requirements are done as part of the software development process in a continuous contact with stakeholders (partners with the role of data providers, product consumers and end users) according to the AGILE development principles.

3.2.1 Non-functional requirements

The non-functional requirements have been classified according to the global solution of INTER-IoT. Most of them are related to basic features that, despite they are not critical for the main objective of the project, which is the interoperability of platforms, they are so for the project success, such as API availability or scalability.

Heterogeneous information representation [42]
<p>INTER-FW must provide information from different sources.</p> <p>The method of integration of multiple information and knowledge representing the same real-world sensing object into a consistent, accurate, and useful representation. It helps to fully take the usage of the IoT information resources for different application and service within an IoT system or between different information systems.</p>
<p>Acceptance criteria:</p> <p>Different information representation at different actors of the pilots is able to share information consistently. A cross-pilot with different information semantics and representation demonstrates the correct implementation of this requirement.</p>
MoSCoW Priority: Must have

API REST [52]
<p>INTER-FW must provide an Rest API.</p> <p>The public API of the framework follows the RESTful design principles.</p>
<p>Acceptance criteria:</p> <p>To have a REST API implemented and documented in a swagger-like tool.</p>
MoSCoW Priority: Must have

Provision of authentication credentials [63]
<p>INTER-FW must provide an authentication access with credentials.</p> <p>As regards the management of the User credentials, the platform has to:</p> <ul style="list-style-type: none"> - allow access only through individual authentication credentials (consisting of a User ID and an authentication device, e.g. Password); - prevent the reassignment of User ID to another user; - allow the definition of access profiles sets that guarantee the principles of "need to know" and "segregation of duties"; - allow the extraction of the information required to verify the correct allocation of authentication credentials and their authorization profiles; - carry out automatic checks at least monthly of the users inactive for more than six months in order to suspend, unless the users for which it has been required and authorized a derogation on the basis of an operational need.
<p>Acceptance criteria:</p> <p>Have a secure and efficient authentication system.</p>
<p>MoSCoW Priority: Must have</p>

Access to personal data needs to be previously authorized [263]
<p>INTER-FW must check permission for accessing to personal data.</p> <p>Access to personal information must be previously authorized by the owner.</p>
<p>Acceptance criteria:</p> <p>System does not allow access to personal data without previous permission of the owner.</p>
<p>MoSCoW Priority: Must have</p>

Requests filtering [280]
<p>INTER-FW must provide Requests filtering in the API.</p> <p>When sending a requests to INTER-FW, it is possible to specify filtering: The system shares a common filter format when possible. This filtering allows:</p> <ul style="list-style-type: none"> - Selection of platform(s). - Selection of device(s). - Selection of property type(s). - Selection of property filtering(s). - Selection of geo-queries (if allowed by the IoT platform).
<p>Acceptance criteria:</p> <p>All the previous filtering types are accessible in the API and performed to the underlying connected IoT platforms.</p>
<p>MoSCoW Priority: Must have</p>

Scalability. Computing resources [3]
<p>INTER-FW should be scalable.</p> <p>Scalability relates to the ability of systems to seamlessly cater for higher demand in computing resources of data, devices, people and applications.</p> <p>The implemented mechanisms should ensure low communication overhead, ensuring fast decision making and high performance.</p> <p>Scalability has two approaches that must be targeted:</p> <ul style="list-style-type: none"> - Allowing an increasing number of devices/platforms in a specific deployment. - Allowing a flexible deployment strategy to support from a small number of devices/platforms to a large number of them.
<p>Acceptance criteria:</p> <p>A scalable system with 500 devices sending real-time data from one or more platforms should be managed by INTER-FW.</p>
<p>MoSCoW Priority: Should have</p>

Alignment with AIOTI architecture and view [4]
<p>INTER-FW should be aligned with AIOTI architecture.</p> <p>AIOTI architectural model is suitable for guiding the development of INTER-IoT architecture. The use of AIOTI view of the architecture of Internet of Things is useful, in order to utilize its results and from other projects to avoid re-inventing a new architectural model from scratch, and to be aligned and compatible with those projects.</p>
<p>Acceptance criteria:</p> <p>Clear alignment of INTER-IoT's architecture with the AIOTI architecture view and the architecture used in other projects and standardization bodies.</p>
<p>MoSCoW Priority: Should have</p>

Extensibility [10]
<p>INTER-FW should be extensible with new functionalities.</p> <p>The system should expose functionality to the infrastructure maintenance to update the functionality when needed with new INTER-FW versions, without affecting existing clients.</p>
<p>Acceptance criteria:</p> <p>INTER-FW must be able to incorporate new functionalities without affecting an existing client that uses the previous functionality.</p>
<p>MoSCoW Priority: Should have</p>

Confidentiality [69]
<p>INTER-FW access should be confidential.</p> <p>Avoid data falsification or disclosure.</p>

<ul style="list-style-type: none"> - If the need of data processing ended, such data must be deleted permanently and irreversibly in order to prevent unauthorized treatment. - It must be guaranteed the logical isolation of data belonging to different customers on a single platform. In particular, it must be guaranteed the segregation of single customer views, in order to allow processing of data only to persons in charge of the processing (preventing access / views by unauthorized persons). - Special procedures for extraction and transmission of the data processed by the platform must be available. - In order to ensure the confidentiality of data stored in the platform encryption must be provide of identification codes or other solutions that make health data unintelligible to those who are authorized to access (i.e. identification data decoupled from health / sensitive ones).
<p>Acceptance criteria:</p> <p>The different actors can access the objects for which access was granted, and not to those it hasn't. Besides the normal positive accessibility tests, tests including negative ones, particularly with colliding device names and other potential issues.</p>
<p>MoSCoW Priority: Should have</p>

<p>API for proprietary systems interoperate with other systems [86]</p>
<p>INTER-FW should have an API for interoperability with proprietary systems.</p> <p>An upgrade or import-export process is necessary to integrate proprietary systems with the new systems. Could be done with an API from platform to platform or with a firmware upgrade of the devices.</p>
<p>Acceptance criteria:</p> <p>API that allow interoperability among platforms.</p>
<p>MoSCoW Priority: Should have</p>

<p>Implementation must be done by phases and progressively [91]</p>
<p>INTER-FW implementation should be done progressively.</p> <p>The process of implementation a new IoT protocol has to be compatible with both at the same time, i.e. at least it should have a gateway between the new and the old systems.</p>
<p>Acceptance criteria:</p> <p>Create a process that let all the device runs while the migration is progressively done.</p>
<p>MoSCoW Priority: Should have</p>

<p>Data provenance [98]</p>
<p>INTER-FW should ensure the data provenance.</p> <p>Data provenance metadata should allow to identify what is the origin data e.g. which artifact collects the data.</p>
<p>Acceptance criteria:</p>

Accessing data in IoT platforms ecosystems should allow tracking which artifact gathered the data.
MoSCoW Priority: Should have

Portability [132]
INTER-FW should be portable to different systems. Service providers must be able to switch between customers / users.
Acceptance criteria: Different systems recognize the same user.
MoSCoW Priority: Should have

Future-proof [278]
INTER-FW should work with previous versions. Future-proof: Future versions of the protocol must work with prior versions and provide all the same capabilities as prior versions.
Acceptance criteria: Features are implemented and working in the prototype, as validated by the pilots. In particular, the first and second release of the implementations should be backwards compatible, as old use-cases are tested in the new test-bed.
MoSCoW Priority: Should have

Auditability and Accountability [58]
INTER-FW could be auditable and accountable. Configured operations performed in the system must be tracked uniquely to the entity that generated it. The platform should allow: <ul style="list-style-type: none"> - To retrieve users and/or devices that carried out or are in charge of the activities in the system and their logged operations. - Producing an Audit log with trace of the most important data and their values before and after changes; - Maintain records for a period not less than six months; - Provide synchronization technologies in order to keep aligned the date and time recorded in the logs associated with the access. The criteria for registration of the aforesaid Log (products so as to not be editable) must at least enable the identification: <ul style="list-style-type: none"> - The event that triggered the log (login, logout, login failure); - The user, the date and the start / end connection. - The sensitive data updates (before and after).

Acceptance criteria: Allow auditability and accountability anytime, to ensure security operational aspects.
MoSCoW Priority: Could have

QoS Integration [142]
<p>INTER-FW could provide QoS.</p> <p>When appropriate, the IoT architecture should use the Quality of Service (QoS) features supported by underlying networks. This allows for all underlying networks to offer the same level of quality thus enhancing homogeneity.</p>
Acceptance criteria: Identify QoS of all underlying networks and integrate all corresponding QoS features to the entire system. A checklist of all the QoS features of the system should be drawn.
MoSCoW Priority: Could have

3.2.2 Functional requirements

Functional requirements identified depict some features that cannot be dismissed if the system is wanted to work properly. These represent the basic structure of the future software design and implementation. Due to the early phase of the project, more functional requirements could be identified as the WP3, WP4 and/or WP5 developments advance.

Allows roaming across platforms [1]
<p>INTER-FW must allow roaming across platforms.</p> <p>Objects that are moving can switch platform to which they are connected. Change between a platform and the other should be automatic and transparent to the device.</p> <p>From INTER-FW point of view, a moving device could be set as 'roamable' to specify INTER-LAYER that if it's not available in the expected platform, it should try to discover it in the rest of connected platforms and update the device registry.</p>
Acceptance criteria: Check that a moving object changes the IoT platform it's connected to, and a client of INTER-FW can query about it, transparently.
MoSCoW Priority: Must have

API for third-party developers [47]
<p>INTER-FW must provide an API for third-party developers.</p> <p>INTER-FW shall expose a public API for third-party developers and any potential INTER-IoT user.</p> <p>This API is the only way to access the functionality offered by all the components in the system.</p>

Acceptance criteria:

API resources can be accessed from external parties (different from the project consortium or the Open Call participants).

MoSCoW Priority: Must have

API for data publication [51]

INTER-FW must provide an API for data publication.

To have access at the application & services layers and at the semantic level from external IoT platforms. The API should be used for publishing sensor data into an existing IoT platform through an API without knowing anything about protocols.

Acceptance criteria:

To have an API available for publishing sensor data into an existing IoT platform, specifically into a previously created sensor registered into the platform.

MoSCoW Priority: Must have

Design of required ontologies [186]

INTER-FW must use a common ontology.

Use of required ontologies - a generic ontology of the Internet of Things. Creation of GOIoT, a global IoT ontology, providing common understanding of the IoT (generic) meta-structure, and enabling semantic interoperability. It is required to be designed or chosen from available ones in order to produce semantic alignment. GOIoT is based on current main IoT ontologies, such as W3C SSN, SAREF, etc.

Acceptance criteria:

Within INTER-IoT at least generic ontology of the Internet of Things is available. INTER-LAYER must offer semantic interoperability through the creation and use of a Central Ontology.

MoSCoW Priority: Must have

Each data unit is identified univocally [254]

INTER-FW must have unique data units.

Each minimal unit of meaningful data transmission (e.g. a heart rate measurement or a truck location event) must contain an identifier allowing retrieve the source of data and the network/platform for traceability.

Acceptance criteria:

Data transmissions have an identifier for traceability.

MoSCoW Priority: Must have

Each sensor has a unique INTER-IoT identifier [256]
<p>INTER-FW devices must have a unique identifier.</p> <p>An identifier system must be developed to be able to identify each device.</p> <p>Granularity in identification must reach the device level.</p> <p>Device-level services are provided at INTER-FW level [22] [33] [34] [35].</p>
<p>Acceptance criteria:</p> <p>Each device can be uniquely identified. Furthermore, there should not be a limitation to the number of devices that can be connected due to the lack of identifiers. The amount of identifier codes must be sufficiently large to accompany all current and future devices.</p>
<p>MoSCoW Priority: Must have</p>

Provides a sensor-level interface [259]
<p>INTER-FW must provide functions for interoperability of devices.</p> <p>Device interoperability can be managed by the INTER-FW with tools/APIs to do it. These APIs enable to access single devices and perform the different operations about it.</p>
<p>Acceptance criteria:</p> <p>The device interoperability functions are available at INTER-FW level.</p>
<p>MoSCoW Priority: Must have</p>

Manage user permission [260]
<p>INTER-FW must manage user permission.</p> <p>User permissions are managed at framework level.</p>
<p>Acceptance criteria:</p> <p>There is a field for user permission in the user configuration.</p>
<p>MoSCoW Priority: Must have</p>

API allows create/update/remove users [264]
<p>INTER-FW must provide an API to create/update/remove users.</p> <p>API to allow applications to implement user management tools.</p>
<p>Acceptance criteria:</p> <p>Allow methods for CRUD users by using the API.</p>
<p>MoSCoW Priority: Must have</p>

API allows device declaration and configuration [265]
<p>INTER-FW must provide an API to declare and configure devices.</p> <p>API to allow applications to implement device management tools.</p>

Acceptance criteria:

Allow methods for CRUD devices by using the API.

MoSCoW Priority: Must have

API allows resources/capabilities discovery [266]

INTER-FW must provide an API to discovery resources.

API allow applications to discover resources and capabilities of the platforms.

Acceptance criteria:

Allow methods for discover devices, platforms and capabilities of them.

MoSCoW Priority: Must have

API allows subscription to data streams/queues [270]

INTER-FW must provide an API to subscribe to data streams.

If data streams or queues are available, third parties can discover and connect to them to develop their own applications.

Acceptance criteria:

Allow methods for subscription to synchronous data sources.

MoSCoW Priority: Must have

Users manage how their public data is seen [77]

INTER-FW should allow users to select which of his public data can be seen.

Devices/IoT platforms as data sources are owned by different third parties. The owner of the object should be able to manage who and when other users have access to their information.

IoT platforms should support data ownership management, data-flow monitoring, and access management. Data visibility is managed according to owning entities policies. This is managed globally (platform independent)

At the configuration of an IoT platform registered into INTER-IoT, the software integrator may be able to specify a list of devices and/or operations which are accessible from external agents through INTER-IoT, how long, with whom, etc.

Acceptance criteria:

The user has tools to manage the access to their devices and platforms capabilities.

MoSCoW Priority: Should have

Semantic support for virtual smart objects [223]

INTER-FW should support ontology for virtual objects.

INTER-IoT ontology, GOIoTP, includes support for smart objects that are not sensors, but act as smart devices, such as virtual devices, human interfaces or algorithms. Many ontologies do not include objects that are not sensors, although they are potential and relevant IoT smart objects.

Acceptance criteria:

INTER-LAYER should be able to include at the semantic layer other smart objects rather than sensors, such as virtual devices or human interfaces.

MoSCoW Priority: Should have

Location semantic support for mobile smart objects [224]

INTER-FW should provide location semantic support for mobile objects.

Special field in the semantic specification of an IoT object indicating the location of the device. This is important information regarding to IoT smart objects, especially in the case of mobile sensors and devices, that it is not paid attention in main ontologies, but has a considerable relevance in IoT.

Acceptance criteria:

INTER-LAYER should include the representation of the geolocation in its semantic ontology, GOIoTP, as it is a relevant datum to handle in IoT.

MoSCoW Priority: Should have

Hops between platforms avoid data losses [253]

INTER-FW should avoid data losses in hops between platforms.

Hops across frameworks need to be achieved with full data availability.

Acceptance criteria:

There are implemented mechanisms to avoid data losses when a device is changing of platform/network.

MoSCoW Priority: Should have

The INTER-IoT unique ID is used to find the platform-specific ID of the device [257]

INTER-FW should find a platform-specific device ID from the INTER-IoT unique ID.

The platform specific ID of each element needs to be retrieved from a unique ID assigned in INTER-IoT. This ensures traceability.

Acceptance criteria:

A platform-specific ID can be retrieved from an INTER-IoT ID.

