(FP7 614100)

D2.1.1 Initial Requirement Report

31 January 2014 – Version 1.0

Published by the IMPReSS Consortium

Dissemination Level: Public

Project co-funded by the European Commission within the 7th Framework Programme and the Conselho Nacional de Desenvolvimento Científico e Tecnológico
Objective ICT-2013.10.2 EU-Brazil research and development Cooperation

Document control page

Document file: D2.1.1 Initial Requirement Report v1.0.doc
Document version: 1.0
Document owner: Jesper Thestrup (IN-JET)

Work package: WP2 Requirements Engineering and SDP Architecture
Task: Task 2.2 SDP requirement analysis and engineering
Deliverable type: R

Document status: ☒ approved by the document owner for internal review
☒ approved for submission to the EC

Document history:

Version	Author(s)	Date	Summary of changes made
0.1	Jesper Thestrup (IN-JET)	2013-10-21	TOC with initial descriptions
0.2	Jesper Thestrup (IN-JET)	2014-01-27	First full draft ready for internal review
1.0	Jesper Thestrup (IN-JET)	2014-01-31	Updated with internal review comments. Final version submitted to the EC

Internal review history:

Reviewed by	Date	Summary of comments
Peeter Kool (CNET)	31-01-2014	Accepted with minor comments
Enrico Ferrera (ISMB)	31-01-2014	Accepted with minor comments

Index:

1. Executive summary

This deliverable contains the method and work undertaken to produce an initial set of typical user generated domain specific applications. These descriptions are the basis for the scenarios that software developers could be asked to design and develop in the context of IMPReSS energy efficiency applications and services.

The deliverable was due in M3 (November 2013) but due to delays in funding on the Brazilian side, the verification of the end-user scenarios has been delayed resulting in the number of initial requirements collected has been lower than planned. However, the quality of the requirements in combination with several discussions in teleconferences and meetings has allowed WP3 to define the first IMPReSS architecture with only a slight delay.

1.1 Methodology

The IMPReSS project has applied an iterative approach and participatory design (Asaro 2000) for developing the technological parts of the System Development Platform (SDP) which is initiated from the requirement engineering process. The methodology involves iterative methods with annual evaluations and milestones. The methodology calls for requirements and stakeholder analysis derived from scenario thinking.

After the successful completion of a prototype cycle, each RTD work package will report their Lessons Learned during the development and integration work as well as other relevant knowledge gathered in the process. The Lessons Learned and the validation reports will be analysed and used for updating of the requirements with new, deleted and changed requirements. Any modification of the requirements is thus documented and traceable by its corresponding Lessons Learned.

The re-engineered requirements will be documented in deliverables *D2.1.2 Requirement and Lesson Learned Report* in M18.

1.2 Scenarios

The scenarios have been chosen along four principles: They must have a) cross sectoral applicability, b) address heterogeneity of the Internet of Things and their networks, c) require management of large amounts of data, and d) human centric applications.

The applications to be built will take the real world scenario involving mixed criticality applications that share electrical power, device, and services and communicate among them to negotiate the electrical usage with the power provider in order to balance the load of the global SmartGrid network.

The first domain application will be deployed in the Teatro Amazonas Opera House as an attractive showcase to demonstrate the potential of a smart system for reducing energy usage and CO_2 footprint in an existing public building. The electricity consumption in Teatro Amazonas is very high. The main consumption of electricity is for lighting and air conditioning, but management does not have any indication on what electricity is used for. Equipment, sensors and devices must fit into the theatre seamlessly in respect for the building. No interruption of the daily operation of the theatre is allowed. Sensor must thus be unobtrusive, aesthetically and well designed showing respect for the building.

The IMPReSS platform will allow developers to integrate existing legacy systems in the theatre, use the available building blocks, orchestrate them to create an intelligent energy management system that relies on natural resources (like daylight and solar energy) and control the operation of both passive and active environmental systems to ensure the best possible comfort conditions with the most efficient use of energy. Six use cases have been developed to describe the scenarios implemented in the real world.

The second domain application will be deployed in the Federal University of Pernambuco (Universidade Federal de Pernambuco, UFPE) located in Recife. The UFPE Campus has two energy measurement cabins covering 13.8 kV primary voltage distributions. Implemented as early as 2003, these are supplied by the local energy distribution company and represent the Campus energy connections to the outside world. Legacy control systems exists and IMPReSS applications should reuse as much as possible, e.g. existing controls and meters. The University would also like to use existing PLC (Power Line Communication) tools.

The first pilot at UFPE will provide a IMPReSS energy management framework addressing reduction in energy usage and CO_2 footprint over the entire Campus. The university wants to measure their own consumption per building and no management tools to show energy consumption are in place at present. We expect to see reduction up to 10-15% in energy consumption from this pilot.

The second pilot should prove that cost oriented use means that cheaper energy may be purchased and stored during off-peak hours and distributed during high consumption hours. A publish-subscribe model may be used to dynamically manage the distribution of stored energy among the different storage and consumption sites. The pilot will deploy flywheels, electrochemical capacitors, and mature storage technologies such as conventional pumped hydro, compressed air and lead-acid batteries.

In the third pilot, the University wants to increase the amount of co-generation and investigate how a small photo voltaic installation of 4kW can be incorporated.

Eleven use cases have been developed to describe the scenarios implemented in the real world.

1.3 Requirements

Developers have analysed the scenarios mentioned above and the use cases in order to facilitate the development of technical requirements. A total of 31 requirements have been created in the period. 17 of these requirements are functional requirements and 14 of them are non-functional requirements related to operation, maintainability, security, and culture/political.

All requirements are listed in Appendix 1.

2. Introduction

The aim of the IMPReSS requirement engineering process is to maintain a continuous discovery and analysis of user centric requirements, needs and prospects, to be used in the design, development, implementation and validation of the SDP and the pilot applications.

After the requirements have been collected, prioritized, and validated the technical partners have analysed and evaluated them and unfold detailed design principles for the IMPReSS technical components. The work leads to a proper architectural design for software development platform including development tools and infrastructure components.

This deliverable contains the method and work undertaken to produce an initial set of typical user generated domain specific applications, which in turn forms the scenarios that software developers typically will be asked to design and develop in the context of IMPReSS energy efficiency applications and services. Software developers have then assisted in analysing the scenarios and deriving the technical requirements for the IMPReSS SDP.

The deliverable thus covers the work of the WP2 partners during the first five months of the project. The deliverable was due in M3 (November 2013) but due to delays in funding on the Brazilian side, the verification of the end-user scenarios has been delayed resulting in the number of initial requirements collected has been lower than planned. However, the quality of the requirements in combination with several discussions in teleconferences and meetings has allowed WP3 to define the first IMPReSS architecture with only a slight delay. The deliverable also documents the achievement of milestone *MS21 Requirements established and engineered* in M5 instead of M3 as planned.

The aim is now to continue to add requirements to the repository for inclusion in the first year prototype and to report these requirements in the updated deliverable D2.1.2 Requirement and Lesson Learned Report due in M18.

2.1 Purpose, context and scope of this deliverable

The purpose of this deliverable is to document the requirement process undertaken and set out an initial set of qualitative and quantitative requirements.

In chapter three, the specific methodologies and processes that will be used are described. The methodology is based in evolutionary design and refinement re-engineering principles. Lessons Learned obtained during project progress will be used to arrive at adjustments of the initial requirements incorporating and inclusion of new requirements.

The following chapter four contains a description of the energy management systems in the two domains: Teatro Amazonas and the Federal University of Pernambuco Campus in Recife. The domain descriptions are a crucial factor for the future establishment of new, innovative procedures and processes based on the IMPReSS platform and understood by all partners. Current state of play with opportunities and constraints for building management systems are described along with some high level policies within complex building energy management. A number of scenarios for intelligent energy management systems are then identified. They focus on optimizing the energy consumption, storage of energy, motivational tools for energy conservation etc. The scenarios are completed with a series of specific use cases that will be implemented in the pilots. The scenarios help defining the requirements for the IMPReSS system development platform that can be used to facilitate the rapid development of applications for managing mixed criticality applications

The scenarios and use cases are then converted into technical scenarios in chapter five. The technical scenarios prioritize functional and non-functional requirements of the system development platform in a more technical sense, without actually setting requirements for the design of the system. This phase is left to the RTD work in the project.

