Intermediate deliverable D3.1

EEPOS ICT and decision support systems platform - Specification

Revision 1

Preparation date 2013-08-28 (m11)

Due date 2013-03-31 (m06)

Lead contractor............................. FTM

Authors:

Janne Rasi.................................... FTM

Kaspar Pae .... YIT

Arturs Purvins .... DERlab

Kalevi Piira................................... VTT

Veijo Lappalainen .... VTT

Roland Kopetzky............................ ENO

Florian Judex .... AIT

Table of contents

1. PUBLISHABLE EXECUTIVE SUMMARY

Overall project mission: contribute to increased energy efficiency by neighbourhood management and control systems capable to optimize, in near-real time, the local generation-consumption without compromising the comfort or performance of the buildings and neighbourhoods with the aid of latest ICT developments.

To achieve this goal we need internal tools to ease connectivity between EEPOS internal components and to connect to external services and datasources. Automation platform (WP2) offers solutions to connect to sensors and automation devices. But for the EEPOS to be truly useful for customers we need to integrate to ICT solutions like SAP and other enterprise resource planning solutions, ICT platform will provide necessary means to achieve this connectivity.

ICT platform works as a mediator between EEPOS tools, NEMS and external services. It offers easy way for EEPOS tools to connect to different data sources. The overall system diagram in *Figure 1* shows how the ICT platform will be structured.

Figure 1. System diagram of the ICT platform
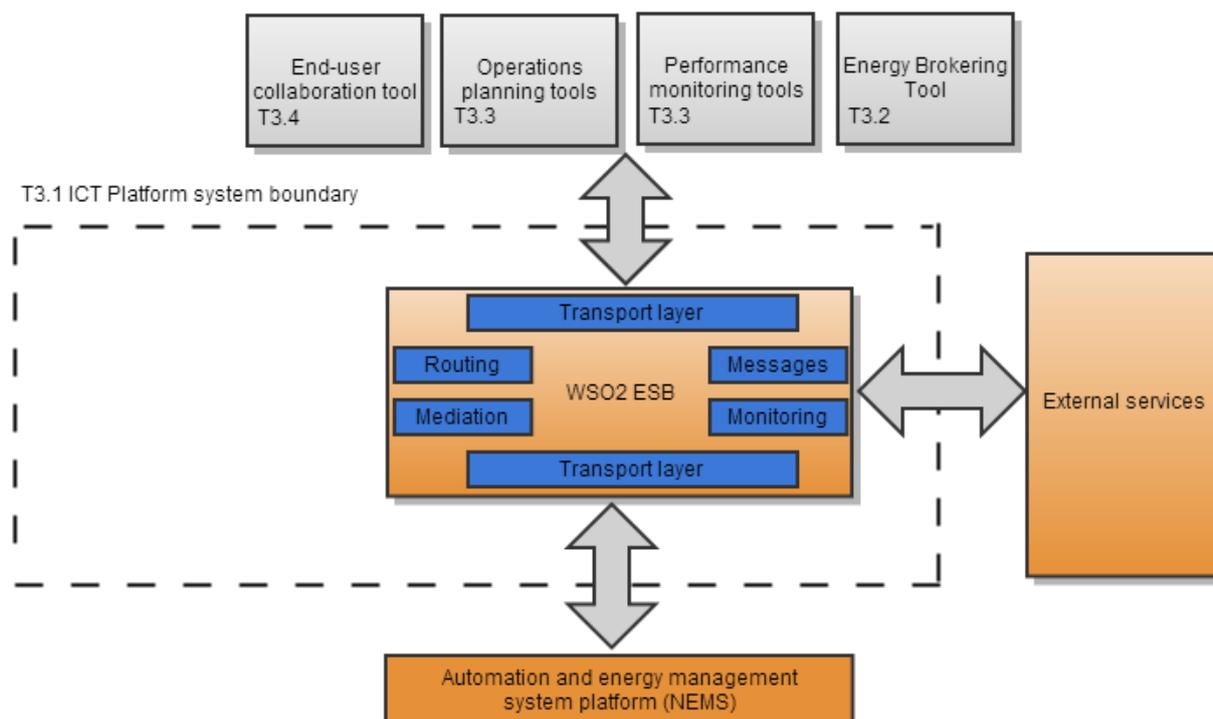
With this specification the project goals can be achieved with the given resources which don't allow the implementation from the scratch. The basic platform component is 100% open source Enterprise Service Bus (ESB) from the company WSO2. It was a natural choice because the competing commercial products would have been very expensive and most likely can't offer any more functionality.