MoSCoW Priority: Should have

API allows historic data query [268]

INTER-FW should provide an API to query historic data.

API allows to query historic data for reports, charts, etc.

Acceptance criteria:

Allow methods for database-like querying.

MoSCoW Priority: Should have

Stores recent data for recovery [272]

INTER-FW should store recent data for recovery.

A copy of the recent data shared among platforms is stored in a given time window.

Acceptance criteria:

Shared data must be able to be recovered after a connectivity failure.

MoSCoW Priority: Should have

Stores system status for recovery [273]

INTER-FW should store system status for recovery.

The system state and configuration is persisted allowing to recover it after an unexpected failure.

Acceptance criteria:

Status data must be able to be recovered after a platform failure.

MoSCoW Priority: Should have

Time triggers [276]

INTER-FW should support time triggers for improving the communications.

It must be possible to provision timers in the protocol stack to indicate when to re-send a packet that has not been confirmed. These timers should be individually provision according to the destination address in the packet.

Acceptance criteria:

Time triggers are implemented and working in the prototype, as validated by the pilots. A mock-up component can be set to not return ACKs and test the functionality.

MoSCoW Priority: Should have

A user knows its permissions [261]

INTER-FW could allow a user to know its permissions.

The permission to access system resources are known by the user at INTER-FW level.

Acceptance criteria:

There is a method to ask the framework about the permissions of the current user.

MoSCoW Priority: Could have

Manages group-based permissions [262]
INTER-FW could manage permissions for groups. Permissions can be managed at group level in order to simplify business processes.
Acceptance criteria: Users can be managed in groups.
MoSCoW Priority: Could have

Indivisibility [277]
INTER-FW could transfer a sequence of packets as a logical unit. The protocol must have a means to transfer a sequence of packets as a logical unit, like a firmware upgrade, a data log, or provisioning information.
Acceptance criteria: Features are implemented and working in the prototype, as validated by the pilots.
MoSCoW Priority: Could have

3.2.3 Requirements by type

In the INTER-FW product, up to 11 different types of requirements have been identified, according to its main purpose or the function they perform in the INTER-FW design.

Architecture

- Scalability. Computing resources [3]

Interoperability

- Allows roaming across platforms [1]
- Alignment with AIOTI architecture and view [4]
- API for proprietary systems interoperate with other systems [86]
- Portability [132]
- Hops between platforms avoid data losses [253]
- Each data unit is identified univocally [254]
- Each sensor has a unique INTER-IoT identifier [256]
- The INTER-IoT unique ID is used to find the platform-specific ID of the device [257]

Privacy

- Users manage how their public data is seen [77]

Security

- Provision of authentication credentials [63]
- Confidentiality [69]

- Data provenance [98]

Privacy/Security

- Manage user permission [260]
- A user knows its permissions [261]
- Manages group-based permissions [262]
- Access to personal data needs to be previously authorized [263]

Semantics

- Heterogeneous information representation [42]
- Design of required ontologies [186]
- Semantic support for virtual smart objects [223]
- Location semantic support for mobile smart objects [224]

Usability

- Auditability and Accountability [58]

Functionality

- Extensibility[10]
- Stores recent data for recovery [272]
- Stores system status for recovery [273]
- Time triggers [276]
- Indivisibility [277]
- Future-proof [278]
- Requests filtering [280]

QoS

- QoS Integration [142]

API

- API for third-party developers [47]
- API for data publication [51]
- API REST [52]
- INTER-FW provides a sensor-level interface [259]
- API allows create/update/remove users [264]
- API allows device declaration and configuration [265]
- API allows resources/capabilities discovery [266]
- API allows historic data query [268]
- API allows subscription to data streams/queues [270]

Methodology

- Implementation must be done by phases and progressively [91]

According to this, we can find the following distribution by categories, shown in Figure 15:

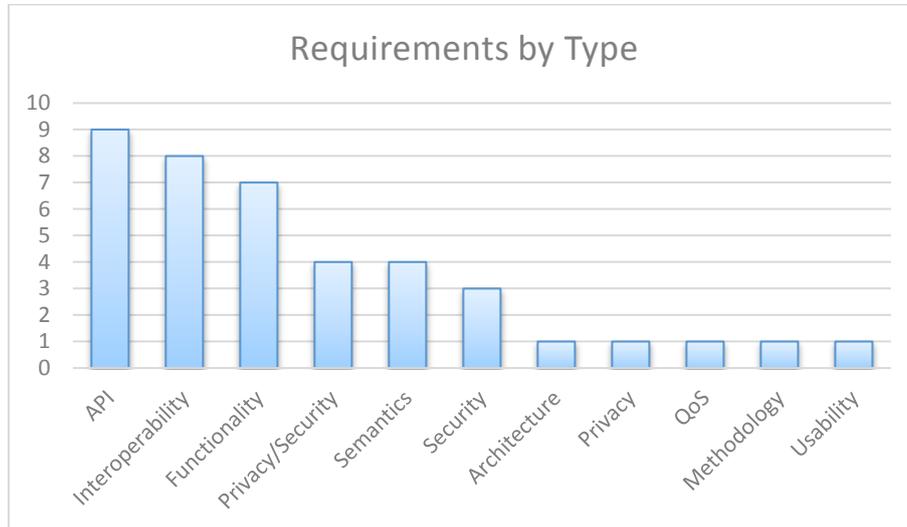


Figure 15: INTER-FW requirements by type

While some of the categories could seem redundant, the reason of this classification is that some requirements share properties of different categories. Three main groups of requirements can be identified: API, interoperability and functionality. They group about a 35% of the requirements, what gives an idea of what the partners and stakeholders consider the key characteristics of the INTER-FW product. Security and privacy related requirements suppose near the 20% percent of the total, so that the fourth important pillar of the INTER-FW is related to the data protection.

3.2.4 Analysis

After the requirements specification, some parameters can be analysed. A statistical analysis of the requirements by some fields can show if the distribution is uniform or not and the relevance that the partners have given to the extracted requirements.

Category

Classification by categories (see Figure 16) reveals that there is a good balance among functional and non-functional. INTER-FW is a product that aggregates the subsystems of INTER-LAYER and provides common accessors, APIs, common services, etc. From a global perspective (i.e. by observing INTER-IoT as a whole), most of these features can be seen as functional (e.g. APIs), and, therefore, other non-functional requirements have been identified by the partners.

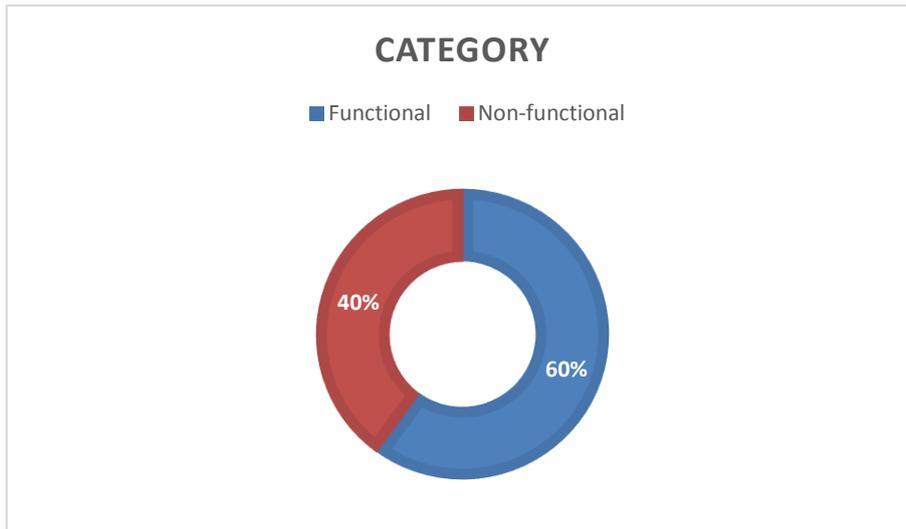


Figure 16: INTER-FW requirements by category

Priority

If the requirements are analysed by priority (see Figure 17), most of them have been identified as “must”, a number of “may” and only a few “nice to have”. Since this document and the requirement acquisition are being performed in the early stages of the project, and the INTER-FW product is a development that lies on the top of INTER-LAYER and will be performed in mid and late stages of the project, is normal that only *musts* and *mays* are identified, since only the initial draft of the product is envisaged and the partners can’t go in deep detail to identify other less priority but *nice to have* requirements.

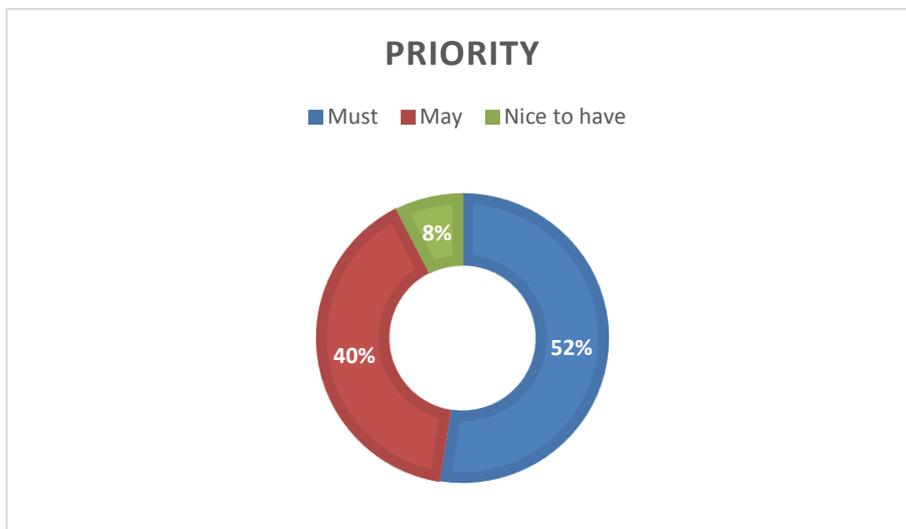


Figure 17: INTER-FW requirements by priority

Priority by category

Finally, there have been split the categories and analysed the priorities on them. As expected, both categories are similarly balanced, since at this stage of maturity, only important requirements can be easily identified.

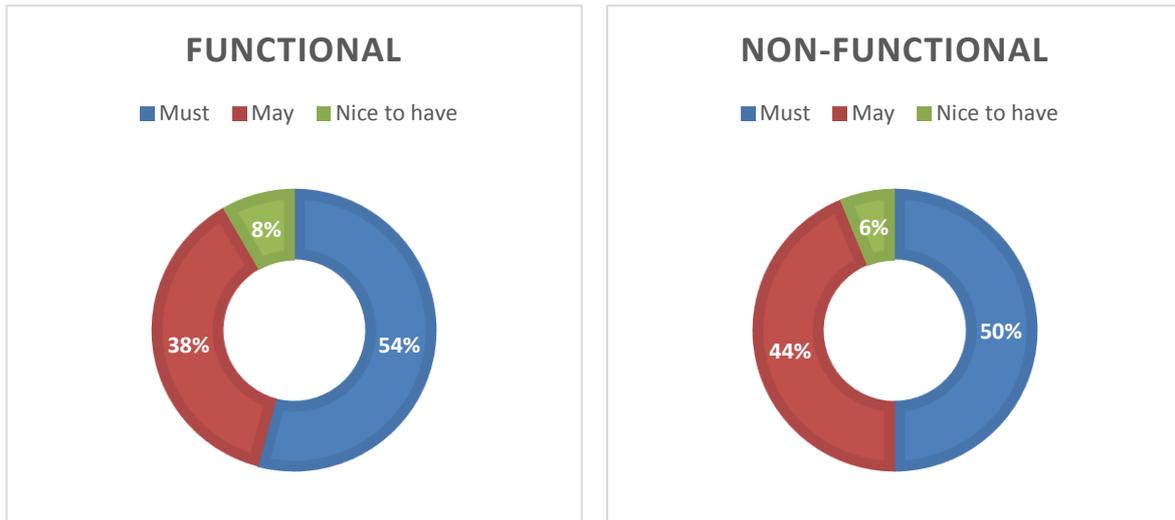


Figure 18: INTER-FW requirements by priority and category

MoSCoW priority

The priority of the requirements is also analysed following the Moscow methodology. In the Figure 19 can be seen that most of the requirements are divided into “Must Have” (42%) and “Should Have” (45%). There are is only a few of them as “Could Have”



Figure 19: INTER-FW requirements by MoSCoW priority

Sources

The source of the requirements is rather fragmented. However, three big groups can be identified: partners’ expertise, which is the origin of near the 35% of the requirements, other research projects and regulation and standardization entities.

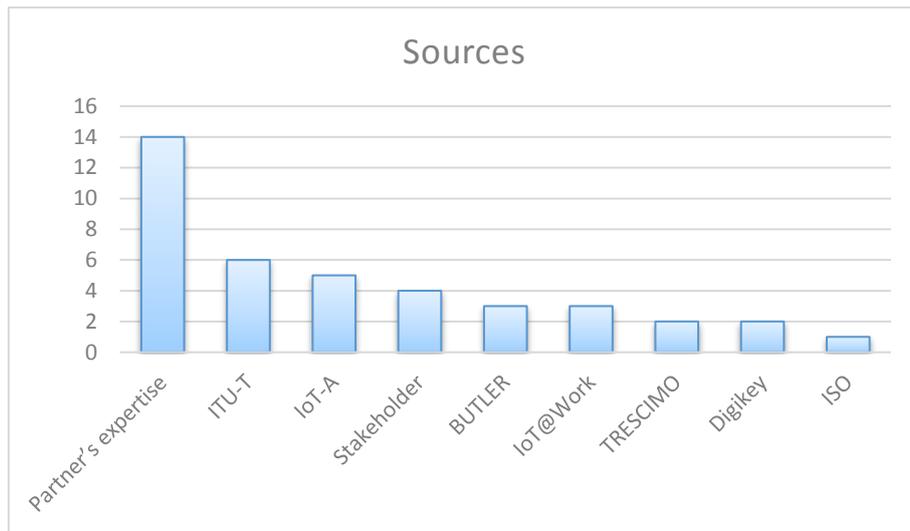


Figure 20: INTER-FW requirements by source

3.3 INTER-METH requirements

In this section we describe the requirements related to the INTER-METH product. They are classified into the classical two main categories (non-functional and functional) and ordered by priority. Since the methodology component of INTER-IoT exposes particular features with respect to a conventional software/system, the requirements identification process has produced interesting results, presented in the following subsections.

3.3.1 Non-functional requirements

In this subsection the non-functional requirements related to the INTER-METH product have been elicited. They represent INTER-METH "quality attributes" which are observable and testable at run time (execution qualities, e.g. usability and scalability) or which are embodied in the static structure of the software system and are independent from the execution time (evolution qualities, such as documentation and compliance to standards).

Open Source [108]
<p>INTER-METH must be open source.</p> <p>An extract of the European Commission strategy in matter of software products is here reported: "Software produced by the Commission services, in particular software produced with the objective of being used outside the Commission, is open sourced and published on the Join up platform and uses the European Union Public License (EUPL)". In general, open source projects may rely on wide and active communities of users and developers (in particular, THINGS underlines the need of "Open API" to improve usability, extensibility and interoperability).</p> <p>According to this vision the INTER-Meth documentation, the INTER-CASE software tool and related documentation should be open.</p>
<p>Acceptance criteria:</p> <p>Source code of the INTER-CASE tool is available on INTER-IoT website.</p>
<p>MoSCoW Priority: Must have</p>

Stability [109]
<p>INTER-METH must be stable.</p> <p>INTER-METH is released in a (reasonably) stable version in order to avoid frequent changes and to provide long-term support to IoT systems/platforms integration/interconnection and to integrated/interconnected IoT systems/platforms.</p>
<p>Acceptance criteria:</p> <p>INTER-METH released version is stable.</p>
<p>MoSCoW Priority: Must have</p>

Usability [110]
<p>INTER-METH must be usable.</p> <p>INTER-METH must be as much as possible easy to be understood, learned and used. This implies that the users' training effort must be reasonable. Moreover, INTER-METH must be</p>

attractive in order to positively influence other organizations to reach a broad adoption. Several stakeholders such as UPV and GIS (multinational solutions provider for government and institutions) highlighted this important aspect.