Chapter six highlights the initial set of functional and non-functional requirements for the IMPReSS platform. The complete set of initial requirements following the Volere template layout is documented in Appendix 1. Finally, chapter seven provides a conclusion regarding the first five months of the requirements engineering process.

2.2 Background

The IMPReSS project aims at solving the complexity of system development by providing a holistic approach that includes an Integrated Development Environment (IDE), middleware components, and a deployment tool as depicted in Figure 1.

The main technical and scientific objectives of the IMPReSS project are:

- Developing an Integrated Development Environment (IDE) to facilitate Model-Driven Development of Smarter Society Services.
- Providing a Service-Oriented Middleware to support Mixed Criticality Applications on Resource-Constrained Platforms.
- Developing easy-to-use and configurable tools for Cloud-based Data Analysis and Context Management.
- Network and Communication Management to manage the heterogeneity of Internet of Things.
- Creating efficient Deployment Tools for Internet of Things applications.

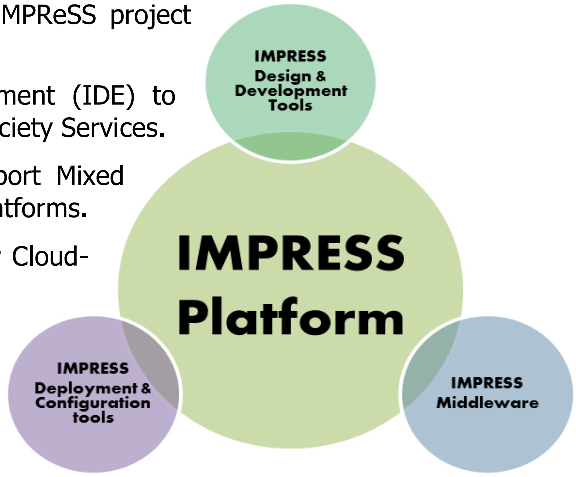

Figure 1 The IMPReSS System Development Concept

The IMPReSS development platform consists of a set of technologies which help to build general-purpose applications accessing to a plethora of sources, such as information from the physical world, analysing and fusing relevant data and perform monitoring and control operations on complex system. This is achieved through the definition of a number of tools and pre-defined modules that can be managed and combined in order to define a specific logic flow. In Figure 2 the IMPReSS development platform is presented.

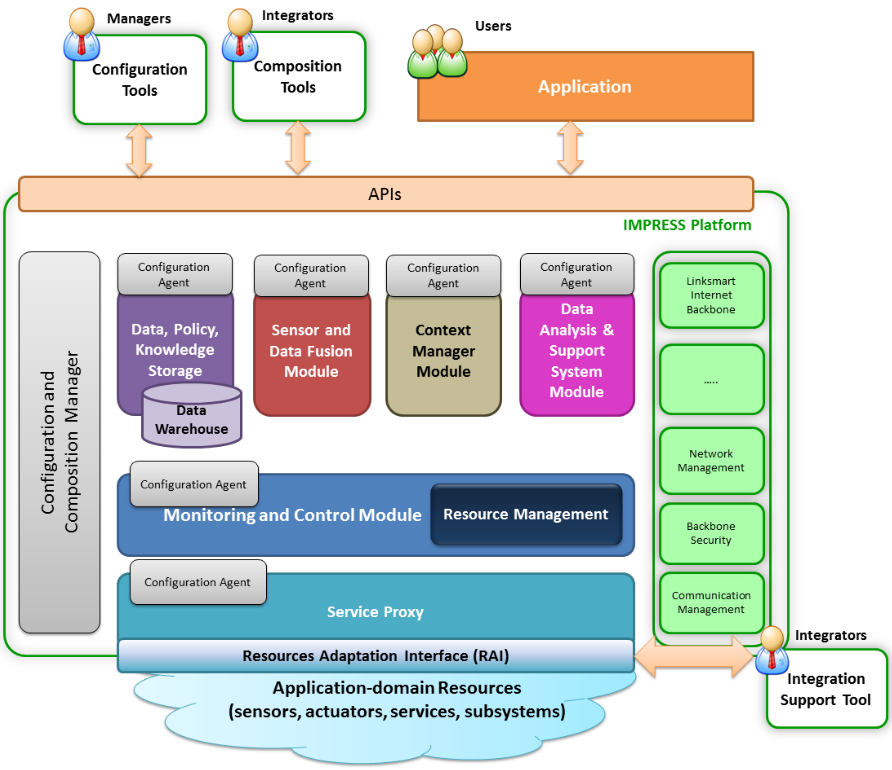

Figure 2 The IMPReSS platform.

3. Methods and Processes

3.1 Overview of the requirement engineering process

The IMPReSS project applies an iterative approach and participatory design (Asaro 2000) for developing the technological parts of the SDP platform which is initiated from the requirement engineering process. The methodology involves iterative methods with annual evaluations and milestones.

The evolutionary requirements engineering, specification and design methodology is underpinned by a strong user-centric development, which complies with the following template in each iteration:

- Requirements for development of sustainable applications
- Developer requirements gathering and engineering
- Architecture design specification and refinement
- Enabling technologies research to implement the architecture
- Prototype development of the platform, system integration and testing
- Evaluation of the development platform in the real application development.
- Lessons Learned and change analysis leading to requirements refinements

This deliverable covers to first two items of the template.

3.2 User driven requirements gathering

The methodology calls for requirements and stakeholder analysis derived from scenario thinking. These requirements would encompass the needs and priorities of all users as well as the wider marketability and exploitation of the resulting solution.

In order to drive of the IMPReSS SDP features, the technological work in the project will be inspired by the scenarios describing development of Smart Cities. Particularly, smart energy management applications have been chosen in order to show the challenges those cities are facing in order to make the population smarter.

The applications have been chosen along four principles: They must have a) cross sectoral applicability, b) address heterogeneity of the IoT and their networks, c) require management of large amounts of data, and d) human centric applications.

The applications to be built will take the real world scenario involving mixed criticality applications that share electrical power, device, and services and communicate among them to negotiate the electrical usage with the power provider in order to balance the load of the global SmartGrid network. The applications will be developed during the annual prototype development and deployed in the university campus in Recife and in the Teatro Amazonas, both in Brazil.

3.3 Requirement gathering and formulation

The analysis of scenarios and use cases enables the development of an initial set of requirements. The requirement process is based on the Volere requirements mastering process. The Volere process ensures that all important aspects of requirements are carefully addressed and that the methods applied have proven their value in practical work.

The Volere template ensures that the results are documented in a way that can be communicated efficiently to developers. Assigning a rationale and an appropriate classification is done in several group discussions within the project team. Finally, quality checks have been performed with the aim of eliminating redundancy in clusters of equivalent statements and phrasing the essential meaning in one statement.

During the first months of the project, the Atlassian JIRA tool[1] has been installed and configured with the Volere template. JIRA is a web-based issue tracker that allows implementing and tracking a collaborative workflow and is used as a tool for gathering and sharing requirements amongst developers. Figure 3 shows a screenshot of JIRA with one open requirement.