2. INTRODUCTION

2.1 Purpose and target group

This report will define the ICT platform that will be implemented later in this task. Target group is the people who will be doing the actual implementation and also partners from different task that are connected to this one.

Other tasks needs and requirements will be addressed in this report.

2.2 Contributions of partners

Fatman Oy (FTM) will be the lead contractor and responsible for this report. Other partners will provide information for the use cases, comments and feedback.

Section / Chapter	Contributor
1, 2, 3, 4, 5, 6, 7	FTM
4.3.6, 7	VTT
4.3.3	DERlab
4.3.2	ENO
4.3.1	YIT
4.3.5	AIT

Table 1. Contributions of partners

2.3 Scope

Creating a complex ICT platform can be very long process and since we have limited resources we have to focus on the needs of the demonstrations and take some shortcuts regarding security and other features. Left out features will be addressed in the specification briefly, so it's possible to add these features later.

The ICT platform will provide interfaces for connecting to EEPOS internal interfaces and external third party systems. Platform will not provide any capabilities for data processing, this has to be done in external software (T3.2, T3.3 and T3.4 for example).

Creating the platform from scratch would have been a too big task and a risk regarding the task resources. That is the reason we will be using WSO2 ESB (http://wso2.com/products/enterprise-service-bus/) as the ICT platform.

2.4 Relations to other activities

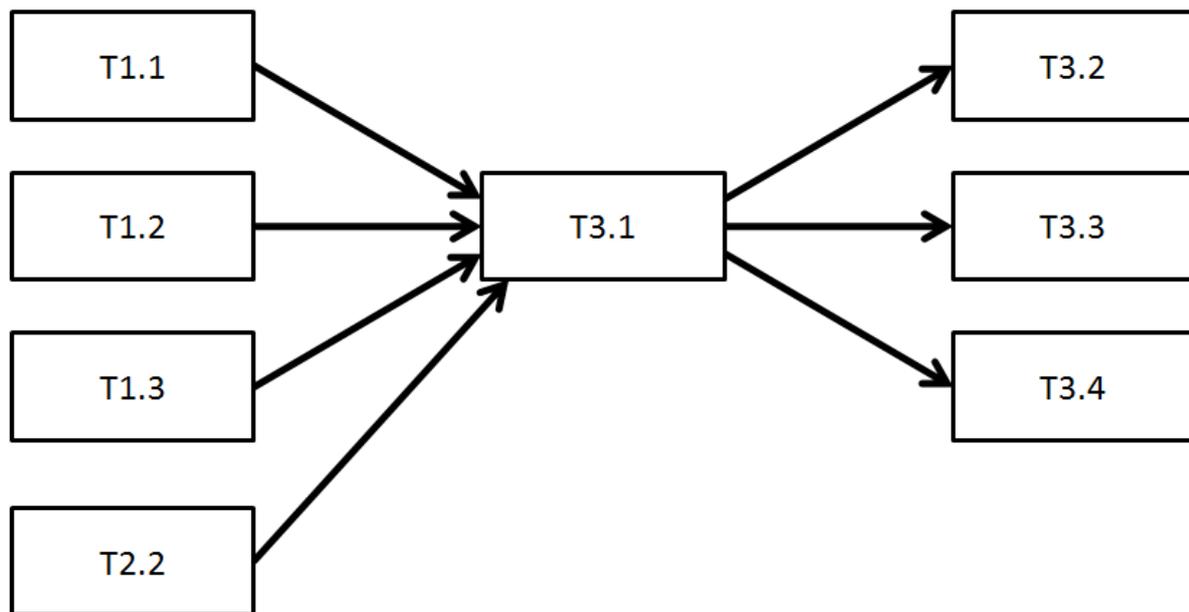*Figure 3.* Relations to other activities

This specification will be based on the information we get from the WP1 tasks (T1.1, T1.2 and T1.3).