Acceptance criteria:

The methodology is usable according to well-established usability evaluation procedures.

MoSCoW Priority: Must have

Error minimization [113]

INTER-METH must minimize errors.

At methodological level, the way to minimize errors is to follow a “correct-by-construction” approach such that the methodology itself helps the IoT system designer/integrators to construct a (formal) model before that any implementation/integration detail is produced. The model is used to reason about the proposed solutions, ensuring that all the required functional and non-functional requirements are fulfilled and the correct behaviour of the integrated system exhibited.

Acceptance criteria:

Testing procedures should be devoted to validate the correct-by-construction platforms integration and, eventually conceptual errors should not be found at all in the designed system model.

MoSCoW Priority: Must have

Extensibility and Personalization [114]

INTER-METH must be extendable and customizable.

A robust methodology aiming at integrating different IoT platforms should be easily adaptable to any specific platform to be integrated. The extensibility and personalization features of such engineering methodology should allow a simple and fast integration among different heterogeneous non-interoperable IoT solutions.

Acceptance criteria:

Fast and easy customization to different IoT platforms.

MoSCoW Priority: Must have

Maintainability [119]

INTER-METH must be maintainable.

The IoT scenario is highly dynamic and INTER-METH aims to play an important role even in the next years. With reference to integrated platforms, the effort needed to isolate and correct defects, to make future focused modifications or to cope with a changed environment/standard must be reasonable.

Acceptance criteria:

Modification/Updates do not require unreasonable efforts.

MoSCoW Priority: Must have

Effectiveness and Optimization [120]
<p>INTER-METH must be effective and optimise integration.</p> <p>As highlighted by several stakeholders (CSE, ISECO, NEWAYS, INFOPORT, UNICAL), INTER-METH is expected to have a high-impact on “reducing development time and cost” but at the same time “enhancing effectiveness, agility and quality”. The IoT platforms integration process must be optimized and strengthened by the INTER-METH application.</p>
<p>Acceptance criteria:</p> <p>INTER-METH makes IoT platforms integration processes faster, higher quality and lower resource waste.</p>
<p>MoSCoW Priority: Must have</p>

Methodology and tools to integrate a proprietary IoT platform [169]
<p>INTER-METH must support the integration of proprietary IoT platforms.</p> <p>Every platform that wants to access the ecosystem of IoT platforms needs to have a methodology that explains the steps for integration at all levels.</p>
<p>Acceptance criteria:</p> <p>A detailed methodology with the steps to follow.</p>
<p>MoSCoW Priority: Must have</p>

Support for platforms with and without explicitly defined ontology [181]
<p>INTER-METH must support the integration of platforms without explicitly defined ontology.</p> <p>Two situations are considered: (i) platform(s) with explicitly defined ontology (or, at least, taxonomy); (ii) platform(s) with no explicitly defined ontology/taxonomy. To achieve semantic interoperability IoT platforms should exchange structural data with explicitly documented structure and semantics (by means of e.g. ontology conceptual model, XML Schema, JSON schema, database schema). INTER-METH must provide guidelines on how to handle the situation when no ontology is present but stated precondition is satisfied, as well as when IoT platform has defined ontologies.</p>
<p>Acceptance criteria:</p> <p>Platform with explicitly defined ontology (ontological description of data structure and semantics) can cooperate with other platforms via INTER-IoT. Platform without explicitly defined ontology (explicitly documented structure and semantics but not with ontology but e.g. XML schema) can be prepared to cooperate with other platforms via INTER-IoT. Methodology considers both cases and gives guidelines on how to attach them to IINTER-IoT, as well as indicate suitable tools to make the process easier.</p>
<p>MoSCoW Priority: Must have</p>

Tools / libraries to support design [31]
<p>INTER-METH should have tools & libraries to support design.</p> <p>It should include tools/libraries to support users in designing and implementing semantic and data interoperability in the reference architecture.</p>

Acceptance criteria:
A library that can be consulted to create new applications over the platform must be in place.
MoSCoW Priority: Should have

Transitional INTER-IoT protocols and tools [85]
INTER-METH should integrate with legacy systems. Main aspects of INTER-IoT protocols should be public to be adopted by new devices developments or the design of new devices.
Acceptance criteria:
Legacy systems can still work with the new IoT approach and interoperate with other platforms.
MoSCoW Priority: Should have

Enable (automated or semi-automated) linking of relevant data models [96]
INTER-METH should have links with relevant data models. Finding correspondences between data models facilitates application integration and enable interaction between artifacts.
Acceptance criteria:
Existence of procedures and recommendations for tools that can be used to make the linking process seamless, or automated / semi-automated in some parts.
MoSCoW Priority: Should have

Documentation [111]
INTER-METH should have documentation. As several stakeholders have highlighted (e.g. VPF, ABC, SRIPAS), INTER-METH should be supplied with a detailed documentation (in terms of “know-how knowledge, guidance, step-by-step instructions”) to improve its usability. Documentation is based on standard notations for software/system engineering methodologies (e.g. SPEM by OMG).
Acceptance criteria:
Complete and detailed documentation is delivered also according to a standard notation.
MoSCoW Priority: Should have

Online documentation in INTER-IoT new tools and services [87]
INTER-METH could have online documentation. A course or online course should be prepared to get easy access to the knowledge.
Acceptance criteria:
Provide online documentation for the new tools and services.
MoSCoW Priority: Could have

Compliance [112]
<p>INTER-METH could be compliant with standards.</p> <p>Over the years, several organizations (ISO, IEEE, IPC, OMG, etc.) provided successful standards and protocols. Some of them may be maintained and exploited from INTER-METH, as different stakeholders highlighted (e.g. OGC, CSE, PRO, Intel, AYAC, Agile, or SPEM). In fact, they may represent useful starting points and represent source of interoperability, avoiding unnecessary duplication of efforts.</p>
<p>Acceptance criteria:</p> <p>INTER-METH complies with standards considered consistent with the methodology itself.</p>
<p>MoSCoW Priority: Could have</p>

Scalability [115]
<p>INTER-MERTH could be scalable.</p> <p>Scalability is a fundamental feature of any distributed platform/system. Thus, INTER-METH provide support by-design to drive an integration of platforms that results into an integrated platform that preserves the maximum level of scalability.</p>
<p>Acceptance criteria:</p> <p>INTER-METH supports an integration of heterogeneous platforms that preserves scalability.</p>
<p>MoSCoW Priority: Could have</p>

3.3.2 Functional requirements

In this subsection the functional requirements related to the INTER-METH product have been elicited. They reflect the INTER-METH main functionalities (e.g. Case Tool support, Model Driven support) as well as some secondary but still important operations (e.g. Specification of unique attributability, Legal and licensing issues avoidability).

Ontology support [74]
<p>INTER-METH must provide ontology support.</p> <p>INTER-METH provides mechanisms to support through design semantic interoperability between: (i) platform(s) with explicitly defined ontology (or, at least, taxonomy); (ii) platform(s) with no explicitly defined ontology/taxonomy.</p>
<p>Acceptance criteria:</p> <p>INTER-METH realizes semantic interoperability.</p>
<p>MoSCoW Priority: Must have</p>

Security and Trust Management [117]
<p>INTER-METH must include security and trust related concerns.</p> <p>INTER-METH takes into account security-related concerns, namely the set of hardware, software, procedures, and policies components for defending and controlling access to integration of devices, networks, data/information, and software against malicious entities and attacks. Here security is related to the integrated platforms. For example, trust issues</p>

refer to scenarios in which different integrated/interconnected platforms cooperate without previous collaboration history. Trust management enables to make sure that the shared data and services/operations are real and trustworthy, especially with crowdsourcing generated data and machine generated data.

Acceptance criteria:

INTER-METH guides the integration of IoT systems/platform so preserving the overall security, trustability and protection to their devices, data, information, and software.

MoSCoW Priority: Must have

Design support for systematic IoT platforms integration/interconnection [159]

INTER-METH must support a systematic approach.

It is widely recognized that using an engineering methodology is fundamental in any engineering application domain, since the manual and non-systematic application of complex techniques, methods and frameworks would very likely lead to an increase of the degree of errors during integration. INTER-METH provides a methodology that systematically support the development of voluntary interoperability among heterogeneous IoT platforms (even belonging to different domains). In this direction, by means of different guidelines, models, facilities and tools, INTER-METH supports the systematic IoT platforms integration/interconnection in the development phases of analysis, design and implementation. The needs of a "systematic methodology and a well-defined approach to support IoT interoperability at any level of abstraction and within every application domain" have been underlined by several stakeholders (e.g. THINGS, XLAB, SRIPAS, ABC, VEMCO, etc.).

Acceptance criteria:

Integration of different platforms is driven by a common and well-defined methodology.

MoSCoW Priority: Must have

CASE-tool support [160]

INTER-METH must support a CASE-tool.

INTER-METH is supported by INTER-CASE (Computer Aided Software Engineering tool for integration) in order to foster the effective and efficient integration of different heterogeneous IoT platforms. It help to automate each phase (analysis, design, implementation, deployment, test, maintenance) of the integration process by using INTER-METH, thus providing guidelines, graphical facilities, engineering patterns and methods, and data repositories. The need of a CASE-tool supporting and, if possible, automating such processes has been underlined by several stakeholders (e.g. Telefonica, DGCONNECT, ISECO, DISI-UNIBO, Symblote, etc.).

Acceptance criteria:

The CASE-tool is developed according to the above mentioned functionalities.

MoSCoW Priority: Must have

Methodology for the new IoT platform attachment to the IoT Platform Semantic Mediator [184]
<p>INTER-METH must include the methodology for IPSM.</p> <p>Methodology for the new IoT platform attachment to the Inter Platform Semantic Mediator should be proposed. Methodology should address special cases e.g. platform with explicit ontology, platform without ontology.</p>
<p>Acceptance criteria:</p> <p>Methodology describes the procedure of attachment of a new IoT platform to INTER-IoT to achieve data and semantics interoperability. Depending on platform characteristics appropriate guidelines are given. Methodology list the conditions that should be satisfied by IoT platforms to be able to join.</p>
MoSCoW Priority: Must have

Model-driven support [161]
<p>INTER-METH should support model driven engineering.</p> <p>Model Driven Engineering (MDE) raises the level of abstraction in systems/programs specifications, facilitates the understandability of the main system concepts and increases automation in systems/programs development. INTER-METH approach is based on meta-models that are defined at different levels of abstraction to support the development phases of analysis, design, integration, and validation.</p>
<p>Acceptance criteria:</p> <p>Meta-models are available to support each development phase.</p>
MoSCoW Priority: Should have

Guidelines for formal documents data interchange formats, structures and services [185]
<p>INTER-METH should have guidelines for exchange data.</p> <p>Methodology should include guidelines on how to formally document IoT platforms data interchange formats and structures, and services descriptions in order to attach them to Inter Platform Semantic Mediator. IoT platforms make use different data formats and semantics, however each IoT platform should be able to provide formal descriptions e.g. by means of ontology conceptual model, XML schema.</p>
<p>Acceptance criteria:</p> <p>Methodology includes guidelines on how to formally document IoT platforms data interchange formats and structures, and services descriptions in order to attach them to Inter Platform Semantic Mediator.</p>
MoSCoW Priority: Should have

Privacy [116]
<p>INTER-METH could include privacy related concerns.</p> <p>INTER-METH takes into account the constraints deriving from the processing of personal and health data. INTER-METH defines and implements privacy policies to determine which information can be revealed in the integrated platform, who can access to such information, and for what purposes such information may be used.</p>
<p>Acceptance criteria:</p> <p>INTER-METH guides the development of a system that successfully provides different degree of freedom defining appropriate privacy policies and eventually guarantee high end-to-end privacy level.</p>
<p>MoSCoW Priority: Could have</p>

Legal and licensing issues avoidability [118]
<p>INTER-METH could include legal and licensing related concerns.</p> <p>There may be legal issues involving privacy of information, intellectual property rights, export of restricted technologies, patent-infringement, etc. As suggested by the DISI-UNIBO stakeholder, the methodology should made heterogeneous IoT platforms and products interoperable but in a manner that it “takes care of the all technological, organizational, ethical and legal constraints”.</p>
<p>Acceptance criteria:</p> <p>No complaints for patent/license/privacy violation are raised.</p>
<p>MoSCoW Priority: Could have</p>

Specification of unique attributability [162]
<p>INTER-METH could specify unique attributability.</p> <p>INTER-METH is the methodology supporting the interconnection /interoperability of heterogeneous IoT platforms. However, it may happen that for different reasons there is the need of identifying the single platform contribution to the system functionalities (e.g. two interconnected platforms provide the same service but the user have to choose a particular service provider). DICGIM UNIPA underlines such requirement.</p>
<p>Acceptance criteria:</p> <p>IoT platforms identities and functionalities are distinguishable.</p>
<p>MoSCoW Priority: Could have</p>

3.3.3 Requirements by type

In the following, we classified the requirements of the INTER-METH product by type in order to have a clear and concise picture of the importance of each category.

Usability

- Open-Source [108]

- Documentation [110]
- Usability [111]
- Extensibility and Personalization [114]

Functionality

- Tools / libraries to support design [31]
- Stability [109]
- Compliance [112]
- Error Minimization [113]
- Maintainability [119]
- Effectiveness and Optimization [120]
- CASE-tool support [160]
- Specification of unique attributability [162]

Methodology

- Methodology and tools to integrate a proprietary IoT platform [169]
- Methodology for the new IoT platform attachment to the IoT Platform Semantic Mediator [184]
- Guidelines for formal documents data interchange formats, structures and services [185]

Semantics

- Ontology support [74]
- Support for platforms with and without explicitly defined ontology [181]

Interoperability

- Transitional INTER-IoT protocols and tools [85]
- Enable (automated or semi-automated) linking of relevant data models [96]
- Design support for systematic IoT platforms integration/interconnection [159]
- Model-driven support [161]

Architecture

- Scalability [115]

Legality

- Legal and licensing issues avoidability [118]

Privacy

- Privacy [116]

Security

- Security and Trust management [117]

Commercial

- Online documentation in INTER-IoT new tools and services [87]

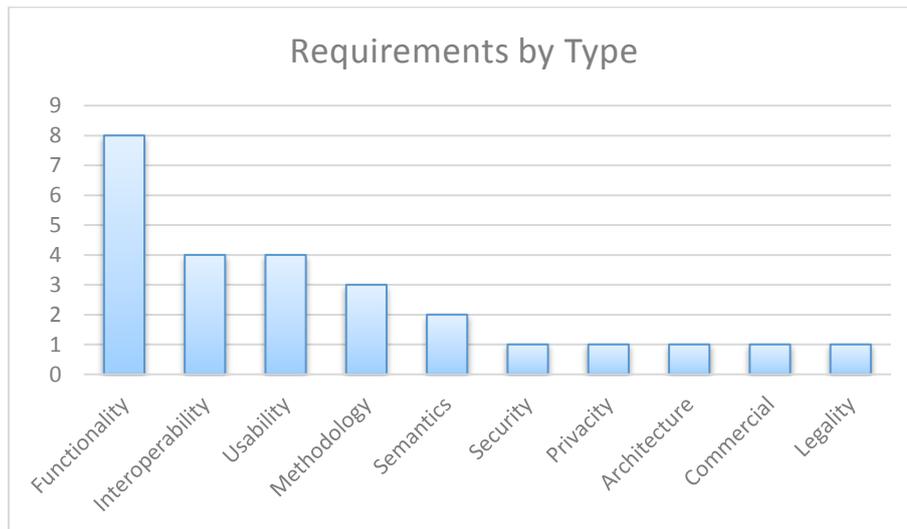


Figure 21: INTER-METH requirements by type

Since the goal of the INTER-METH is to provide an effective instrument to drive the voluntary interoperability among heterogeneous IoT platforms, it follows that most of the gathered requirements refer to the functionality, interoperability, usability and methodology.