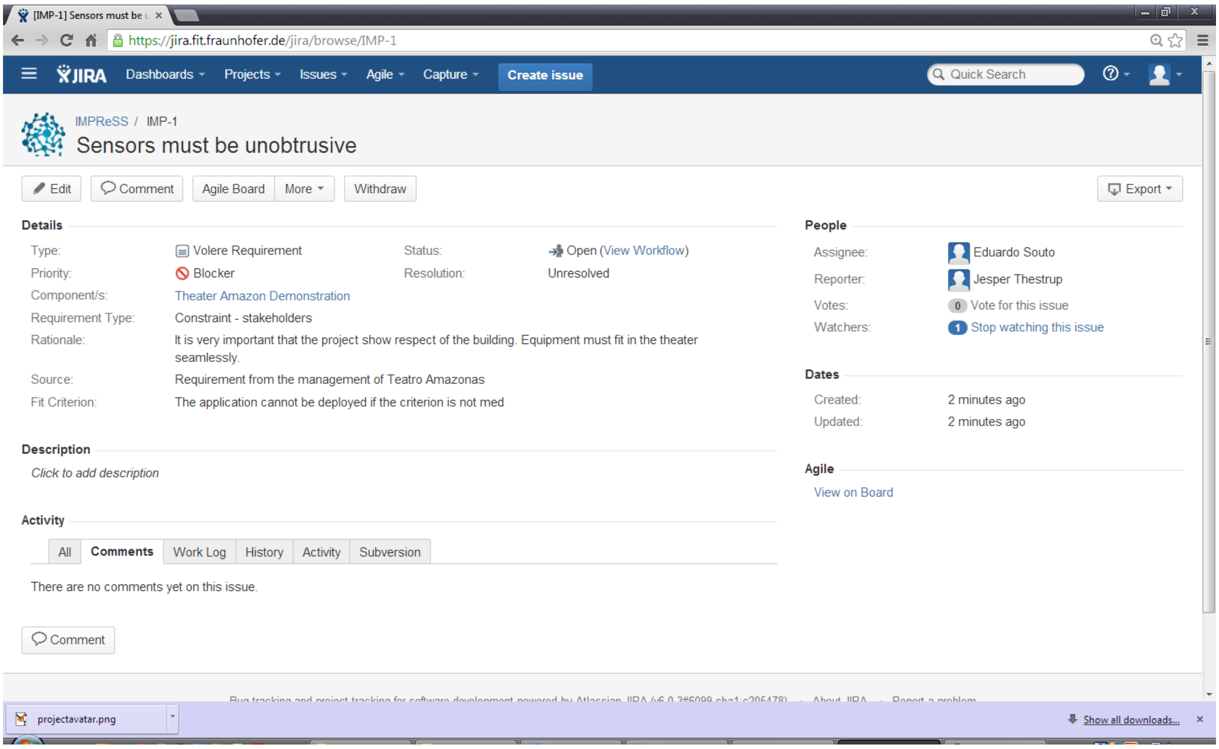

Figure 3 Screenshot if JIRA showing a single requirement based on Volere

All requirements in the Volere schema fall in one of three categories:

Functional Requirements
User requirements that explicitly refer to the functionality of IMPReSS are referred to as functional user requirements. Functional requirements further classify into general, system, component and human-computer interface (HCI) requirements.

Non-functional Requirements
Non-functional requirements address the quality of the future system and are classified to various criteria according the Volere schema (usability, performance, operational requirements, maintainability etc.).

Constraints
Restrictions imposed on the platform due to the budget, the time or the way the platform is designed or will work or interact with third-party components. Constraints are the same as other requirements except that they are mandated usually at the beginning of the project.

The quality and management control of the progress of implementation of requirement is realised by processing all requirements along the steps of a workflow is depicted in Figure 4 below.

Each requirement has a *state* (Status) in the workflow that changes depending on the current workflow step. The process is designed to ensure that the quality of the requirements is always controlled by two persons. One person enters the requirement and the other person checks and passes the requirement through the quality gateway.

[1] https://www.atlassian.com/software/jira

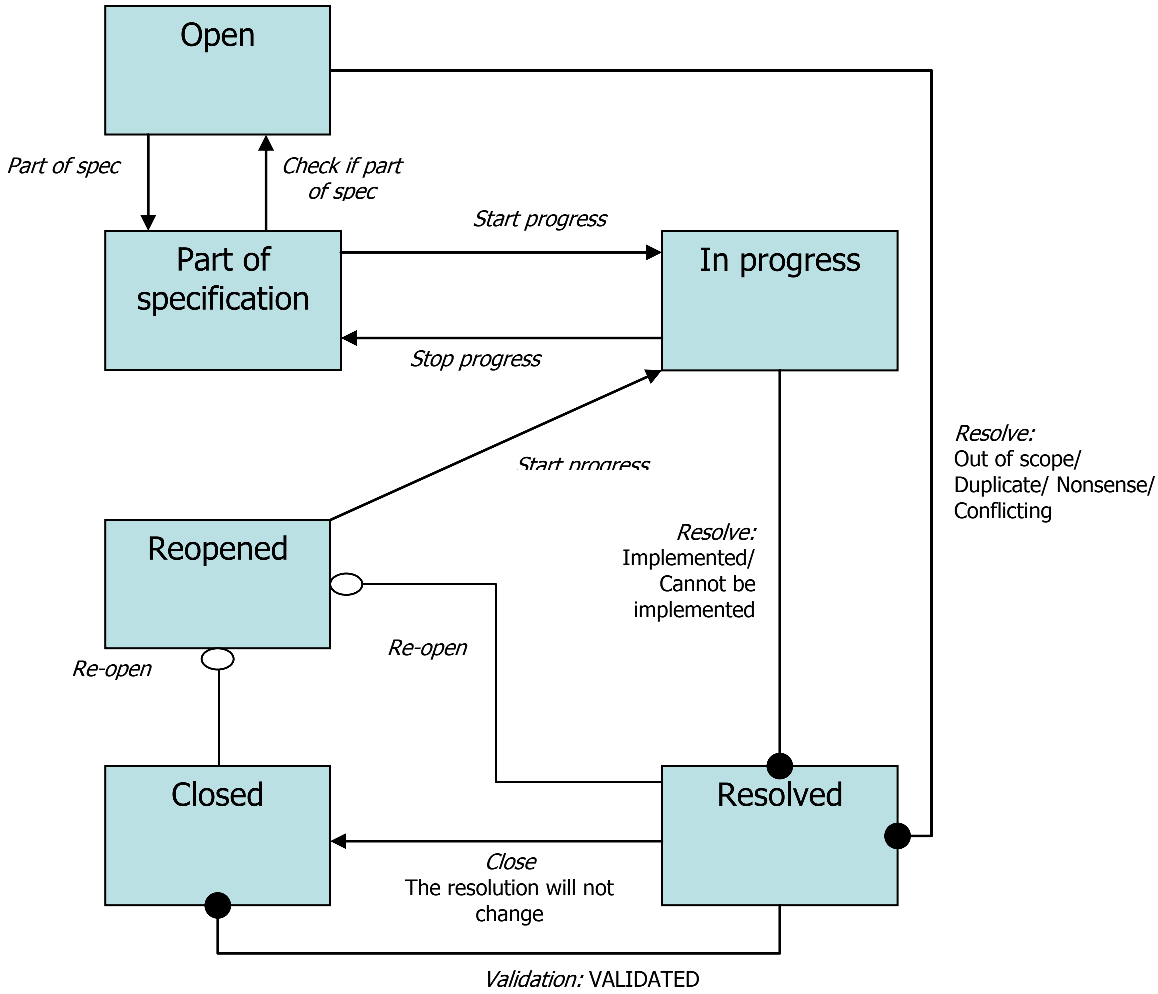

Figure 4 The JIRA workflow

All requirements have to pass the quality gateway. This means that they are complete and all fields are sensibly filled in. A requirement that passed the quality gateway cannot (should not) be edited any more during the current iteration cycle. The last step in the quality check is to decide whether a requirement becomes part of the specification or discarded. The latter can happen for different reasons: A requirement can be a duplicate of another requirement, it may be conflicting with (an)other requirement(s), it may be nonsensical or it may be out of the project's scope. Requirements resolved this way are assigned Resolutions 'Duplicate', 'Conflicting', 'Nonsense' or 'Out of Scope', respectively.

A requirement, which has been quality checked and deemed 'Part of Specification' is then assigned to a specific person (the Assignee) for implementation and management. The status is then 'In Progress'. When a requirement's status is 'In Progress' it can be stopped temporarily and the status changes to 'Part of Spec'. If during the development it is concluded that the requirement is either completely implemented or cannot be implemented, the Status is changed to 'Resolved' with the appropriate Resolution.