The task using the platform later will provide their requirements as use cases (WP3 tasks T3.2, T3.3 and T3.4).

Use cases specified in chapter 4.3 collect requirements and feature requests from different demonstrations and tasks that will be using the ICT platform.

3. ENTERPRISE SERVICE BUS (ESB)

Enterprise service bus (ESB) is a software architecture construct, which lives between the (business) applications and enables communication among them. Ideally, the ESB should be able to replace all direct contact with the applications on the bus, so that all communication takes place via the ESB.

ESB provides its fundamental services via an event-driven and standards-based messaging engine (the bus). Thanks to ESB, integration architects can exploit the value of messaging without writing code. Developers typically implement an ESB using technologies found in a category of middleware infrastructure products, usually based on recognized standards.

ESB is a relatively new concept. The term has only been in common use for the last several years. People often associate ESBs and Service Oriented Architecture (SOAs) but they are somewhat orthogonal concerns. However, like SOA, ESB is essentially a collection of enterprise architecture design patterns that is now implemented directly by many enterprise software products.

The WSO2 ESB is a fast, light-weight and versatile Enterprise Service Bus product released under the Apache License v2.0. Using WSO2 ESB you can filter, transform, route and manipulate SOAP, binary, plain XML, and text messages that pass through your business systems by HTTP, HTTPS, JMS, mail, etc. as *Figure 3* illustrates.

ESB will be the foundation of the ICT platform. Using ESB functionalities we can configure a working setup that will support our EEPOS platform and help us connect different parts together. [1]

Figure 3. ESB Concept

(http://docs.wso2.org/wiki/display/ESB460/About+ESB)
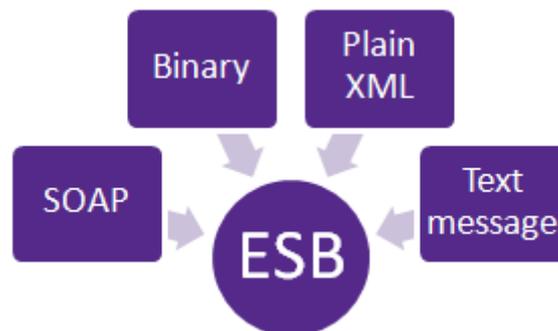
WSO2 ESB is released under Apache Software License Version 2.0, one of the most business-friendly licenses available today. (http://www.apache.org/licenses/LICENSE-2.0)

3.1 ESB Architecture

Application infrastructure on the enterprises may be inherently complex, comprising of hundreds of applications with completely different semantics. Some of these applications are custom built, some are acquired from third parties and some can be a combination of both and they can be operating in different system environments.

Integration among these heterogeneous applications is vital to the enterprise. Different services may be using different data formats and communication protocols. Physical locations of services can change arbitrarily. All these constraints mean your applications are still tightly coupled together.