3.3.4 Analysis

In the following subsections the gathered requirements have been clustered according to their category, assigned priority and source. This represents a good starting point, providing valuable indications for the actual development of the INTER-METH product.

Category

The INTER-METH product has been full-fledged analyzed, eliciting requirements belonging to all the expected categories.

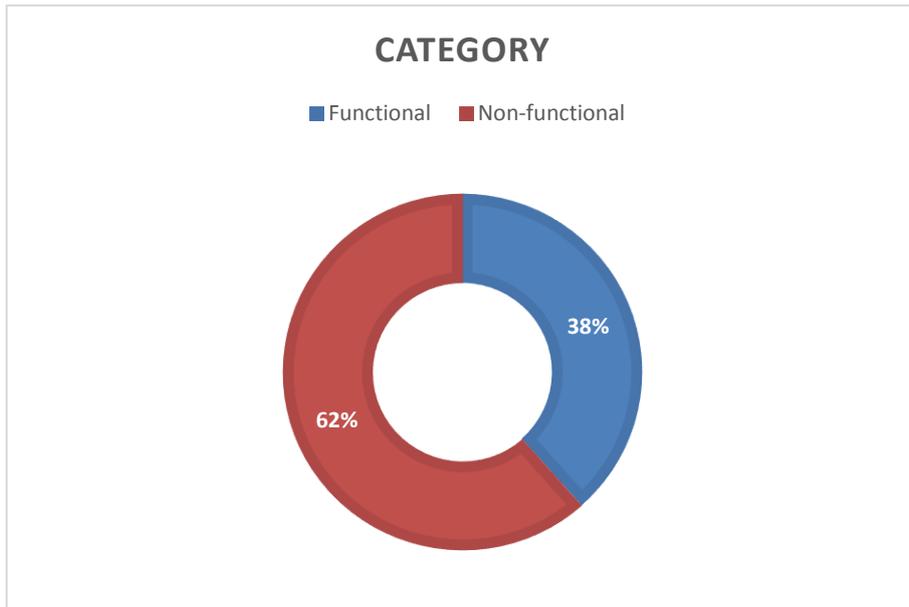


Figure 22: INTER-METH requirements by category

By looking the distribution between the different requirements categories (Figure 22), it is possible to note that the amount of non-functional requirements is greater than functional requirements.

Priority

Although the requirements gathering process has been thoughtfully conducted, in this early phase we followed a pragmatic approach by focusing mostly on the mandatory requirements. Further secondary requirements may be smoothly added at a later stage.

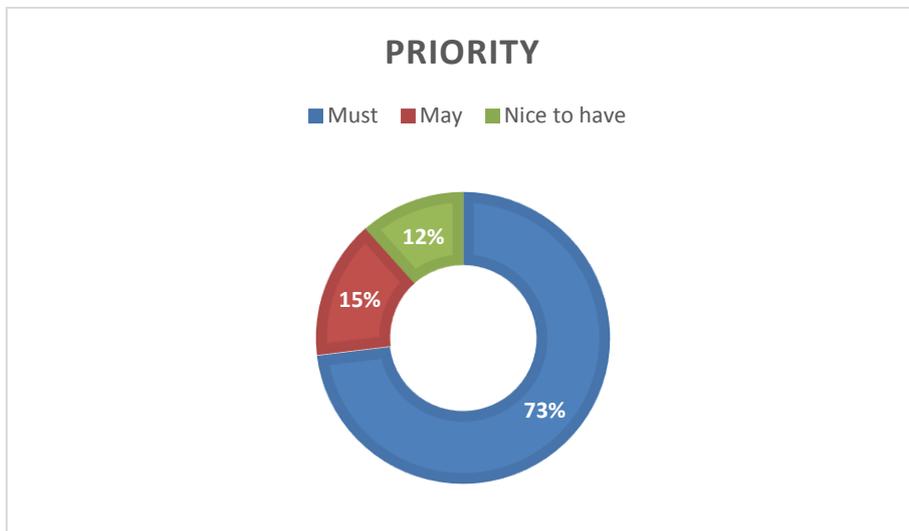


Figure 23: INTER-METH requirements by priority

Regarding the desired priority of the identified requirements, it is possible to note that most of them (i.e. 73%) need to be addressed in a mandatory fashion while only the 15% are preferable and a very little part (i.e. 12%) would be nice to have.

Priority by category

The requirements previously clustered on the basis of their priority or of their category, in the following are simultaneously clustered according to both criteria.

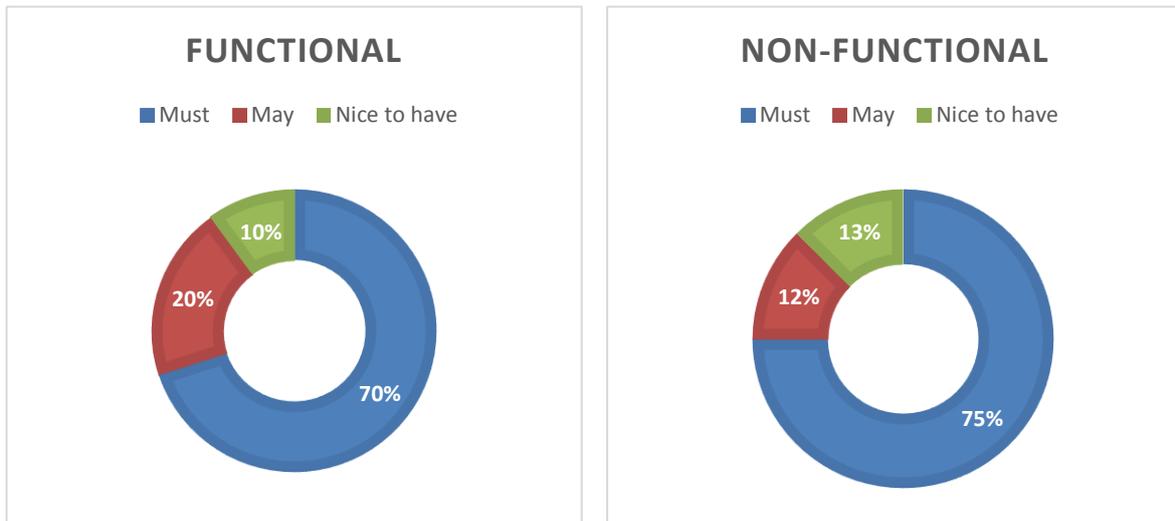


Figure 24: INTER-METH requirements by priority and category

By analysing both categories, it is possible to argue that the non-functional requirements are almost all mandatory to be satisfied during the implementation of the INTER-METH product. The preferable (Privacy and Legal and licensing issues avoidability) or the desirable (Specification of unique attributability) requirements are mainly functional.

MoSCoW priority

In the Figure 25 Figure 13is described the priority of the requirements following the Moscow methodology. In this case, around a half of them are “Must Have”. The rest are divided equally between “Should Have” and “Could Have”.

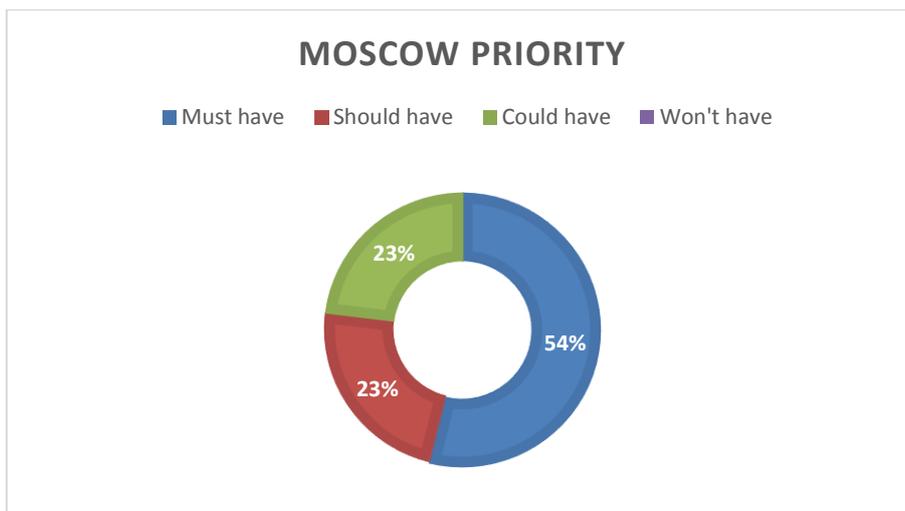


Figure 25: INTER- METH requirements by MoSCoW priority

Sources

The requirements gathering process has been strongly driven by the partner's expertise and by the indications provided by the stakeholders (in the preliminary phases) since the INTER-METH product represents a widely new need in the IoT arena.



Figure 26: INTER-METH requirements by source

In detail, Figure 26 shows as 16 of the sources of the identified requirements come from the partner's expertise rather than from well-established international standard organizations or general strategy adopted by the European Commission.

3.4 INTER-LogP requirements

The INTER-LogP product is different from the previous three, because it is focused on a specific domain. Therefore, the requirements are not so oriented to how the system works but to the functionality that it should have.

3.4.1 Non-functional requirements

In this subsection the non-functional requirements for the logistics and port transportation environment are presented.

Creation of new services or improvement of existing ones to access and utilize data from different platforms [248]
INTER-LogP must provide access to resources of a virtual entity from different platforms. Access to resources and services of a virtual entity from another IoT platform or application when certain rules are met. Modification of services or creation of new ones that take advantage of shared information.
Acceptance criteria: The port IoT platform, the terminal IoT platform and the haulier IoT platform need to exchange data about the trucks and containers entering in the port area. This data should improve the port management by offering faster and predictable services (e.g. the Estimated Time of Arrival). IoT platform has to coordinate with emergency systems.
MoSCoW Priority: Must have

High responsiveness [54]
INTER-LogP should have a high capacity of response. Users of INTER-LOGP need to receive a response in an appropriate timeframe.
Acceptance criteria: The appropriate response time related to logistics and transport scenarios are: 1 second (desired), 2 seconds (acceptable), 5 seconds (maximum).
MoSCoW Priority: Should have

Priority in alarms [84]
INTER-LogP should provide a priority in alarms. Alarms should go in priority way, not more than a second, and launch triggers.
Acceptance criteria: The system must provide a way to prioritize and the dispatch time needs to be monitored.
MoSCoW Priority: Should have

Quality of Service [81]
<p>INTER-LogP could ensure quality of service.</p> <p>In the INTER-IoT project there are two business use cases related to health and logistics and transport. In the case of health, if communication fails, a patient may suffer serious consequences. In the logistics case, you can assume considerable losses for a company. It is therefore essential to ensure the quality of services. In addition, safety and security could also be in risk when quality of services is not guaranteed (e.g. dangerous goods inside container boxes).</p>
<p>Acceptance criteria:</p> <p>Mechanism to ensure quality of service has to be in place (determinism, content-based prioritization, data security, bandwidth efficiency, massive scalability, real-time peer-to-peer capability, etc.).</p>
<p>MoSCoW Priority: Could have</p>

Multiple interface options [139]
<p>INTER-LogP could have multiple interface options.</p> <p>The system shall provide the users with several user responsive interfaces, according to the individual needs and preferences of a user or the situation the user is in.</p> <p>Possible types of user interface are:</p> <ul style="list-style-type: none"> - graphical user interfaces over all kind of devices (smart phones, tablet PCs, video screens, virtual reality glasses, touch screens etc.). - geographical representations of data like map representations. - audio interfaces, both for alarms/notifications and for interaction.
<p>Acceptance criteria:</p> <p>Enable automatic interface settings based upon the nature of device connected (tablet, PC, smartphone etc.) or manual settings for personalized choices made by user. Interface settings should be automatically selected upon connection, and all users should have a visible option to personalize the interface.</p>
<p>MoSCoW Priority: Could have</p>

3.4.2 Functional requirements

This section shows the functional requirements in the logistics and port transportation environment.

Provide exchange of virtual objects between platforms [194]
<p>INTER-LogP must provide exchange of virtual entities between platforms.</p> <p>INTER-IoT need to provide that a company shares a virtual object with other company when the physical object is on its facilities. You can share the whole virtual object or a part of it.</p>
<p>Acceptance criteria:</p> <p>When a truck accesses to the port, the haulier company has to provide the virtual object of the truck to the port authority IoT platform.</p>
<p>MoSCoW Priority: Must have</p>

Detection of passive physical entities to start communication with other platforms [166]
<p>INTER-LogP should detect new passive physical entities.</p> <p>When you have multiple passive physical entities, you need a mechanism for quick identification of objects that ensure you know where they are at all times. This allows to identify entities in any environment.</p>
<p>Acceptance criteria:</p> <p>A truck is identified through its license plate or a passive RFID tag.</p>
<p>MoSCoW Priority: Should have</p>

Provide services to associate and link two virtual entities [167]
<p>INTER-LogP should allow to link two virtual entities.</p> <p>INTER-IoT needs to provide services to associate and link two virtual entities handled by different and heterogeneous IoT platforms when they are in proximity. So that they can exchange information immediately between them. It is also possible disassociate the virtual entities.</p>
<p>Acceptance criteria:</p> <p>Association between truck and container so the haulier can access to all the information of a container automatically.</p>
<p>MoSCoW Priority: Should have</p>

Provide an alert system [168]
<p>INTER-LogP should provide an alert system.</p> <p>INTER-IoT needs to provide an alert system among heterogeneous IoT platforms associated with a subscription system (requirement 201) that notifies events when the attributes of a virtual entity change according to predefined values or ranges.</p>
<p>Acceptance criteria:</p> <p>Sensors warn when their battery is below 10% or down.</p> <p>Alarm when the temperature exceeds the rank.</p>
<p>MoSCoW Priority: Should have</p>

Provide the creation and monitoring of geofences [195]
<p>INTER-LogP should provide geofences functionalities.</p> <p>There are actions that must be performed when an object enters or leaves an area. Therefore, there must be a mechanism to detect it, by using geofences.</p>
<p>Acceptance criteria:</p> <p>When a truck access to the port facilities has to be detected to maintain its position monitored.</p>
<p>MoSCoW Priority: Should have</p>

IoT platforms are able to stop sharing data at any moment [252]
INTER-LogP should be able to stop sharing data at any moment. The road haulier company is able, at any moment, to finalize the connection with the port IoT platform if it decides to do that.
Acceptance criteria: If an IoT platform needs to stop sharing data with another platform, it can do it immediately.
MoSCoW Priority: Should have

Service to manage energy consumption of devices [79]
INTER-LogP could provide methods for energy management. The framework provides methods for energy management (status, enable/disable, power saving mode, etc.) to end users, when native platforms allow it.
Acceptance criteria: System to manage the energy consumption or power status of your objects.
MoSCoW Priority: Could have

Beacons to request the communication from other platforms and devices [197]
INTER-Logp could use beacons to request communication. There are some objects that need to send data to nearby devices. This communication may be indoor or outdoor and low energy consumption, as it don't have access to a power supply.
Acceptance criteria: In the port access gates are installed beacons that inform trucks of certain lane information to access the port.
MoSCoW Priority: Could have

Capacity to achieve a heterogeneous computing platform environment [198]
INTER-LogP could use heterogeneous computing facilities. In INTER-IoT there are several platforms, and each has multiple devices or sensors. This generates a lot of information that must be stored and processed. Tools that enable processing of a large amount of data coming from several different platforms, such as Big Data tools, are needed.
Acceptance criteria: Data processing of all trucks accessing to the port to extract data for detecting congestion and providing efficient traffic management.
MoSCoW Priority: Could have

Identification of an object through multiple techniques [246]
INTER-LogP could identify objects through multiple techniques.