The final step for an implemented requirement is its validation. Validation is conducted by internal or external end-users according the procedures in *D2.3 Validation Framework.*

Any requirement with Status 'Resolved' or 'Closed' may be reopened; the Resolution automatically changes to 'UNRESOLVED'. Typically the Status is changed to 'In Progress' for further work to be initiated. A resolved requirement may also be reopened because it fails validation. A closed requirement might be reopened because it was decided that it was not 'Out of Scope' after all.

3.4 Re-engineering of requirements

After the successful completion of a prototype cycle, each RTD work package will analyse and report their development results, RTD experiences, Lessons Learned in the development and integration work and other relevant knowledge gained during the development cycle. Moreover, knowledge gained from formal testing, system integration and validation will be collected together with latest development in technology, regulatory affairs and markets, which influence the IMPReSS SDP and its exploitability.

3.4.1 Lessons Learned

Lessons Learned are a principal component of a project culture committed to Knowledge Management. Lessons Learned help to support project goals in the RTD work of:

- Promoting recurrence of successful outcomes
- Precluding the recurrence of unsuccessful outcomes

As part of the continuous improvement program adopted by the IMPReSS Project Board a systematic and continuous collection, indexing and dissemination of Lessons Learned will be undertaken in WP2. This section will establish criteria for the Lessons Learned process in IMPReSS and discuss how to turn Lessons Learned into Lessons Applied.

Lessons are learned during project RTD work, during testing and integration, as a part of the validation of project prototypes and during literature search and technology watch reports. Lessons can thus be learned throughout the project work. As such, Lessons Learned constitute both individual and organisational knowledge and understanding gained by experience, either negative (missed targets, solutions that do not work as expected, wrong choice of technology) as well as positive (easier implementation than expected, faster response time, more interoperable devices than expected).

In order to implement a workable Lessons Learned process, we need first to define what we understand with the term "lesson". We use the following characterisation for a lesson:

- It must be significant in terms of the project progress and ability to meet its goal.
- It must be valid, i.e. the experience gained must be repeatable.
- It must be applicable to the IMPReSS project
- It may contain or address pertinent info
- It may provide information of interest

Not all experiences will qualify as being Lessons Learned and it is important that reported Lessons Learned not merely restate existing information and existing experiences not related to IMPReSS work.

The IMPReSS Lesson Learned process has 6 steps:

- Collection
- Verification
- Storage
- Dissemination
- Reuse

Collection
The collection process focuses on collecting Lessons Learned from many sources internal and external to the project. The collection will be undertaken in practically all work packages. Lessons Learned is collected from the iterative requirements engineering process which can be reused to improve the performance and efficiency of future iterations. The RTD work undertaken will provide a large amount of Lessons Learned, by virtue of the many researchers participating in this work and the many small and large experiences gained individually and as teams. The challenge here is to identify and properly describe the Lessons Learned and filter them according to significance, validity, and applicability to the IMPReSS project. The field trials will also provide a range of experiences that can be classified as Lessons Learned. So will the implementation and integration work.

Verification
Verifying the collected Lessons according to established standards is the second step in the process. All Lessons Learned must be verified for correctness, significance, validity, and applicability. The verification will be together with the Project Manager and relevant WP leaders. The Project Manager will decide to add and remove Lessons Learned as necessary. Some of the criteria that may be used for verification are:

- Relationship with the project flow
- Relevance to the project outcome
- Significance in terms of quality parameters such as robustness, ease of use, functionality
- Systemic process issues
- Credibility or reputation of the originator

Storage
In the first instance, the Lessons Learned will be entered into a reserved area of the IMPReSS Wiki. The area has been created and is maintained by WP2. It contains a simple categorisation tagging for filtering purposes. For the sake of simplicity, a very simple template will be provided with no special structure or format needed. The Lessons Learned repository will act as an organisational memory for experiences incurred by all project members during the cause of the project.

Dissemination
A very important part of the process is of course to inform other users in the feedback cycle. All project workers are encouraged to continuously consult the Lessons Learned repository, not only with the purpose of reporting, but also to continuously follow Lessons Learned by other project partners. The Lessons Learned will be documented in deliverable *D2.1.2 Requirement and Lesson Learned Report* in M18.

Reuse
The IMPReSS project encourages and promotes lessons to be used by other than the submitter. The WP leaders have a responsibility to consult the Lessons Learned repository regularly and at least before any major decision affecting the scientific work and project outcome is to be made. The WP leaders are obliged to take part in the engineering process of requirements, which is based on a timely assessment of the reported Lessons Learned.

3.4.2 Requirements improvement opportunities and change request reports

The last step in the process relates to the identifying of incremental and innovative improvements that will measurably improve the project's requirement specification. Input from the Lessons Learned from the validation reports will be analysed and used for updating of requirements and any modification of requirements (addition, deletion or change) is thus documented by its corresponding Lessons Learned. From the relevant analysis new and/or updated requirements will be extracted. The identification, formulation and validation of requirements will be performed together with the Project Manager and WP leaders. JIRA will be used for gathering and sharing requirements changes.

The new and updated requirements will be scrutinised and prioritised by the Project Manager and relevant WP leaders. A handful of the requirement changes with the highest priority for each WP will be selected. New user requirements that conflict with existing requirements will be resolved by the team. This process will be part of the quality gateway process described above. Conflict resolution will be solved by evaluating the relevant requirements against the overall project vision, the exploitability of the final project results, and the cost of implementation. Important conflicts with substantial impact on the architecture description shall be referred to the Project Board for discussion and resolution.

The Project Manager and the WP2 leader will evaluate and describe the impact on the future development work arising from the re-engineered requirements and report this in a Change Request Report, which also will contain the relevant Validation Reports. The evaluation will take place both in the context of scientific and technological progress, adherence to the overall vision of the project and impact on the overall project plan.

The re-engineered requirements will be documented in deliverables *D2.1.2 Requirement and Lesson Learned Report* in M18.

4. Application domain analysis and scenarios

Accurate identification of intelligent energy management systems in the theatre Amazonas and university buildings, and the business processes used in their domain, is a crucial factor for future establishment of new, innovative procedures and processes based on the platform and understood by all partners. Current state of play for building management system will be described by using cases from Brazil and Europe as well as from the effort for energy optimizations e.g.: Scheduling heating and cooling systems. The precise definition of single roles and procedures, together with policies within complex building management systems will allow the requirements to be correctly identified and described. Visionary scenarios for intelligent building management system with the goal to further optimize the energy consumption will be investigated.

4.1 Teatro Amazonas Energy Pilot

The first domain application will be deployed in the Teatro Amazonas Opera House as an attractive showcase to demonstrate the potential of a smart system for reducing energy usage and CO_2 footprint in an existing public building, that obviously allows no significant retrofitting and construction works.

The Teatro Amazonas is located in Manaus, in the heart of the Amazon rainforest. It is the location of the annual Festival Amazonas de Ópera (Amazonas Opera Festival) and the home of the Amazonas Philharmonic Orchestra who regularly rehearses and performs here along with choirs, musical concerts and other performances.

The theatre was built during the Belle Époque at a time when fortunes were made in the rubber boom. Construction of the theatre was first proposed in 1881 by a member of the local House of Representatives, Antonio Jose Fernandes Júnior, who envisioned a "jewel" in the heart of the Amazon rainforest. In 1882, the State legislature approved some limited financing, but this was considered insufficient. Later that year, the president of the Province, José Lustosa Paranaguá, approved a larger budget and initiated a competition for the presentation of plans.

Figure 5 The Teatro Amazonas historic building

The chosen project was made by the Gabinete Português de Engenharia e Arquitectura, an engineering and architecture office from Lisbon. By 1884, construction was ready to begin under the Italian architect Celestial Sacardim. Work proceeded slowly over the following fifteen years with some stops and restarts from 1885 to 1892. By 1895, when the masonry work and exterior were completed, the decoration of the interior and the installation of electric lighting could begin more rapidly. The theatre was inaugurated on December 31, 1896, with the first performance occurring on January 7, 1897, with the Italian opera, La Gioconda, by Amilcare Ponchielli.