An ESB can be used for loosen up these couplings between different services and service consumers

A diagram below describes WSO2 ESB architecture from a messaging perspective. There are lots of other areas like Tasks, Events which are not described in this diagram. It shows how messaging is handled in the ESB.
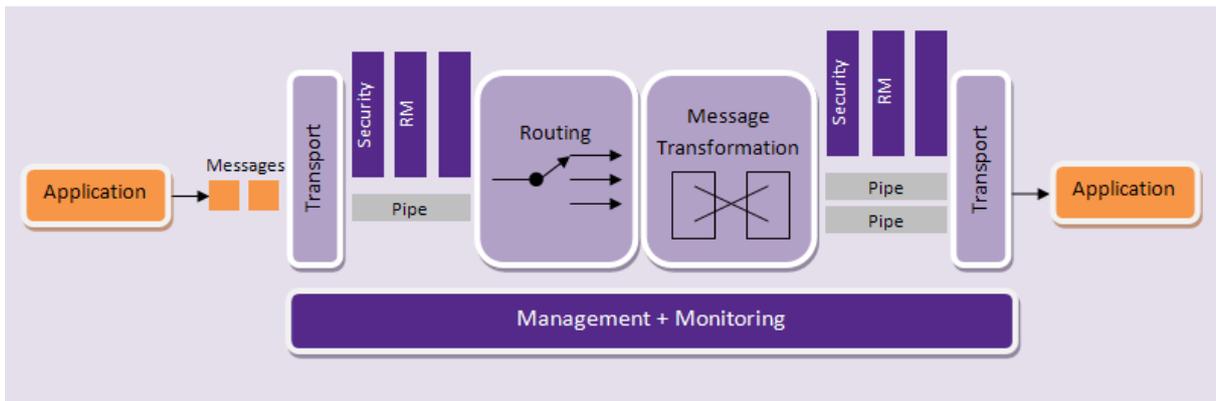

Figure 4. ESB Architecture

1. An application sends a message to the ESB.

2. The message is picked up by the ESB transport.

3. Transport sends the message through a message pipe. Quality of service aspects like Security and Reliable Messaging of the message is taken care in this pipe. Internally this pipe is the in-flow and out-flow of Axis2 (Apache Axis2, core engine for the ESB). ESB can operate in two modes:

- Message Mediation - A single pipe is used.
- Proxy Services - Separate pipes connecting transport to different Proxy Services are used.

4. Both message transformation and routing can be seen as a single unit. As the diagram specifies, there is no clear separation between message transformation components and routing components. WSO2 ESB calls this the mediation framework. Some transformations happen before routing decision has taken. Some transformations happen after the routing decision has taken. This part is the Synapse implementation.

5. After this message is injected to the separate pipes depending on the destinations. Here again quality of service aspects of the messages is determined.

6. The transport layer takes care of the transport protocol transformations required by the ESB.

The diagram shows how a request propagates to its actual endpoint through the ESB using its architecture. Response handling is the reverse of this operation.

All these components can be managed and monitored through WSO2 ESB management console. [2]

(http://docs.wso2.org/wiki/display/ESB460/ESB+Architecture)

3.2 ESB Installation prequisitions

ESB needs JAVA environment to run. It can be installed to multiple operating systems including Windows, Linux, Solaris, Mac OS X or anything that can run JDK 1.6.x runtime.

Our installation will be Linux based. Specific distribution will be chosen later when we start the implementing phase. Current test setup is using Ubuntu 64-bit. [3]

Detailed system requirements can be found at:

http://docs.wso2.org/wiki/display/ESB460/Installation+Prerequisites

3.3 ESB Functions

Demonstrator focused ICT systems architecture is specified covering neighbourhood, building and end users ICTs including internal and external data communication interfaces.

3.4 User Characteristics

Users of the ICT platform will be all the partners of the EEPOS project. Everyone will use the platform in their own demonstrations and tasks. ICT platform will offer interoperability for the project and make it possible to easily build an ecosystem from all the different parts.
All the requirements are coming from the use cases described in the chapter 4.3 by EEPOS partners.

3.5 General Constraints

Possibilities for the platform are clearly very optimistic but for project resource reasons we have to keep our scope very focused on the needs of the tasks and demonstrations. Also ICT platform task will not define the data models used between different parts of the EEPOS architecture, task 2.2 will be responsible for that.