There should be the possibility of identify an object through different techniques, giving priority to one of them.
Acceptance criteria: It is possible to identify a truck by means of an automatic reading of its RFID tag or its plate number.
MoSCoW Priority: Could have

Position detection of objects through WiFi [196]
INTER-LogP won't provide positioning through wifi although it is nice to have. The need of getting the position of the objects with accuracy and reliability becomes necessary to detect the position through different mechanisms. Therefore, the WiFi signal received could be used at different access points to calculate the position of the object, as a complement to other methods such as GPS.
Acceptance criteria: An object is detected in the terminal through the WiFi signal.
MoSCoW Priority: Won't have

3.4.3 Requirements by type

The above requirements can also be grouped according to the function they are going to perform.

Interoperability

- Detection of passive physical entities to start communication with other platforms [166]
- Provide services to associate and link two virtual entities [167]
- Provide exchange of virtual objects between platforms [194]
- Creation of new services or improvement of existing ones to access and utilize data from different platforms [248]
- IoT platforms are able to stop sharing data at any moment [252]

Communications

- High responsiveness [54]
- Beacons to request the communication from other platforms and devices [197]

Operational

- Provide the creation and monitoring of geofences [195]
- Position detection of objects through WiFi [196]
- Identification of an object through multiple techniques [246]

QoS

- Services should provide Quality of Service [81]

Functionality

- Service to manage energy consumption of devices [79]
- Priority in alarms [84]
- Provide an alert system [168]
- Capacity to achieve a heterogeneous computing platform environment [198]

Usability

- Multiple interface options [139]

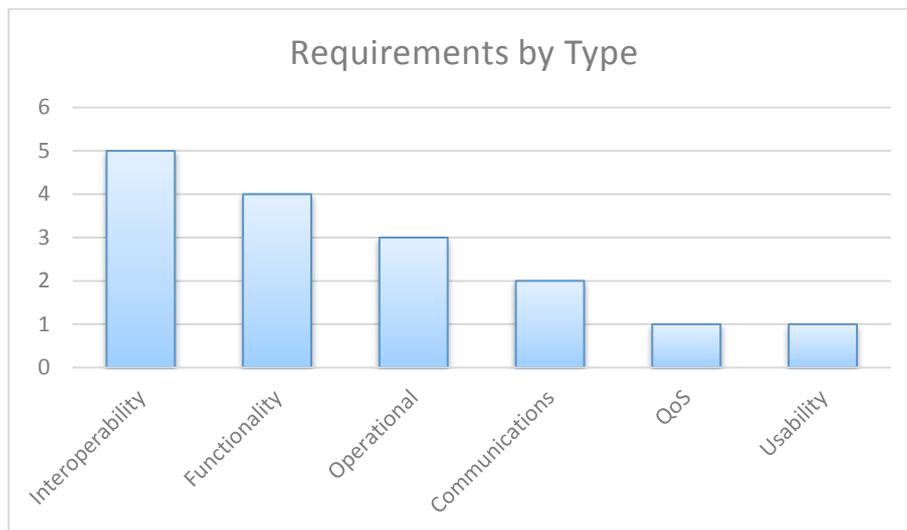


Figure 27: INTER-LogP requirements by type

The main objective of the project is the interoperability among platforms, devices and systems so most of the requirements are related with the objectives mentioned. Other requirements in INTER-LogP are mainly divided into functionality and operational, since INTER-LogP is a product focus in a business use case oriented to logistics and port transport and therefore there are specific requirements on the operation of this environment.

3.4.4 Analysis

In this subsection we analyse some of the results that can be drawn from the above requirements.

Category

Requirements can be divided by category, in functional or non-functional requirements.

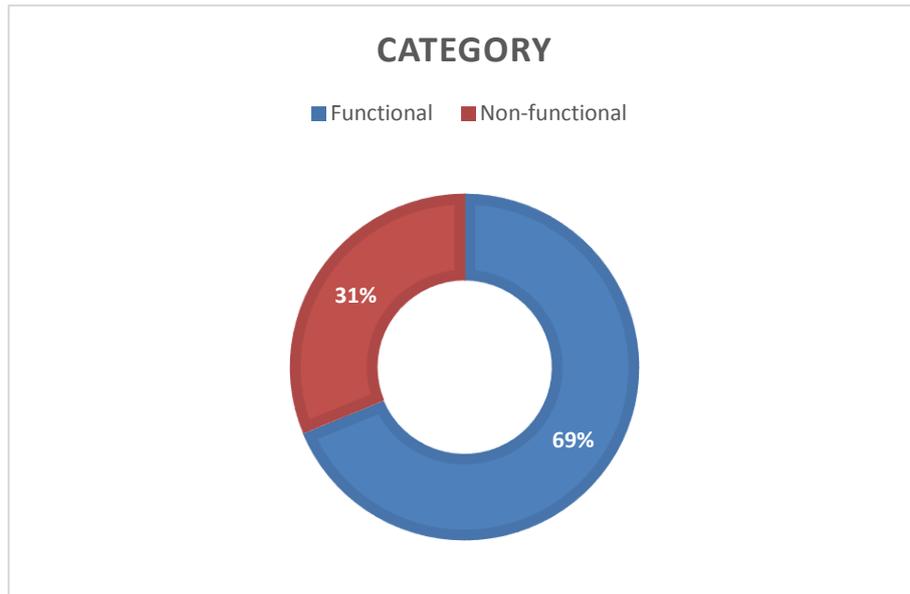


Figure 28: INTER-LogP requirements by category

As INTER-LogP is an application domain most of the requirements describe what the different platforms and devices should do. However, as the port environment has its own regulations, there are some requirements describing how the system works (non-functional).

Priority

Although all the requirements defined must be taken into account, there are some of them that are especially relevant. Others are suggestions that interest the end users and they have to be evaluated.

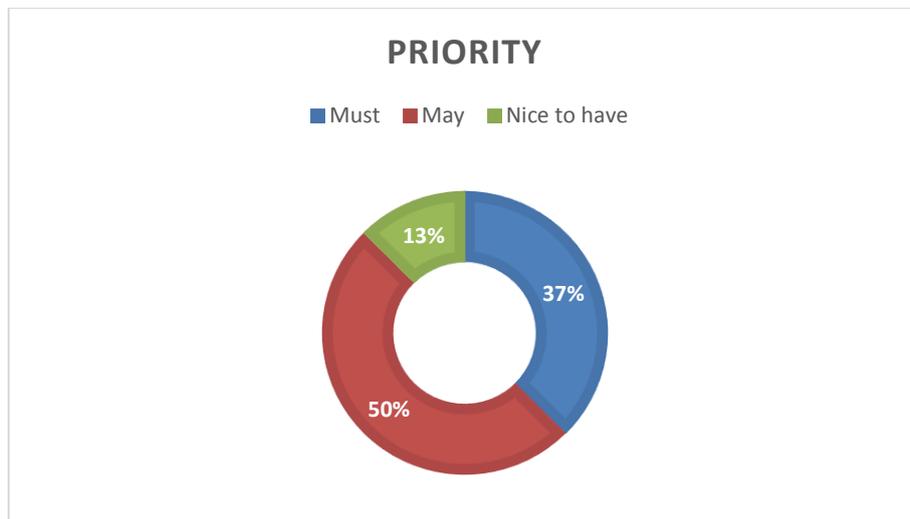


Figure 29: INTER-LogP requirements by priority

Less than half of the requirements have been considered essential and therefore must be implemented. Half of them are estimated, so we should include them. And only two are considered to be interesting to have them.

Priority by category

It is interesting to distinguish the priority between functional and non-functional requirements.

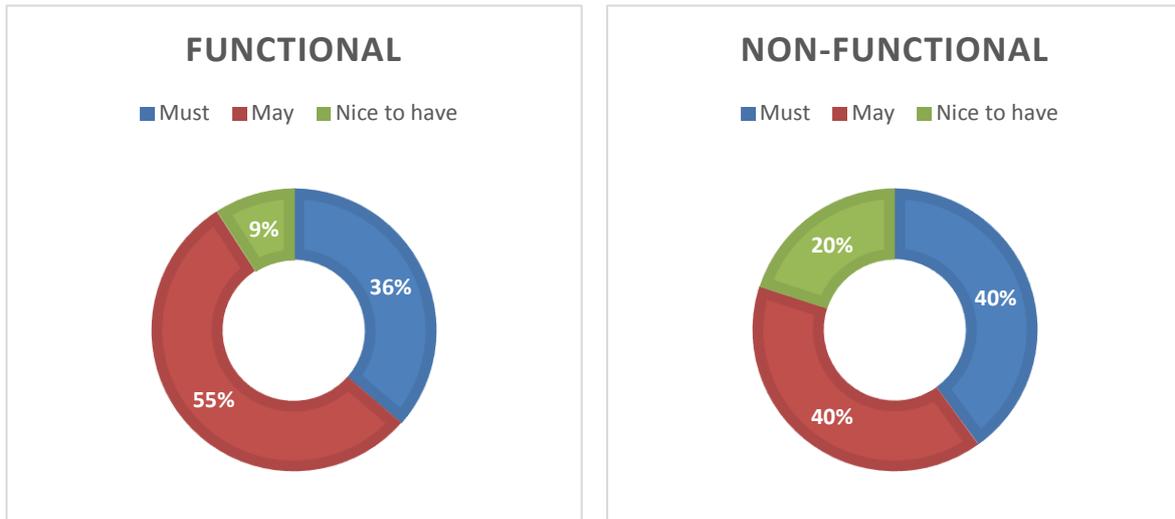


Figure 30: INTER-LogP requirements by priority and category

As you can see in the Figure 30, in the case of non-functional requirements that describe how the system works, is balanced the number of must and may requirements. But in the functional requirements, the results are similar to the whole requirements, where there are more optional requirements than important functionalities.

MoSCoW priority

The priority of the requirements is also analysed following the Moscow methodology. In the Figure 31 can be seen that most of the requirements are “Should Have” (45%), because are optional functionalities to the core of the product. Only 12% of the requirement are “Must Have” and many of them are “Could Have” (38%). There are some desirable requirements which will probably not be developed (“Won’t Have”, 6%).

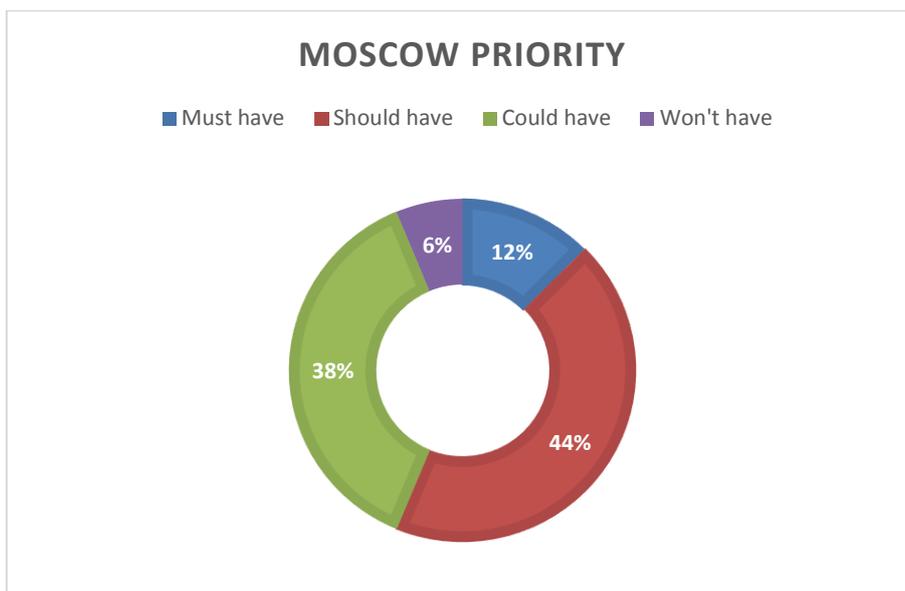


Figure 31: INTER-LogP requirements by MoSCoW priority

Sources

It is also important to highlight where the requirements come from.

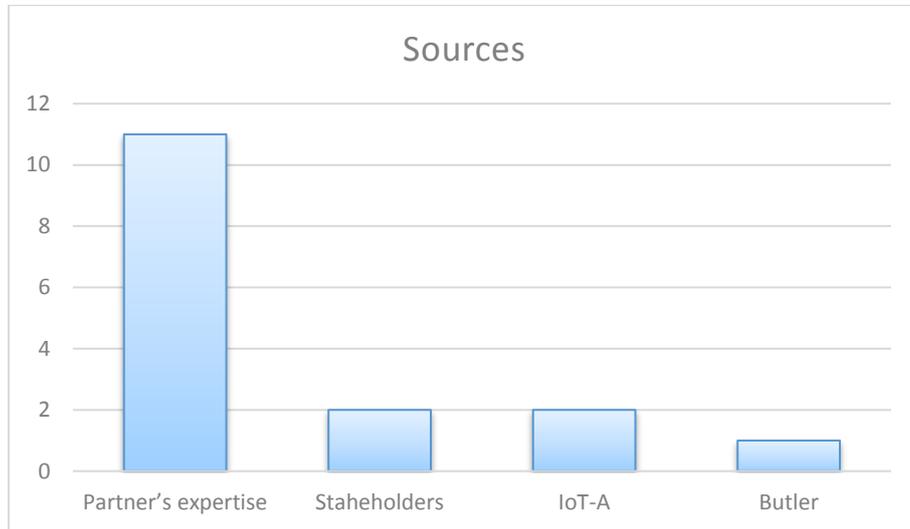


Figure 32: INTER-LogP requirements by source

Most of the requirements come from years of experience of the partners in the field of transport and port sector. There are some requirements directly suggested by stakeholders both logistics and technological companies. Finally, there are some other requirements extracted from relevant IoT projects.

3.5 INTER-Health requirements

The requirements written for INTER-Health are inspired by three different themes:

- 1) Security and privacy particularly compelling for Health Pilot (non-functional)
- 2) Performance and environmental needs (non-functional)
- 3) Architectural components and elements (functional); the standard layers for health systems are usually three:
 - a. Device
 - b. Gateway
 - c. Server
- 4) User interface, services and functionalities for health roles (patients, caregivers doctors) (functional)

Starting from detailed requirements uploaded on JIRA, a work of synthesis has been done based on observations and comments of the first draft extracted by JIRA.

Only for requirements joined two elements have been added:

- The information “including nnn” next to the requirement name, where nnn is the number of the requirement joined into the main one, for example:

‘User Access Gateway for Caregivers [177] (including 156 and 148)’

Means that requirements 156 and 148 have been merged into 177

- The information: “*Specific Instances and implementation possible solutions*” has been added in the description to include the specifications of the merged requirements.

3.5.1 Non-functional requirements

The non-functional requirements concerning the eHealth domain for the INTER-Health product are presented in this section.

<p>Constraints on processing of personal and health data [62] (including 143, 145, 146, 152)</p>

INTER-Health should conform to personal and health data processing rules.

The processing of Personal and Health data must conform to the rules and criteria laid down in “Personal Data Protection Code - Legislat. Decree no.196 of 30 June 2003” and in “Measures and arrangements applying to the controllers of processing operations performed with the help of electronic tools -27 November 2008”.