Currently the theatre has a capacity for 701 people, distributed between the audience and the three floors of cabins. In the Great Hall, with Baroque features, highlights the ceiling painting, called " A Glorificação das Bellas Artes na Amazônia ", 1899, authored by Domenico de Angelis. Noteworthy are the ornaments on the columns of the ground floor, with masks in honour of famous classical composers and playwrights such as Aeschylus, Aristophanes, Moliere, Rossini, Mozart, Verdi and others. About the vaulted ceiling are affixed four screens painted in Paris by Carpezot House - the most traditional of the time - they are portrayed allegories to music, dance, tragedy and a tribute to the great Brazilian composer Carlos Gomes. From the centre hangs a golden chandelier with crystals, imported from Venice, which descends to the level of the seats for carrying out maintenance and cleaning.

4.1.1 Opportunities and constraints

Although most of the electricity in Brazil is from hydropower, electrical energy is very expensive in Brazil. Electricity rates vary in Brazil; some cities are more expensive than other cities. A survey[2] in 2011 revealed that in Rio de Janeiro, the rate for consumers per MWh was R$ 420 whereas in Belo Horizonte, the rate was R$ 580. However, in 2012, the Brazilian government announced a major cut in electricity taxes that lowered the high energy costs for industries and residential consumers. This was an attempt by President Dilma Rousseff to re-energize her country's once-booming emerging economy. The new scheme reduced power rates by up to 28 percent for industries and 16 percent for households. Presently, the rate in Manaus is 266,12 R$ (88.58 €) per MWh including taxes (ANATEL).

The electricity consumption in Teatro Amazonas is very high. The main consumption of electricity is related to lighting and air conditioning, but management generally does not have any indication on what electricity is used for.

According to the management of the theatre, it is very important that the project not only adheres to preservation laws of Brazil, but also shows respect for the building. Equipment, sensors and devices must fit into the theatre seamlessly and no interruption of the daily operation of the theatre is allowed. Sensor must thus be unobtrusive, aesthetically and well designed showing respect for the building.

The electrical energy supply system is very old and technically outdated. So are the installations in the building. Any management system will thus be an improvement. A diesel generator backup has been installed, because the electricity supply is not stable and the entire installation must be robust under unstable power supply conditions. The standard power supply in the theatre is 110V @ 60Hz.

Water supply management is a secondary interest for Teatro Amazonas. Severe water shortage in the North-Western part of Brazil has caused massive problems and the present distribution systems are not optimal. It is estimated that app. 30% of the water is lost in distribution.

4.1.2 Scenarios for electrical energy management

The IMPReSS platform will allow developers to integrate legacy systems, use the available building blocks, orchestrate them to create an intelligent energy management system that relies on natural resources (like daylight and solar energy) and control the operation of both passive and active environmental systems to ensure the best possible comfort conditions with the most efficient use of energy.

The intelligent Energy monitoring and smart control of appliances at the Teatro Amazonas will effortlessly optimise efficiently the energy usage without compromising comfort or convenience and simultaneously provide a platform with an enormous publicity potential establishing the necessary public awareness to change the attitude towards energy usage and CO_2 footprint and creating a wide public base for necessary changes in actual human behaviour to realize long-term energy savings. The demonstration will use its real-time energy-awareness services for all users of the Teatro Amazonas.

Additionally this showcase will be a figurehead with enormous potential creating international awareness far beyond the Teatro Amazonas, the City of Manaus and Amazonas State and moreover create a touristic attraction on its own, showing a successful integration of national historic heritage and green awareness and ecological sustainability.

These ideas for concrete application in this scenario will be further investigated in the project:

- LED-light-based real-time ambient energy awareness in the Lobby and in the public spaces around the Teatro Amazonas.
- Public interactive displays informing about details of the savings.

[2] http://smartgridresearch.org/news/electricity-prices-in-brazil-prices-paid-by-consumers-and-businesses-high-electricity-prices-may-have-to-go-down/

- Additional information material on how to realize energy saving with easy behaviour changes at their own home etc.
- Reporting of energy saving related activities of the Teatro Amazonas staff and of similar activities from the staff e.g. from the employees of the Valentino Castle in Turin competing with Manaus etc.

Applications for energy management systems are the primary interest for the theatre. Any management system will be an improvement over the present status.

The management system must be able to measure, analyse and show (also in real time) energy consumption and the building managers must to be able to analyse the relation between energy consumption and occupation in the different sectors of the theatre. Another need is for a management system that can reduce energy costs through intelligent comfort control and inclusion of cheaper, Renewable Energy Resources (RES). Finally, the management system must be able to compensate for instabilities in power supply through short and long term forecasting and management of consumption.

The *end-user benefits* to be realised are:

- Cost reductions
- Lowering of CO_2 emissions
- Visibility of energy reducing measures to city bodies, environmental accounts, and the public
- Comfort/consumption optimization (public, employees, artist)
- Power stability
- Conservation of precious arts objects.

4.1.3 Use cases for electrical energy management

The scenario contains the following use cases (applications):

1. Consumption analysis: This application analyses where electrical energy is used. A management system should be able to measure energy used in broad groups such as cooling, lighting, motion (elevators, hoisting, and pumps), indoor, outdoor, offices, artistic areas, public areas, etc. A different viewpoint would be energy consumption per section of the building. The energy consumption of the building must be monitored in the daily use, during the time of public performances and during periods closure (nights) because usage profiles is expected to be very different.
2. Displaying the results: The energy consumption shall be displayed in technical reports to operators, in management reports to management and on public display boards outside the building. The reports shall include present consumptions, savings, environmental protection KPI's for the theatre to the public, the city, the government. The report shall show the energy profile both as historical data and as current consumption levels.
3. Comfort control: This application controls the comfort of the building with the aim to reduce energy. By controlling temperatures and humidity in various locations (public spaces, offices) the application is able to control and minimize the energy consumption.
4. Stabilising energy supply: The application will control and manage energy usage and perform prioritised actions for overcoming eminent power shortages or instabilities. The control philosophy is to rather leave 10% without power and remain operation in the remaining 90% of the building. The system shall able to switch off equipment and devices that are left on unintentionally. The application relies on network monitoring (inside the theatre) based on network topology map and known constraints and is thus able to plan distributed UPS (Uninterruptible Power Supplies) placed on strategic radials of the network.
5. Prediction energy consumption: This application uses algorithms for predicting energy usage based on historic data, external forecasting sources such as cost of energy, weather, etc.
6. Including Renewable Energy Rources: This application shall allow for the inclusion of new, renewable sources of energy such as voltaic panels, wind, biofuels, etc. Also energy storage units shall be included

4.2 University Energy Pilot


Figure 7 The main campus building at UFPE

Federal University of Pernambuco (Universidade Federal de Pernambuco, UFPE) is a public university located in Recife, Brazil, established in 1946. UFPE has 70 undergraduate courses and 175 postgraduate courses. The university has three campuses: Recife, Vitória de Santo Antão and Caruaru. Its main campus, or "Cidade Universitária", has 10 centres in 149 hectares. It is located in the west part of Recife, in the Várzea neighbourhood.

UFPE ranks among the top Brazilian universities, being the ninth university both in size and scientific production, and the seventh among the federal institutes. UFPE has been elected twice as the best university of north and north-east Brazil by Guia do Estudante (a national university ranking magazine) and Banco Real (ABN AMRO).

4.2.1 Opportunities and constraints

The UFPE Campus has two energy measurement cabins covering 13.8 kV primary voltage distributions. Implemented as early as 2003, these are supplied by the local energy distributer company and represent the Campus energy connections to the outside world. Billing is based on the readings at these two entry cabins which causes a serious lack of high granularity intra-campus energy consumption data. The first cabin manages 40% of the total load currently reaching 2,900 kW whereas the second cabin leads with the remaining 60% which also represent 4,450kW. The concentrated supply system offers little sectoring and is purely radial. As a result, when a failure takes place somewhere, there is little possibility for isolating this and confining to a small area.