For Italian privacy regulation some specific instances are required:

Informed Consent: the person in charge of data processing must collect the informed consent of the involved person to the processing of data disclosing health status (for more details see requirement 145).

Information sheet: Art. 13 of the Privacy Codex provides that the people in charge of data processing or collecting need to be informed orally or in writing about the processing of data (for more details see requirement 146).

Privacy Codex: The behaviour of healthcare operators must be respectful of the right of personal dignity and confidentiality of every citizen and it has to be appropriate to various

situations in which benefits are provided according to the Legislative Decree 196/2003 (for more details see requirement 143).

For United Kingdom regulation some specific instances are required:

Compliance with the Data protection act: Compliance with the data protection act is required in all cases of personal data collection, usage and storage. Sensitive data (such as health data) require a higher level of protection (for more details see requirement 151).

Information Security and Information Governance good practice guidelines: The health and social care information center implements good practice guidelines for use within the NHS. Adoption of any system by NHS funded organizations requires compliance with best practice guidelines for information security and governance (for more details see requirement 152).

Acceptant criteria:

Specific functions to ensure the rules and criteria of the Nationals regulations for the Personal Data Protection (EU Regulation 2016/679 and EU Directive 2016/680).

MoSCoW Priority: Must have

Application response time [71]

INTER-Health must guarantee adequate response times.

The resulting integrated platform should guarantee the same performances of the original platforms in terms of response time.

Acceptant criteria:

The "navigation" functionalities on different contents by using either Smartphone or Personal Computer to access to the platform, have to guarantee a response time of a few seconds.

MoSCoW Priority: Must have

User Authentication to access INTER-Health services [103]

INTER-Health must provide user authentication.

Nowadays users shall authenticate to the services using their username and password (see Non Functional Requirements); if needed, a stronger way of authentication must be implemented.

To guarantee the correct identification of the INTER-Health user (such as Patients and Sanitary staff) but also the technical addicts such as platforms administrators.

Acceptant criteria:

Username and password should comply with the policies described in non-functional requirements [63].

MoSCoW Priority: Must have

Availability of sensor data [127]

INTER-Health must provide remote monitoring.

Health monitoring data must be viewable from a remote location to facilitate patient triage and inform decision making.

Acceptant criteria:

If both people in the vicinity of the patient and those viewing the patient data remotely can view the same vital signs, this requirement is fulfilled.

MoSCoW Priority: Must have

Informed Consent [145]

INTER-Health must follow informed consent practices.

The informed consent must be:

- freely express,
- specifically performance,
- must be known by the involved person, of legal age, not banned and able to understand or want. For different reasons of incapacity or inability, the privacy code has the legitimacy of one of the following subjects:
 1. operator of the power, in the case of under-age or person prohibited or subject to support administration;
 2. family, close relative or partner (all placed on the same level) for cases of physical impossibility or inability to understand or want the individual;
 3. residually, the Data Processor of the institution where is the involved subject.

Acceptant criteria:

The person tasked with processing of data must acquire the consent by signing the specific form.

MoSCoW Priority: Must have

Information sheet [146]

INTER-Health must provide an information sheet.

The person tasked with processing must provide the information sheet, to the involved person or the person from whom you collect the data, in the manner determined by the data processor of the structure and using the forms, where prepared by the Company.

During the health use case, the information sheet shows in a simple but detailed way:

- Purpose of the study;
- Detailed study protocol;
- Advantages and the risks involved;
- Any costs or compensation for subjects who choose to participate;
- Voluntary participation;
- Right of withdrawal at any time;
- References of the person responsible of the project.

Acceptant criteria:

The person tasked with processing of data must be sure that involved person read and know the information sheet.

MoSCoW Priority: Must have

Medical Device informatics [164]
INTER-Health devices should adhere to ISO/TC 215 standard. ISO/TC 215 Health informatics sets international standards for medical data transfer.
Acceptant criteria: Devices to be included in INTER-IoT most likely adhere to this standard. Annual review of this standard should insure compliance.
MoSCoW Priority: Should have

Easy-to-use user interface [70]
INTER-Health could provide different interface features for different category of end users. All end user interfaces should be easy to use. For older people the confidence with the use of smart phones, Internet applications and information technology in general is very limited. For younger people the confidence on the use of smartphones Internet technologies and applications increases. As services are offered to both types of users, this requirement needs to be considered as a reference target for the design of web and mobile interfaces It is assumed that health care providers (doctors, nurses, technicians) have experience with Internet applications. Access to the management, monitoring and consultation could be done through a web interface.
Acceptant criteria: All the offered features should be easy to use by each category of end users.
MoSCoW Priority: Could have

3.5.2 Functional requirements

In this section are presented the functional requirements concerning the eHealth domain for the INTER-Health product.

Definition of reference meaning for health information [106]
INTER-Health must define a reference for health information. Health information can be detected using different devices according to different way of measurement (unit of measure that could differ from country to country and, also, depending on devices manufacturers). To use same information coming from different systems and going to others, it is mandatory to establish specific criteria to: 1. Define a common meaning if it is possible. 2. Determine a correspondence between different data that have the same meaning and different values. 3. Set transcoding tables between different values of the same data.

Acceptant criteria:

Presence of well-defined criteria to guide the correct data interpretation.

MoSCoW Priority: Must have

User Access Gateway for Patients [176]

INTER-Health patients must have an access gateway.

Gateway main functionalities for patients are:

- Access to services (providing username and password).
- Setting Profile communication and devices pairing.
- Managing measures on the device and releasing them to the gateway which stores them on a local database.
- Possibilities of inserting measures manually.
- Sending measures to the platform.
- Reporting locally measures already stored.

In term of interoperability, the INTER-Health gateway uses the gateway architectural scenarios described in the requirement 175.

Acceptant criteria:

Use a gateway for gathering all wireless medical devices.

MoSCoW Priority: Must have

Personal data collected on Computerized Nutritional Folder [218]

INTER-Health must collect personal data.

The data of the recruited subjects, collected on electronic nutritional folder refer to:

- Personal and identification data.
- Anthropometric data (weight, height, BMI, waist circumference).
- Food research (eating habits and physical activity).

Acceptant criteria:

Respect of the policies described in non-functional requirements [143, 145].

MoSCoW Priority: Must have

Exchanging discrete medical measures across platforms [101]

INTER-Health should exchange medical measures across platforms.

A discrete health measure must be accessed and used in many platforms, also different from the one that first physically picked up the information. Semantic Analysis about the meaning of the data is needed.

Acceptant criteria:

The measurement is accessed from different platforms.

MoSCoW Priority: Should have

Exchanging complex medical measures across platforms [102]
<p>INTER-Health should exchange complex medical measures across platforms.</p> <p>A complex health measure (i.e. a file made of different parameters covering a period of time) can be used in many platforms different from the one that first physically picked up the information. Data usage and elaboration must be done in accordance with the protocol used to store the information.</p>
<p>Acceptant criteria:</p> <p>No value changes in different systems.</p>
<p>MoSCoW Priority: Should have</p>

Personal data and user profile management [104]
<p>INTER-Health should manage personal data and user profile.</p> <p>User data and provisioning is based on:</p> <ol style="list-style-type: none"> 1. A set of identification data (such as surname, name, tax code, country and so on). 2. Role and profiling. 3. Contact data and addresses. 4. Anthropometric and health information. <p>Users can be recorded both locally (on the owner platform) or in the Cloud (on one or more client platforms).</p>
<p>Acceptant criteria:</p> <p>Personal data should comply with the policies described in non-functional requirements [61].</p>
<p>MoSCoW Priority: Should have</p>

Exchanging synthetic or statistical health information between platforms [107]
<p>INTER-Health should exchange synthetic or statistical health information.</p> <p>Events, Dashboards, Images, Reports, Graphs and Charts should be exchanged or executed independently of the owner platforms.</p> <p>Different information is produced at different levels all over different platforms interoperable in IoT Galaxy; the purpose of this requirements is to use synthetic or almost worked data where they are reusing the results without reworking them totally. Many configurations are possible.</p> <ol style="list-style-type: none"> 1. To use a platform (master) as main point of access, linking the other functions accessed through APIs to the platforms owner where the functions are executed. 2. To use a distributed approach calling different APIs with many interfaces between different platforms at the same level. 3. To use a Business Intelligence platform for synthesis, elaboration, statistics and presentation, keeping operational analytical data in the owner platforms. <p>In case 1 e 2 dashboards and reports, produced by the owner platforms are exchanged without adding elaboration; in case 3 some algorithms could be written and executed directly on the Business Intelligence platform.</p>

Acceptant criteria:

Relevant information such as reports, graphs or charts is exchanged among platforms.

MoSCoW Priority: Should have

Timestamped event recording [154]

INTER-Health should handle time stamped events.

To highlight specific events within a patient history, a time stamped generic event could be generated. The time stamp associated with this event should be to the accuracy of seconds.

The ability for Inter-IoT to handle time stamped events with associated meta data should be supported.

Acceptant criteria:

The ability to handle time-stamped electronic data from a device or application for use in other applications.

MoSCoW Priority: Should have

Seamless patient monitors [157]

INTER-Health should allow the remote monitoring of physiological data of patients.

Any obstacle to the treatment of a patient must be eliminated, if possible without compromising the care of the patient. The inability for medical professionals using medical applications to access specific patient data taken externally, can affect care.

Acceptant criteria:

All physiological data (e.g. from ECG, Blood Pressure and SpO2 monitors) should be accessible by specific applications connected to the Inter-IoT.

MoSCoW Priority: Should have

Bioethic Committee [158]

INTER-Health should have an approval of the ethics committee.

The clinical trial is an extraordinary means to evaluate the efficacy of a drug or a medical device, the risks involved and, ultimately, to decide whether it is appropriate to make it available for the population.

However the importance of research can never justify the violation of the rights of persons participating in the trial. For this reason the European Union has adopted a set of rules, called Good Clinical Practice, which govern the research correct.

To ensure the observance of "Good Clinical Practice", they have been set up specific bodies (Ethics Committees) who evaluate and closely monitor each trial overseeing the correctness and evolution over time.

Ethics Committees are independent bodies formed by health operators and not, with the task of evaluating the protocols of each trial from the point view of the scientific, ethical and feasibility.

In Italy it is not possible to conduct any human trials without first this has been assessed and approved by an ethics Committee. Actually there are more than 200 distributed among hospitals and local health authorities.

Other tasks of ethics Committees are:

- monitor the progress of the studies;
- promote information and training for doctors and patients;
- provide opinion and directions in the case of specific requirements, both individually (in case of uncertainties concerning the treatment to be applied), both at a general level (e.g. in case you need to make decisions related to patient groups);
- check the economic coverage for the costs of the trial;
- check that the protocol of the trial must be guaranteed the right to dissemination and publication of the results by the investigators regardless of the opinion of the promoter and in compliance with applicable laws regarding the processing of sensitive data and intellectual protection confidentiality.

Acceptant criteria:

Expression of consensus for the trial.

MoSCoW Priority: Should have

User Access Service for Patients [172]

INTER-Health should provide access to patients.

User health main functionalities for patients are:

- Access to services (providing username and password).
- Personal settings (contacts, measurements reminders and so on), managed in registration and updated by patients, later.
- Reporting: access to measures by chronological reports or using graphics and dashboards.

Each group of user functionalities can be implemented in different ways in accordance to the choices done in the architectural scenario.

For the patients' functionalities and health services specific features is worth what said for user access services for doctors apart from 'Personal data collected on Computerized Nutritional Folder'.

Acceptant criteria:

Presence of a specific view for patients' functionalities.

MoSCoW Priority: Should have

User Access Service for Doctors [173] (including: 208, 209, 210, 211, 212, 213, 217, 218, 155, 150, 149)

INTER-Health should provide access to doctors.

User health main functionalities for doctors are:

- Access to services (providing username and password).
- Personal information (contacts, receiving alerts) managed by administrators in registration and eventually updated by doctors.

<ul style="list-style-type: none"> • Assigned Patients medical parameters settings (measurements schedules, thresholds). • Medical report management (special reporting allowing specialists to follow report workflow). • General purpose (chronological) or ad-hoc (oximetry, images and so on) reports.
<p>Acceptant criteria: Presence of a specific view for doctors' functionalities.</p>
<p>MoSCoW Priority: Should have</p>

<p>User Access Service for Administrators [174]</p>
<p>INTER-Health should provide access to administrators. Administrators main duties in INTER-Health are:</p> <ol style="list-style-type: none"> 1. User provisioning. 2. Platform authorization and activation of APIs. For the first process, in accordance to the architectural choices, different options are possible. <ul style="list-style-type: none"> • Keeping the main directory to register users, on the "A" platform (that acts as master, the only which can update, through platform provisioning on "A" portal, the database) and then giving access to data, via API to the interested platforms (only query). • Keeping users on "A" platform (master Director) allowing other platforms to insert update and query users by APIs. • Keeping the "A" user database as master and synchronizing the slave directories in other platforms (redundant way). • Use third parties Directory as master and keep it synchronized with all platforms interested. <p>For the second process the main functionalities for administrators are:</p> <ul style="list-style-type: none"> • Define platforms and systems that may interact (client platforms, suppliers platforms). • Set protocols and standards for exchanging data. • Set APIs authorization and so on. • Activate/deactivate the use of interface and so on.
<p>Acceptant criteria: Presence of a specific view for the administrator functionalities.</p>
<p>MoSCoW Priority: Should have</p>

<p>User Access Gateway for Caregivers [177] (including 156 and 148)</p>
<p>INTER-Health caregivers should have an access gateway. INTER-Health gateway for Caregivers is intended to integrate gateways functionalities for assisted measurements (i.e. measures that patients are not able to do or it's better to take with the aid of an expert person). Architectural options are the same of gateways for patients. Caregivers gateway main functionalities in addition to what already described for patients, are:</p> <ul style="list-style-type: none"> • Authentication as assistant. • Authorization to manage data of different users. • Choice of patients between a set of cared users.

<ul style="list-style-type: none"> • Possibilities of inserting measures manually. • Setting Profile communication and devices pairing. • Managing measures on the device and releasing them to the gateway that stores them on a local database. • Sending measures to the platform. • Reporting locally measures already stored. <p>In term of interoperability, the INTER-Health gateway uses the gateway architectural scenarios described in the requirement 175.</p> <p>Specific Instances and implementation possible solutions. In particular situation, for example assisting patients on an ambulance, all text boxes must be 'speech to text' able.</p> <p>Besides, sensor data must be immediately available on gateway display unit to allow triage to continue in an uninterrupted path. Please see requirement 153 for how to address drops in server connectivity.</p>
<p>Acceptant criteria: Presence of a specific view for the Caregivers functionalities.</p>
<p>MoSCoW Priority: Should have</p>

<p>Wearable devices support [217]</p>
<p>INTER-Health could support wearable devices.</p> <p>Wearable Mobile Devices used are equipped with wireless interface and are CE labelled in accordance with Directive 93/42/EEC, which certifies that the device meets the minimum essential requirements of safety of operators and citizen.</p> <p>The detection of the physical activity practice occurs daily and in mobility.</p> <p>The surveys recorded and sent in INTER-Health platform, are available from both health staff and the subject.</p> <p>The information gathered from wearable mobile devices allows to capture real-time status of physical activity of the subject and check the achievement of objectives. If necessary health staff requires additional counselling for subjects who have particular risk situations.</p> <p>Wearable mobile devices detect Routes Steps Number, calories burned and Minutes of physical activity carried out.</p> <p>Information gathered from wearable mobile devices is used to divide patients into different categories (e.g. Inactive or Sedentary Subjects, Moderately active Subjects, Sedentary Subject, etc.).</p>
<p>Acceptant criteria: Wearable devices are used is the scenarios.</p>
<p>MoSCoW Priority: Should have</p>

3.5.3 Requirements by type

The INTER-Health requirements can be grouped according to the function they are going to perform.

Interoperability

- Exchanging discrete medical measures across platforms [101]
- Exchanging complex medical measures across platforms [102]
- Exchanging synthetic or statistical health information between platforms [107]

Legality

- Constraints on processing of personal and health data [62]

Operational

- Timestamped event recording [154]
- Seamless patient monitors [157]
- Bioethic Committee [158]
- User Access Service for Patients [172]
- User Access Service for Doctors [173]
- User Access Service for Administrators [174]
- User Access Gateway for Patients [176]
- User Access Gateway for Caregivers [177]
- Wearable devices support [217]

Privacy

- Informed Consent [145]
- Information sheet [146]
- Personal data collected on Computerized Nutritional Folder [218]

Security

- User Authentication to access INTER-Health services [103]

Functionality

- Application response time [71]
- Availability of sensor data [127]
- Medical Device informatics [164]

Semantics

- Definition of reference meaning for health information [106]

Data model

- Personal data and user profile management [104]

Usability

- User interface [70]

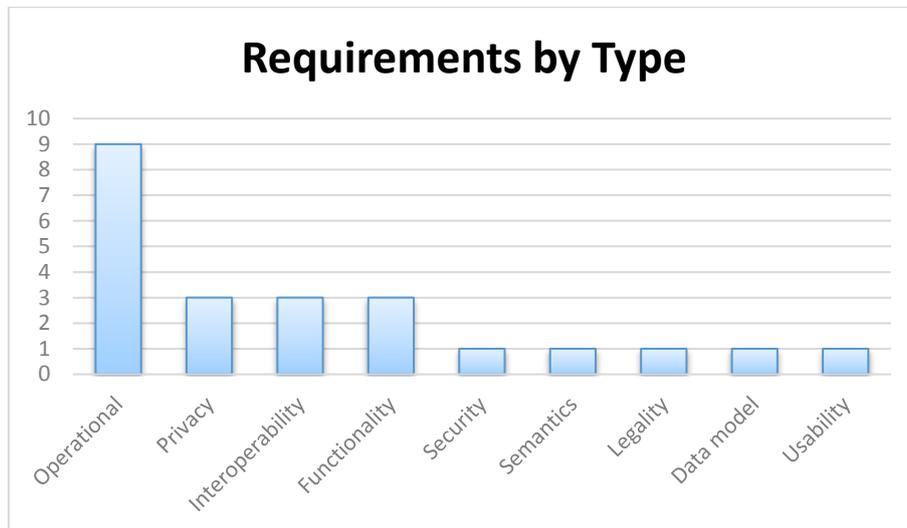


Figure 33: INTER-Health requirements by type

Since the INTER-Health product is an application domain, the more numerous requirements are the operational. There are also requirements specific about interoperability and functionality requirements because the goal is to be able to easily interoperate with other domain products. There is also a need of processing personal data, so there are requirements related to security and privacy aspects necessary to ensure that the legal constraints are correctly handled.

3.5.4 Analysis

In the following section we describe how the INTER-Health requirements could be analysed on the basis of the aggregation by categories, priority and source.

Category

The requirements are categorized in functional and non-functional requirements.

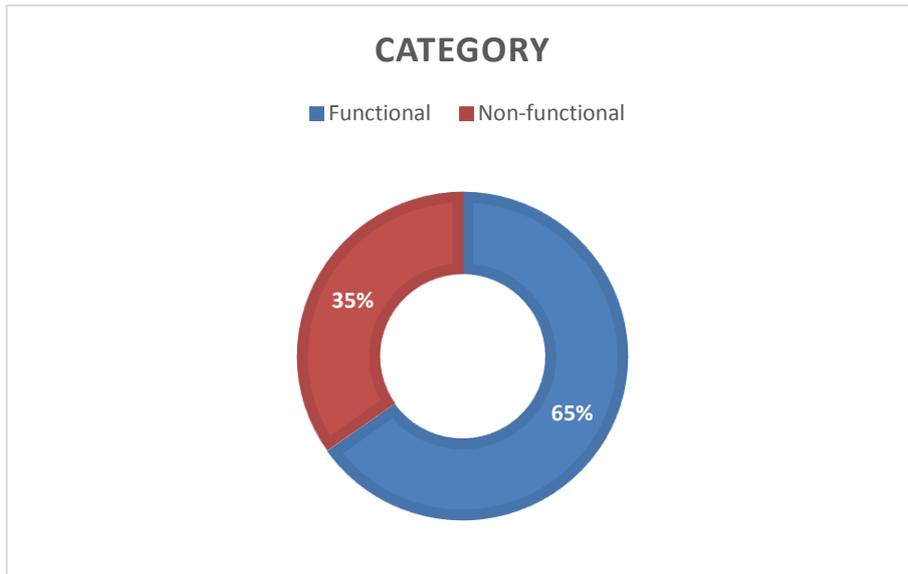


Figure 34: INTER- Health requirements by category

For the INTER-Health product have been identified 8 non-functional and 15 functional requirements. The main part of non-functional requirements refers to product functionalities, privacy and security aspects. The main part of functional requirements refers to operation, interoperability and data model aspects.

Priority

The requirements have been categorized on the priority basis depending on the level of relevance: must, may and nice to have.

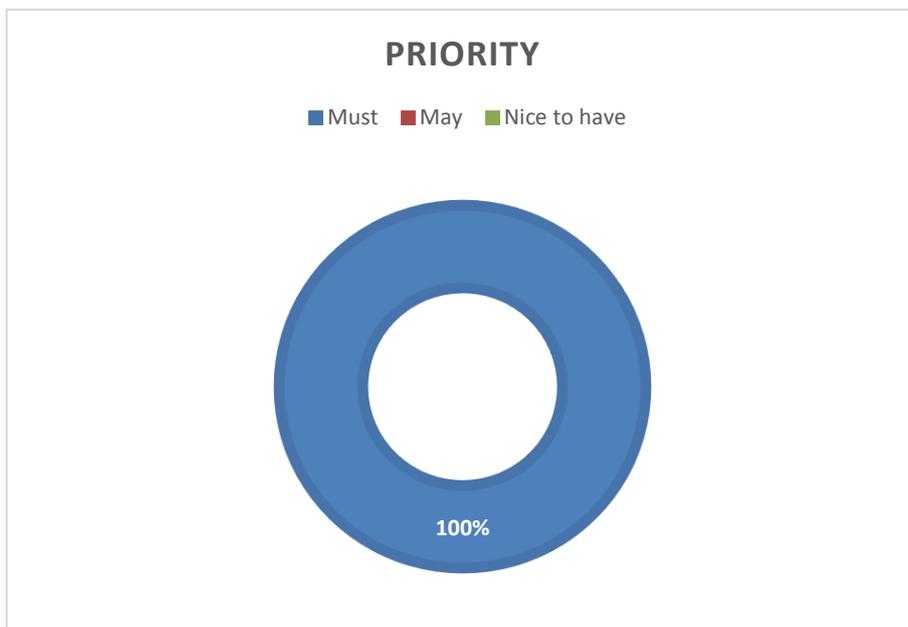


Figure 35: INTER- Health requirements by priority

All the requirements identified must be present in the INTER-Health product. The reason is probably due to the aggregation work that has been done on the requirements focusing on the more relevant ones.

MoSCoW priority

In the Figure 36 Figure 13is described the priority of the requirements following the Moscow methodology. In this case, around 40% of them are “Must Have”, which are related with security and privacy. More than half of the requirements have been rated as “Should Have” and only around 5% are “Could Have”.

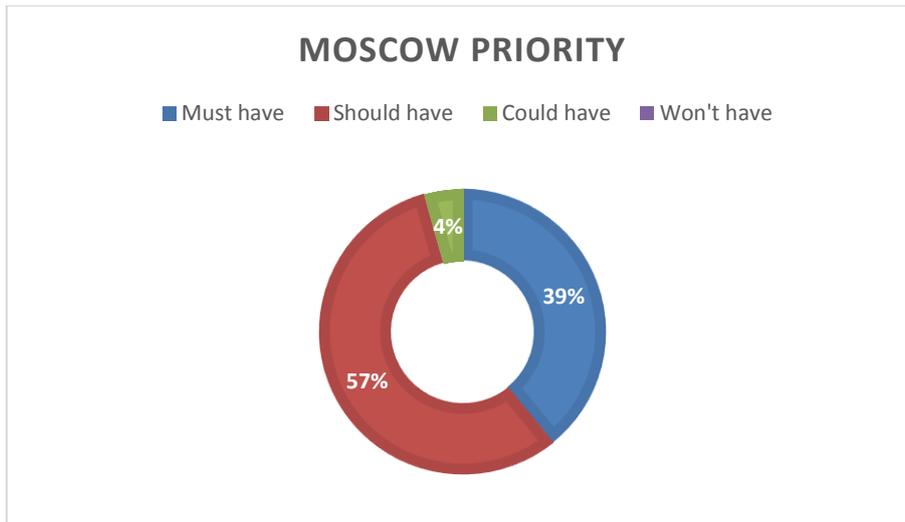


Figure 36: INTER-Health requirements by MoSCoW priority

Sources

The following table shows in which way the requirements have been identified.

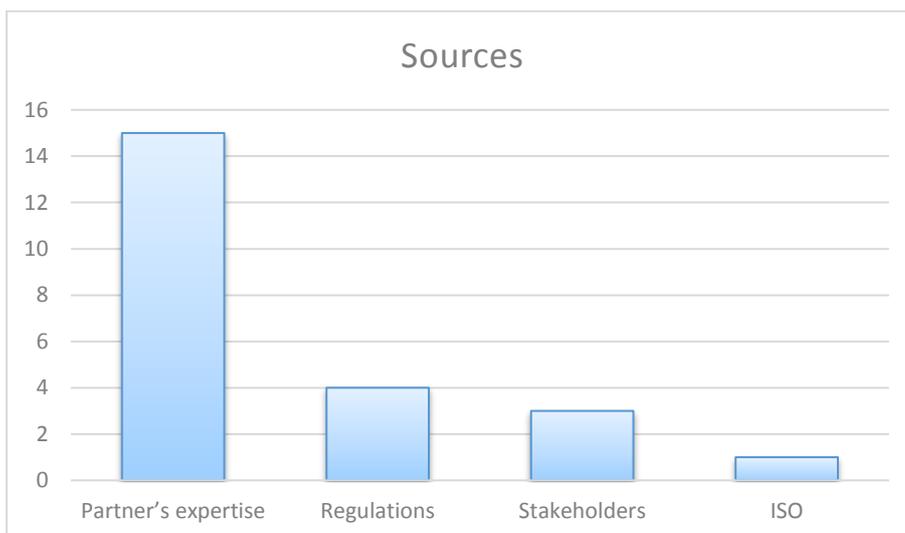


Figure 37: INTER- Health requirements by source

The main source that guided the drafting of the requirements is based on the partners experience on the e-Health domain. Also, regulations and stakeholders constraints are relevant sources of information.

3.6 General requirements

There are some requirements that do not affect only one of the products proposed in INTER-IoT, but they affect the entire project. These requirements are related to data protection, some security issues or being environmentally friendly.

3.6.1 Non-functional requirements

In this section are presented the non-functional requirements related to all products supported by the INTER-IoT project.

Platform independency [14]
<p>INTER-IoT must be platform independent.</p> <p>The INTER-IoT products must be able to connect or communicate to any platform by means of a common procedure which can be supported by any platform.</p>
<p>Acceptance criteria:</p> <p>At least Linux, Windows, IOs and Android must run as platforms or devices.</p>
<p>MoSCoW Priority: Must have</p>

System security [27]
<p>INTER-IoT must have system security.</p> <p>Certain sections or sensors/actuators must be securely shielded from the public.</p> <p>It is obtained by relevant control functions such as access and transport resource control functions: authentication, authorization and accounting (AAA).</p>
<p>Acceptance criteria:</p> <p>Devices cannot be accessible by everyone. Access must be controllable for sensors. The network layer must provide reliable and secure connectivity as required by the pilots.</p> <p>Secured environments can be created for designated users, any sensitive data communication must be secure.</p> <p>Security levels do not allow third parties to take over control of a private system that is working over the IoT.</p>
<p>MoSCoW Priority: Must have</p>

System privacy [28]
<p>INTER-IoT must provide privacy protection.</p> <p>Provide privacy protection for accessing information about physical entities, services or platforms connected to or integrated into INTER-IoT.</p> <p>To maintain this privacy, third party access to the private data or into the system is not possible.</p> <p>Also, the identifier or other critical information of a device (e.g., ID of an RFID tag or MAC address of Wireless Sensor) must not be tracked by unauthorized entities.</p> <p>Additionally, the avoidance of the integration and interaction of false nodes/sensors, or unauthorized smart objects must be ensured.</p> <p>Optional cryptographic mechanisms for information could be conveniently used.</p>

Acceptance criteria:

INTER-IoT provides higher security in device integration and system-device interaction. Security levels do not allow third parties to take over control of a private system that is working over the IoT.

The IDs are only sent (and maybe stored) to/in other authorized entities, typically in the same subsystem, without any tracking purposes.

MoSCoW Priority: Must have

Definable and monitored requirements [41]

INTER-IoT must provide definable and monitored requirements.

To meet the project objectives should specify clear and complete requirements from the start. The requirements should be monitored and adapted to the needs of users at all times.

Acceptance criteria:

The requirements can be monitored and validated.

MoSCoW Priority: Must have

Constraints on processing of personal data [61]

INTER-IoT must comply with the constraints to process personal data.

In the case of processing of personal data, the processing must conform to the rules and criteria laid down in Directives 95/46/EC and 2002/58/EC and National regulations. These conditions fall into three categories:

- Transparency (the natural person has the right to be informed when his personal data is being processed). What we are required to provide, the purpose for which data is treated, all the information that characterizes the treatment of his data and the rights they enjoy and need to collect its approval in this regard.
- Legitimate purpose (personal data can only be processed for specified explicit and legitimate purposes and may not be processed further in a way incompatible with those purposes).
- Proportionality (personal data may be processed only insofar as it is adequate, relevant and not excessive in relation to the purposes for which they are collected and/or further processed; when sensitive personal data such as religious beliefs, political opinions, health, sexual orientation, race, membership of past organisations are being processed, extra restrictions apply).

Acceptance criteria:

Presence of requirements and functions to ensure the rules and criteria of EC and National regulations.

MoSCoW Priority: Must have

Communication channel security [65]

INTER-IoT must ensure security and integrity of data.

Must be foreseen the adoption of hardware equipment and software (e.g. Firewall) to counter attempts of unauthorized access in order to meet the levels of insulation and protection of

<p>personal data processed by the same platform (e.g. virtual machines level, physical machines level, network level, storage and management networks level).</p> <p>Must be implemented secure protocols of communication / transmission (e.g. SSL, SSH) of the health data for all connections including machine-to-machine.</p>
<p>Acceptance criteria:</p> <p>Use a communication protocol that ensures the security and integrity of data.</p>
<p>MoSCoW Priority: Must have</p>

<p>Compliance with the Data protection act (UK) [151]</p>
<p>INTER-IoT must be compliant with the data protection act in UK.</p> <ol style="list-style-type: none"> 1. Personal data shall be processed fairly and lawfully and, in particular, shall not be processed unless – <ol style="list-style-type: none"> (a) at least one of the conditions in Schedule 2 is met, and (b) in the case of sensitive personal data, at least one of the conditions in Schedule 3 is also met. 2. Personal data shall be obtained only for one or more specified and lawful purposes, and shall not be further processed in any manner incompatible with that purpose or those purposes. 3. Personal data shall be adequate, relevant and not excessive in relation to the purpose or purposes for which they are processed. 4. Personal data shall be accurate and, where necessary, kept up to date. 5. Personal data processed for any purpose or purposes shall not be kept for longer than is necessary for that purpose or those purposes. 6. Personal data shall be processed in accordance with the rights of data subjects under this Act. 7. Appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data. 8. Personal data shall not be transferred to a country or territory outside the European Economic Area unless that country or territory ensures an adequate level of protection for the rights and freedoms of data subjects in relation to the processing of personal data.
<p>Acceptance criteria:</p> <p>All systems must be compliant with the Data protection act.</p>
<p>MoSCoW Priority: Must have</p>

<p>Integration with legacy systems [188]</p>
<p>INTER-IoT must integrate with legacy systems.</p> <p>Interfaces toward existing systems must be developed. In order to allow interaction between new and old developments. It is necessary connect legacy systems with new services through standard based protocol gateways to free data from proprietary constraints.</p>

Acceptant criteria:

When legacy systems older than 2010 are used, these systems must be able to communicate with the new INTER-IoT gateways and the rest of the system.