Despite the lack of flexibility, work is underway to install the University's first substation operating at 69kV. This station allows some limited remote readings and simple monitoring of energy consumption. The university manages digital energy measurement devices, but these have so far been only installed in the large consumption Campus sites which currently numbers 14.

Legacy control systems exists and IMPReSS applications should reuse as much as possible, e.g. existing controls and meters. The University would also like to use existing PLC (Power Line Communication) tools.

4.2.2 Scenarios for electrical energy management

In the project pilot we first look to a scenario that provides a sustainable LinkSmart based IMPReSS energy management framework addressing reduction in energy usage and CO_2 footprint over the UFPE Campus. The university wants to measure their own consumption per building, and have replaced the provider's equipment with their own. No management tools to show energy consumption are in place at the present. We expect to see reduction up to 10-15% in energy consumption from this pilot.

The second pilot sees the ubiquitous introduction of LinkSmart managed advanced energy storage units across the Campus. This scenario should prove that their cost oriented use means that cheaper energy may be purchased and stored during off-peak hours and distributed during high consumption hours. A publish subscribe model may be used to dynamically manage the distribution of stored energy among the different storage and consumption sites. The pilot will deploy flywheels, electrochemical capacitors, and mature storage technologies such as conventional pumped hydro, compressed air and lead-acid batteries. A detailed analysis will to estimate the requirements for electricity storage across the whole campus power system and its "corresponding value" to the University. Storage of energy would be beneficial for use, when the energy is cheap.

In the third scenario, the University wants to increase the amount of co-generation and how it can be incorporated. The University has a small photo voltaic installation of 4kW. There is no software interface to the panels.

The *end-user benefits* to be realised are:

- Cost reductions
- Overcome power instability and black-out
- Energy consumption behaviour (visibility, motivation, publicity)

4.2.3 Use cases for electrical energy management

For this scenario, the following specific use cases will be implemented:

1. Different critical applications accessing shared sensors, actuators, and energy supply services such as personal monitoring app, centralized energy policy, safety critical equipment.
2. Include RES and energy storage units (chillers in water Air Conditioning Systems) in the network. The system should store the energy from the utility company during night and supply the energy during day.
3. Promotion of energy conservation with public information campaigns using public display for increasing public awareness. Visualization of the energy consumption in the different rooms as a public display. Show the saving potential to students and campus employee by e.g. showing the consumptions that could have been saved by switching off stuffs, which are not needed.
4. Prediction of future energy consumption from data analysis, history, trends, class scheduling, etc. using plug-ins for specific analytics.
5. Automatic system that is able to response to mixed criticality events such as fall out of power in part of the net. Selection of priority of emergency shut down in case of power black-out, servers to be powered down in an orderly fashion in case of predicted power failure and switched on when power resumes, orchestration of different service schedules in the correct order.
6. System for energy control according to day-ahead tariffs. Negotiation protocol with energy providers to determine how much max consumption is allowed by the provider, buying and selling electricity.
7. Building energy management according to context, e.g. need for cooling and lightning. Energy forecasting tools to plan cooling system accordance to the activity in the rooms
8. Support for real-time trending and monitoring of energy consumption and production of a map energy showing the hotspots (locations with high consumption) and the prediction for future infrastructural investments.
9. Improve system reliability by adding redundant components such as sensors, network links connecting all energy devices and the distribution of decision making through LinkSmart.
10. The use of business level policies to guide and ease the management of energy related devices, storage units, meters, etc.
11. Integrating the LinkSmart platform through a SCADA Gateway hence enabling real-time interaction with external energy supplier entities including CHESF.

5. Technical Use Cases

5.1 Teatro Amazonas Energy Pilot

Developers have provided the following analysis on the use cases in order to facilitate the development of technical uses cases and extraction of requirements.

- Sensor network must rely on data models, resource models, and inter-linkage. The SDP must provide support for the different types of communication and data format that are used.
- The platform should provide a single logical view from the physical distributed system.
- Multiple communication protocols and data format should be supported, but it should not be necessary for the developers to understand the network protocols, i.e. the network should be transparent for the developers.
- The sensors should be aware of their location, i.e. the position in the logical and physical network with minimal setup efforts.
- The device virtualisation should contain a semantic description that can be retrieved by the application in run time.
- Automatic discovery, self addressing, self geocoding, self resource attribute setting, and self-* properties should be supported.
- The platform should allow developers to select different communication paradigms e.g. pub-sub, orchestrating network of boxes, async-notifications, event-driven.
- The platform shall be deployed on resource constrained platform such as raspberry pi.
- The following sensors are foreseen to measure physical properties: Electrical energy, water consumption, temperature, humidity. Moreover, sensors for controllable, programmable circuit breakers, timers and open windows are also needed.

5.2 University Energy Pilot

Developers have provided the following analysis on the use cases in order to facilitate the development of technical uses cases and extraction of requirements.

- Orchestration: Priority and feed back loop must be available. Every application (service, component) must be given a priority ranking so that it can be orchestrated in a prioritised way. Every application must be able to provide feedback to the orchestration module.
- Architecture: The notion of "application" must be clearly defined. An application may be build from different components (or services).
- Developers need a tool to compose the IMPReSS modules.
- Developers need a template to create configurations for his application. Prioritizing of the Units /application according to its criticality.
- The following sensors are foreseen to measure physical properties: Electrical energy, temperature of servers, OCR readers for analogue displays, etc. Moreover, sensors for presence are also needed.

6. Status on the requirements process

A total of 31 requirements have been created in the period. The number of created requirements vs. the number of resolved (implemented or discarded) requirements is shown in Figure 8. The green line indicates the resolved requirements, but there are none as of yet. As the implementation progresses, the requirements get resolved and the green curve rises to eventually close the gap to the created requirements.

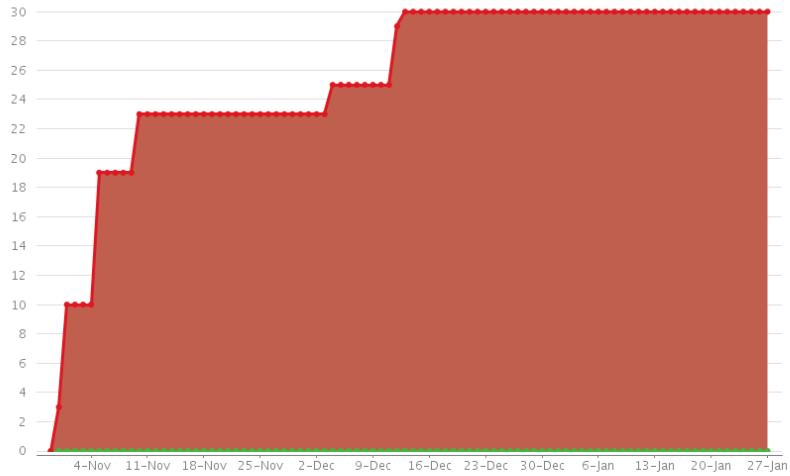

Figure 8 Created vs. resolved requirements as per M4

The requirements cover all fields of the IMPReSS architecture and the two prototypes as shown in Figure 9. Each requirement has been assigned to a specific architectural component or demonstrator prototype:

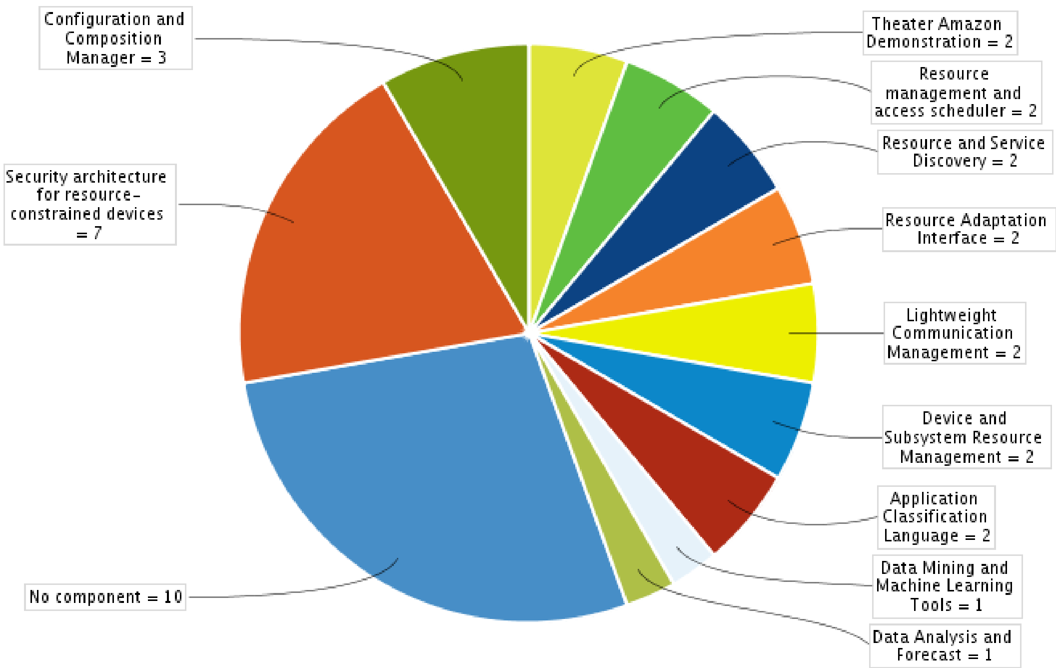

Figure 9 Requirements assigned to components

7. Requirements for the IMPReSS platform

This section contains the condensed list of functional and non-functional user requirements, extracted from the original user scenarios and use cases in the two user domains. The aim of this approach is to provide a simple structured representation of requirements, to be used as a reference for the development of the first pilot applications.

The list of requirements will be updated during the project lifetime, as soon as the need for features is identified. We will apply various methods to improve our understanding of user needs and to improve user-perceived qualities of the prototypes. All the updated requirements will be listed in the updated deliverable D2.1.2 Requirement and Lesson Learned Report due in M18.

Each requirement listed in the following tables obtains a unique ID to refer to. The description of a requirement is a synthetic but clear description of the requirement. The rationale gives a reason why this requirement is relevant for the system and thus has been included into the table. The source gives an indication of where to find the requirement, i.e. scenario, interview or user workshop. According to the Volere scheme the requirements are divided into non-functional and functional requirements, but we will have two more divisions business and efficiency requirements.

7.1 Functional Requirements

Functional requirements are requirements related to the functionality of the IMPReSS SDP platform in view of the developers of energy efficiency, mixed criticality applications and services.

There are 17 functional requirements created.

IMP-3	Devices should be allocated to a logical area
IMP-4	Devices should be allocated to one or more groups
IMP-5	The data should be persisted in NoSQL database
IMP-6	The data should be analyzed using data mining and machine learning techniques to find relevant information and make predictions.
IMP-7	SDP will have a communication layer that allows storing data in the IMPReSS Cloud
IMP-8	The application should provide historical energy consumption and use of electrical devices.
IMP-12	Access prioritization to resources (devices, services, computing power, power supply)
IMP-13	Annotate application with the level of criticality
IMP-15	Model driven tool for orchestrating impress components
IMP-16	Reusable components for trend analysis and forecasting of energy and occupancy data
IMP-17	Dynamically adjustable security level for resource constrained devices
IMP-21	Graphical model-driven commissioning tool
IMP-22	Runtime services/devices discovery and commissioning
IMP-23	Development toolkit for resources integration
IMP-24	APIs definition
IMP-25	IMPReSS architecture views
IMP-26	Templates for smart entities

7.2 Non-functional Requirements

Non-functional requirements address the quality of the future system and are classified by various criteria according the Volere schema (usability, performance, operational requirements, maintainability, etc).

There are 14 non-functional requirements created.

IMP-1	Sensors must be unobtrusive
IMP-2	Impress' cloud must scale horizontally
IMP-9	The SDP should encapsulate the complexity of different technologies, developing a single logic to devices manipulation.
IMP-10	The SDP shall support multiple communication protocols
IMP-11	The software components of the middleware should be modularized, facilitating the inclusion of different technologies with the purpose of integrating heterogeneous resources
IMP-14	The impress core runs on a Gateway that cost below USD50
IMP-18	The IMPReSS platform should support development of IoT systems that are extendable for future needs
IMP-19	The IMPReSS platform should be agnostic to the application domain
IMP-20	The IMPReSS SDP should be easy to use
IMP-27	Data in the IMPReSS network is classified to different categories based on the criticality
IMP-28	Confidentiality of the messages between IMPReSS platform devices can be guaranteed
IMP-29	Integrity of the messages between IMPReSS devices can be guaranteed
IMP-30	Availability of the critical IMPReSS devices must be guaranteed
IMP-31	Data transmitted in the IMPReSS network is classified to different classes based on the confidentiality.

8. Conclusion

The generation of requirements for the IMPReSS platform posed a delicate balance between on the one hand, the requirements end-users placed on the application for energy efficiency that they wanted to see in the pilot applications, and on the other hand, the requirements for a developments system and platform to be used by software developers to develop those applications.

Using simple scenarios in combined workshops was found to be an effective approach to keeping a strong focus on developer users throughout the requirement specification process, and at the same time to allow early end-user involvement.

In the same process, the technical scenarios were created, and used to collect end-user feedback. The user workshop conducted was thus well structured and focused on context of use and developer user requirements.

The user requirements are based on empirical data, i.e. on real users' comments on the vision depicted by the IMPReSS project. The discussions within the user workshops were qualitative, explorative and meant to explore the context of use, as well as the users' values and concerns. In some cases, several users shared a concern, which can be taken as a confirmation of the issue. On the other hand, if only one user mentions an issue, this should not be dismissed merely because it was just a small number of users.

A major challenge was to aggregate the original user statements to a traceable set of more prescriptive user requirements. The Volere template proved to be useful for this step, since the results need to be documented in a way that can be communicated efficiently to the developers of the IMPReSS project. The condensation was done mainly by abstraction, i.e. by eliminating redundancy in clusters of equivalent statements and phrasing the essential meaning in one statement. Assigning a rationale and an appropriate classification was done in several group discussions after the initial workshop.

The next major step within this requirement engineering process is the determination of the impact of the requirements on the architecture and hence the work in each work package. In order to achieve this, the requirements need to be further condensed and the level of quality needs to be improved.

The first iteration of the development process will be finished with testing and evaluating the first prototypes. The test environment will be created according to the technical scenarios that built the foundation for the first iteration. The evaluation of the prototype yields further requirements. Besides new aspects, these requirements can be considered a refinement of the initial set of requirements presented in this document. From our experience in similar projects, we expect that the refined set of user requirements will be more complete, more concrete, and more reliable.

9. References

(Asaro 2000) Asaro, P. M. (2000). "Transforming society by transforming technology: the science and politics of participatory design." Accounting, Management and Information Technologies 10(4): 257-290.