MoSCoW Priority: Must have

IDEs and APIs for rapid new applications development [199]

INTER-IoT must be supported by APIs and IDEs.

The port environment is constantly changing and therefore the need to create new applications. The products created must have APIs and IDEs that allow easily create new applications. They also have to be well documented to facilitate an understanding.

Acceptance criteria:

Have simple APIs and IDEs to develop services with complete documentation.

MoSCoW Priority: Must have

Extensibility of the use cases [122]

INTER-IoT architecture should be extensible.

Architecture must allow for an efficient extension of existing use cases and for a simplified definition of new use cases within the defined functional scope.

Acceptance criteria:

Use of JIRA to create and maintain all the uses cases during the project.

MoSCoW Priority: Should have

Use of standards [123]

INTER-IoT should make use of standards.

Solution should make use of established standards wherever suitable. The use of existing standards should be envisioned. The extension/adaptation of existing standards should be preferred for the definition of a proprietary solution.

Acceptance criteria:

Existing standards are used when possible.

MoSCoW Priority: Should have

Trust management [100]

INTER-IoT could support trust issues.

Trust issues are related to the scenario in which devices and platforms cooperate without previous collaboration history. Trust management enables to make sure that the shared data are real and trustworthy, especially with crowdsourced and user generated data.

Acceptance criteria:

Solution proposes mechanism to achieve trust management.

MoSCoW Priority: Could have

Adaptability [125]

INTER-IoT could be adaptable to changing requirements.
--

INTER-IoT must be adaptable to changing requirements on availability, performance and throughput.

It must be suited to be continuously improved with respect to changing user behaviour. Usage processes, service usage patterns and user feedback should be tracked continuously in order to detect architectural weaknesses, inappropriate design decisions and inappropriate technologies.

Acceptance criteria:

INTER-IoT provides data for the continuous evaluation of usage patterns and their impact on the running solution.

MoSCoW Priority: Could have

Environmental protection [128]

INTER-IoT could use environmental friendly products.
--

The products must be able to select the most environmentally friendly option, if any, given a choice and provided that metadata about environmental characteristics are available (CO2 impact, for instance, or else energy consumption).

Acceptance criteria:

The products are environmentally friendly.
--

MoSCoW Priority: Could have

3.6.2 Requirements by type

The above requirements can also be grouped according to the function they are going to perform.

Interoperability

- Platform independency [14]
- Use of standards [123]
- Integration with legacy systems [188]

Security

- System security [27]
- System privacy [28]
- Communication channel security [65]
- Trust management [100]

Privacy

- Compliance with the Data protection act (UK) [151]

Functionality

- Definable and monitored requirements [41]
- Environmental protection [128]

Architecture

- Extensibility of the use cases [122]
- Adaptability [125]

Legality

- Constraints on processing of personal data [61]

API

- IDEs and APIs for rapid new applications development [199]

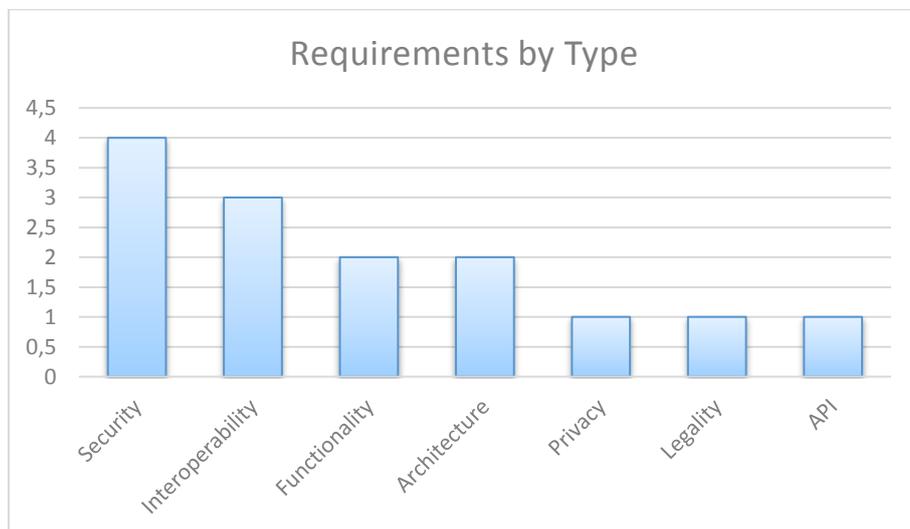


Figure 38: General requirements by type

The requirements are fairly distributed among the different categories. Can be highlighted the requirements about security, interoperability or functionality.

3.6.3 Analysis

In this section we analyse some of the results that can be drawn from the above general requirements.

Category

Requirements can be divided by category, in functional or non-functional requirements.

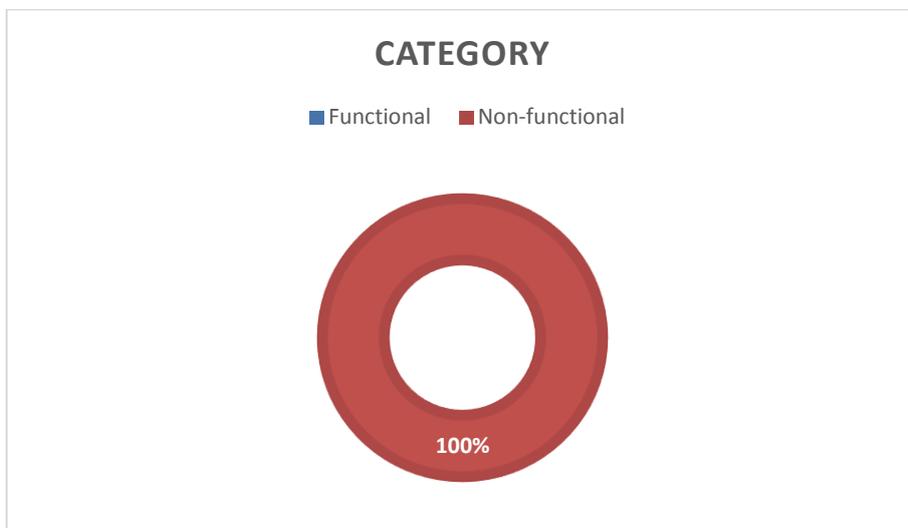


Figure 39: General requirements by category

The requirements in this section are general for all products, so all of them describe how the system should work or which features should have. That is the reason that all requirements are non-functional.

Priority

Although all the requirements defined must be taken into account, there are some of them that are especially relevant. Others are suggestions that interest the end users and they have to be evaluated.

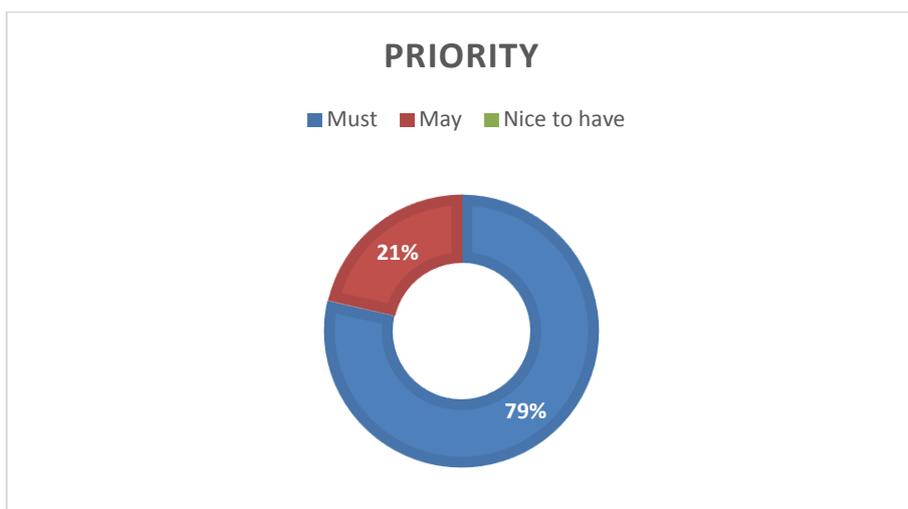


Figure 40: General requirements by priority

As you can see in the Figure 40, more than three quarters of the general requirements are mandatory, so we have to take it into account in the design of the different products. The reason is because they are relevant requirements affecting the entire system.

MoSCoW priority

The priority of the requirements is also analysed following the Moscow methodology. In the Figure 41 can be seen that most of the requirements are Must (64%), because are general functionalities that the system must comply. The rest of the requirements are Should (15%) and Could (22%).

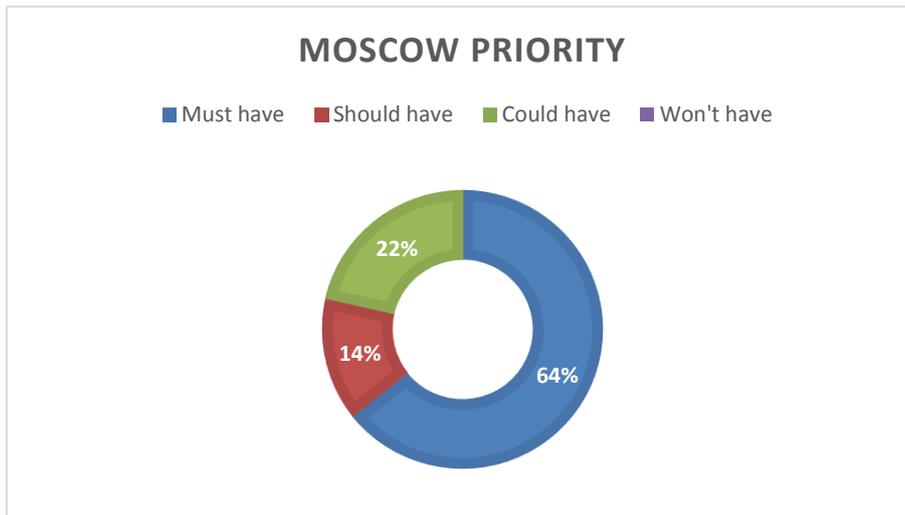


Figure 41: General requirements by MoSCoW priority

Sources

The requirements can also be classified by where they have been extracted.

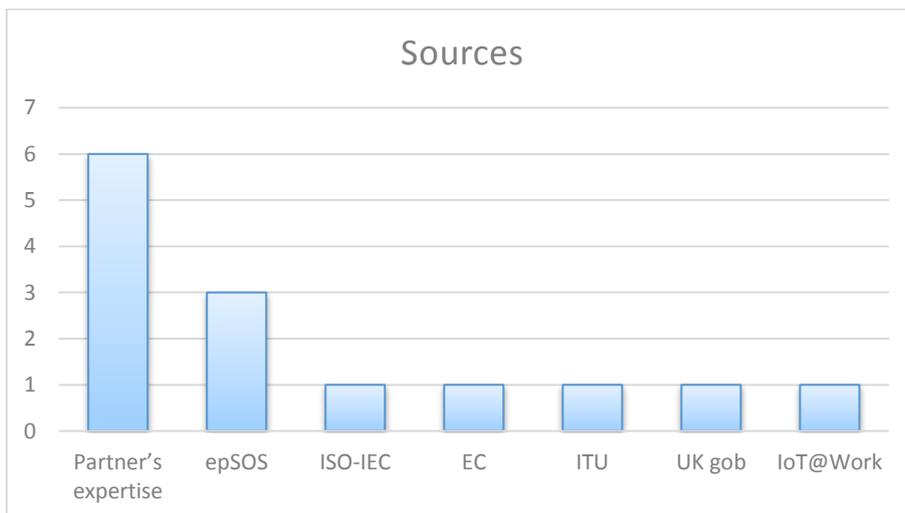


Figure 42: General requirements by source

For the general requirements, most of them come from the extensive experience of the partners and the results of some European projects like epSOS. There are some requirements from international standardization organizations or European and national governments.

4 Conclusions

For any product, design, or project is necessary to begin by establishing what we want to achieve. For that, we must perform a thorough analysis of what is on the market and what our customers may need. With this information we have to specify in detail the features and functionality of our product. This is the aim of the definition of requirements.

The requirements are used to establish the basis for agreement between the customers and the suppliers on what the software product is intended to do. Knowing the needs of customers is easier to develop a successful business model.

The requirements are the basis for the design stage, so a well-defined requirement reduces the development effort. During the specification of the requirements, we should involve almost all the departments of the organization in order to define the necessary requirements for a specific product or service. A complete and correct requirement process reduces the effort wasted on redesign, recoding and retesting. It also provides an efficient mechanism for the product validation and verification.

In order to define detailed requirements, we need to decompose the product into much smaller chunks. This helps to break down the problem into its component parts and makes it easier to establish the solution. It also provides a basis for estimating costs and schedules.

The first stage of the INTER-IoT project consisted in several interviews with stakeholders in order to obtain their needs. We also did a thorough analysis of existing products on the market. The results are in the deliverable D2.1. With all this information, the knowledge of the partners and some regulations and standards we began to identify the requirements.

At this time the project has about 185 requirements. However, since the requirements are an iterative process that takes place throughout the duration of the project, they may be improved or added more if needed. So that all partners have access to the latest version of the requirements, we are using JIRA. We use this tool to improve the requirements when the other work packages will start. As we are using an agile and iterative methodology, we can add new requirements without interfering with what has already been defined.

From the results obtained, we can note that most of the requirements are non-functional. This may be because the project does not attempt to develop a product or platform, but a framework for interoperability between platforms. Therefore, many of the requirements describe system characteristics and features that should be provided. We have also two application domains (INTER-LogP and INTER-Health) where there are more functional requirements, since in this case we try to develop a more specific product.

As this is a framework for the interoperability of platforms, there are a large number of requirements about security, privacy and interoperability. In some products there are types of requirements more numerous such as communications in INTER-LAYER, API in INTER-FW, methodology in INTER-METH, or operational in the application domains. There are also other relevant types such as semantics, architecture, usability, etc.

Concerning the requirements priority, there are two kind of classification. First it is classification from the needs of the stakeholders. According to this, approximately two thirds are mandatory. Otherwise, the requirements are classified depending on the priority to be developed. In this case around 40% are priority. This is because at first we have identified the essential features and functionalities that,

the different products, must have. Furthermore, we have also defined other requirements that might be interesting to have.

The main sources of data we have taken into account when defining the requirements have been the stakeholders' needs and the partner's expertise. Nevertheless, it has also been quite important to consider other sources such as IoT associations and projects (IOT-A, AIOTI, etc.), standardization organizations recommendations (IEEE, ITU, ISO, etc.) and national and European regulations.

From now on, the defined requirements and those that may arise, are used to define the use cases in task T2.4. Furthermore, legal and regulatory requirements are extended in task T2.5. Moreover, we are going to start the design of the INTER-LAYER and INTER-FW products in work packages 3 and 4.