10. Appendix 1: Complete List of Requirements

10.1 Functional Requirements

Key	Requirement Type	Priority	Summary	Rationale	Fit Criterion
IMP-3	Functional	Critical	Devices should be allocated to a logical area	It would be interesting, for clients of the Impress' Cloud, if different devices could be allocated to a logical area, created by the user that represents a physical area (e.g. a room, an office, a bathroom) in the real world.	UFPE
IMP-4	Functional	Critical	Devices should be allocated to one or more groups	It would be interesting, for clients of the Impress' Cloud, if different devices (e.g. heaters, TVs and etc) could be allocated to one or more groups, which is not an equivalent of the physical area these devices are. For instance, one could compare the energy usage patterns among all the heaters, in the same group.	UFPE
IMP-5	Functional	Major	The data should be persisted in NoSQL database	Since the data does not have a well-defined pattern, the solution with NoSQL technologies proved to be more attractive for the flexibility it offers in modeling data.	UFPE, UFAM and FIT
IMP-6	Functional	Major	The data should be analyzed using data mining and machine learning techniques to find relevant information and make predictions.	The large volume of data generated by applications may hide important information that can be easily discovered by advanced data analysis. These techniques enable knowledge discovery, and assist in decision making.	UFAM and UFPE

Key	Requirement Type	Priority	Summary	Rationale	Fit Criterion
IMP-7	Functional	Major	SDP will have a communication layer that allows storing data in the IMPReSS Cloud	Facilitates access to data generated by applications from anywhere on the planet, besides having all apparatus and infrastructure services more cheaply, avoiding unnecessary expenses.	UFAM and UFPE
IMP-8	Functional	Major	The application should provide historical energy consumption and use of electrical devices.	The history of power consumption is important to control the consumption of energy to assist in identifying periods of higher power or even possible irregularities which may occur. Through it is possible to develop consumer policies in order to save energy.	UFAM, UFPE and UFABC
IMP-12	Functional	Major	Access prioritization to resources (devices, services, computing power, power supply)	Applications with different criticality may use the same resources. IMPReSS should be able to prioritize the access to the shared resources particularly when the demands are bigger than available resources.	Best effort algorithms to guarantee the access to the resources and prioritize the access to the shared resources available.
IMP-13	Functional	Major	Annotate application with the level of criticality	Developers may want to explicitely categorize the criticality of applications	Developers are able to define the level of criticality e.g. : - user interaction --> soft real time, delay max 300ms. - monitoring office consumption --> non critical delay max 1 minute
IMP-15	Functional	Major	Model driven tool for orchestrating impress components	Developers may want to wire components without having to understand the APIs	A model driven tool is available which allow developers to connect the required components for his application.
IMP-16	Functional	Major	Reusable components for trend analysis and forecasting of energy and occupancy data	Non-Expert developers would like to provide trend analysis and forecasting of energy consumption and occupancy data without having in dept knowledge of statistics and machine learning algorithms,	reusable components for analyzing energy consumption and occupancy are available, and evaluated with developers without statistic & machine learning background.

Key	Requirement Type	Priority	Summary	Rationale	Fit Criterion
IMP-17	Functional	Major	Dynamically adjustable security level for resource constrained devices	The IMPReSS platform should enable developers to design systems where security levels and mechanisms can be adjusted at run-time.	The functionality described is implemented to the IMPReSS platform.
IMP-21	Functional	Nice to have	Graphical model-driven commissioning tool	The platform manager could not be a computer scientist: the platform should be commissioned without writing code.	A model-driven tool with a graphical user interface is available for platform commissioning
IMP-22	Functional	Major	Runtime services/devices discovery and commissioning	It may happen that new devices or new functionalities are added while the platform is already running. The platform should support runtime device and service discovery and commissioning, without platform restarting.	New devices/services are discovered and displayed on commissioning tool as available resources.
IMP-23	Functional	Major	Development toolkit for resources integration	Developers wants to integrate new resources in a fast and simple way	An development toolkit is available for for a rapid model-driven implementation of interfaces for resources integration.
IMP-24	Functional	Major	APIs definition	Users may want to interact with the platform through software applications in order to visualize some data/info/alerts/ recommendations, direct control of some devices or for fine-grained configurations/customizations.	A set of APIs are available for application development in order to interact with the platform.
IMP-25	Functional	Major	IMPReSS architecture views	IMPReSS architecture must offer different views according to different usages and needs, such as: application developer application user, dataflow/control flow	Documentation describing architecture for application user and application developer
IMP-26	Functional	Major	Templates for smart entities	Application developers may be allowed to create new templates for smart entities	Application developer may need create a new template to modeling a new smart entity in application development.

10.2 Non-functional Requirements

Key	Requirement Type	Priority	Summary	Rationale	Fit Criterion
IMP-1	Non-Functional - cultural and political	Blocker	Sensors must be unobtrusive	It is very important that the project show respect of the building. Equipment must fit in the theatre seamlessly.	The application cannot be deployed if the criterion is not med
IMP-2	Non-Functional - operational	Major	Impress' cloud must scale horizontally	Provide the means for the Impress Cloud to scale horizontally, by adding more clusters running Impress' instances.	UFPE.
IMP-9	Non-Functional - operational	Major	The SDP should encapsulate the complexity of different technologies, developing a single logic to devices manipulation.	The creation of single implementation logic for different technologies, help inexperienced developers create specific applications without having in-depth knowledge in different technologies involved.	FIT
IMP-10	Non-Functional - operational	Major	The SDP shall support multiple communication protocols	This requirement is fundamental since there are many different devices using different communication technologies and SDP must support these technologies to allow integration of these devices	UFABC
IMP-11	Non-Functional - maintainability	Major	The software components of the middleware should be modularized, facilitating the inclusion of different technologies with the purpose of integrating heterogeneous resources	Modularization of components facilitates maintenance of middleware, in addition to facilitating integration of different technologies.	CNET

Key	Requirement Type	Priority	Summary	Rationale	Fit Criterion
IMP-14	Non-Functional - operational	Major	The impress core runs on a Gateway that cost below USD50	IMPReSS aims at affordable intelligent system than can be produced within the near future; therefore the price of the required hardware must be affordable.	The core IMPReSS middleware could run on a gateway which cost below USD 50. The core middleware should enable communication among heterogeneous devices and applications.
IMP-18	Non-Functional	Major	The IMPRESS platform should support development of IoT systems that are extendable for future needs	Devices and environments have different life cycles. For instance, mobile phones and laptops last typically couple years, whereas refrigerators and electric ovens have at least a five year lifespan and houses can last hundreds of years. At the design time we are also not able to know all devices and applications that will be part of the system in the future. Therefore, the IMPReSS platform should support development of IoT systems that are extendable for the future needs. In practise this means that the Resource and Application descriptions and all the related components should be based on technologies that are flexible and thus easily extendable for the future.	IMPReSS platforms based on flexible technologies that support development of extendable IoT systems.
IMP-19	Non-Functional	Major	The IMPRESS platform should be agnostic to the application domain	The aim should be to develop a system development platform that can be used in various IoT application domains (e.g. Healthcare, retail, logistics, transports, energy, home automation, etc.).	The platform is based on general purpose technologies and not optimized to a certain application domain.
IMP-20	Non-Functional	Major	The IMPRESS SDP should be easy to use	To make the SDP acceptable by the developers it should be easy to learn and use.	The basics of IoT system development with the SDP can be learned in one day.

Key	Requirement Type	Priority	Summary	Rationale	Fit Criterion
IMP-27	Non-Functional	Major	Data in the IMPRESS network is classified to different categories based on the criticality	Data transmitted in the IMPReSS network needs to be classified to different classes. Data types can be e.g.: -Emergency data such as fire or burglar alarm data -Monitoring data such as temperature monitoring data -Device control messages -Device condition data such as remaining energy	Sensor data can be classified into different classes.
IMP-28	Non-Functional - security	Major	Confidentiality of the messages between IMPRESS platform devices can be guaranteed	Data transmitted between IMPReSS devices can contain confident information e.g. about the house energy consumption. IMPReSS platform needs mechanism for preventing unauthorized access to confident information.	Transmitted data between IMPReSS devices can be correctly interpreted only by authorized devices.
IMP-29	Non-Functional - security	Major	Integrity of the messages between IMPRESS devices can be guaranteed	IMPReSS platform must guarantee the integrity of the critical messages between devices. Critical messages cannot be modified by unauthorized parties.	Critical messages cannot be modified by unauthorized parties.
IMP-30	Non-Functional - security	Major	Availability of the critical IMPRESS devices must be guaranteed	The most critical devices and services needs to be available all the time (for example >99% uptime. TBD). Security mechanisms in IMPReSS platform needs to guarantee the availability of the most critical devices and services.	Most critical devices and services are available (for example. >99% uptime TBD)

Key	Requirement Type	Priority	Summary	Rationale	Fit Criterion
IMP-31	Non-Functional - security	Major		Data transmitted in the IMPReSS network needs to be classified to different classes based on the confidentiality. Data types can be e.g.: -Level 1 confidentiality (such as patient data) -Level 2 confidentiality (such as energy consumption monitoring data) -Level 3 confidentiality (device or network status data)	Confidentiality levels are set for the data