4. SPECIFIC REQUIREMENTS

This section will specify the ICT platforms functions and interfaces for the implementation phase giving as much as possible detailed information. Most of the information will be derived from the use cases involved partners have specified. However the information is in different formats and some are more detailed than others. More discussions with the other tasks are necessary before it can be successfully implemented.

4.1 Interface Requirements

4.1.1 User Interfaces

ESB offers user interfaces for defining the services and monitoring them. No new interfaces will be implemented for the ICT platform.

4.1.2 Hardware Interfaces

ICT platform will be installed on a virtual Linux server. The only hardware interface will be Ethernet port communicating with TCP/IP. No serial ports or propriety interfaces will be used.

4.1.3 Software Interfaces

List of the different interfaces needed (derived from the use cases) is shown in *Table 2* below. They will be specified in more detail later during the implementation phase as we get more information and specifications from the other tasks.

Interface	Type	Needed for
Automation platform (ogema)	n/a	T2.1, T3.3
Automation platform (jace)	n/a	T2.1, T3.2, T5.1
BIM model server	n/a	T3.4
Energy prices	n/a	T3.2, T3.3, T3.4, T4.2
Weather	n/a	T3.2, T3.3, T3.4, T4.2
Weather forecast	n/a	T3.2, T3.3, T3.4, T4.2

Table 2. List of the interfaces derived from the use cases

4.1.4 Communications Interfaces

ICT platform can send emails and SMS for alarms and notifications.

4.2 Functional Requirements

Main function of the ICT platform will be mediating messages between different tools created during the project, automation platform and specified outside information sources.

Specific requirements will be updated later as the implementation starts and we know better the services and interfaces we are connecting to.

4.3 Requirements from EEPOS tools

4.3.1 Use Case #1: Demonstration in Finland

For Finnish demonstration the EEPOS platform will be installed into a newly built residential area. This area will be marketed as an ecological and low-energy neighbourhood. From YIT point of view, we aim to build good living environments for the residents, thus our need for the EEPOS platform will revolve around making people's lives easier. This also includes establishing lean-procedures for service and maintenance operations. The ICT platform helps us achieve this goal and make it also possible to integrate to our enterprise resource planning (ERP) systems.

The EEPOS platforms End-user collaboration tool should enable the neighbourhood form a collective community where the all participants are able to contribute to high energy efficiency, environmental way of living and maximum resource utilization. This means that the end-user collaboration tool could connect all residents with the system through a kind of 'forum square /market place' where the achievements of residents independently or collectively can be highlighted and discussed. It could provide a certain backbone for decision making regarding renovation, service and maintenance, refurbishment or service procurement issues, as well. There might be a certain kind of ranking competition listing the top 10% performers.

As a neighbourhood management system, this system should also provide capabilities for incorporating utility billing functionalities and act as a central database for important information regarding the neighbourhood and its principals, systems, agreements etc. It might offer a neighbourhood calendar function for people to mark up relevant events, plan maintenance operations etc.

For the professional personnel, the ICT platform should offer through the performance monitoring and operation planning tool the access to the neighbourhood operations efficiency data as well as the service and maintenance information regarding building and neighbourhood level technical systems. The system should provide the historical and real time energy consumption and other relevant operational data as well as standard analytic functionalities so that issues that need attention would become clearly visible without much further effort by the servicing personnel.

4.3.2 Use Case #2: Demonstration in Germany

For the German Demonstrator the housing cooperative Bauverein Langenfeld eG (BVL) provided four of its residential buildings. These buildings are equipped with a BEMS based on the Ennovatis Smartbox technology which will be enhanced with additional functionalities and services to fit the EEPOS needs. There will be no direct communication between the BEMS and the ICT platform. The BEMS supports the management of the local grid (connected devices, energy producers, ...), communicates with the NEMS and uses the offered services. The NEMS (based on OGEMA) communicates with the ICT Platform as described

in chapter 4.3.3. The following figure shows the general structure of the communication levels in the German demonstrator:
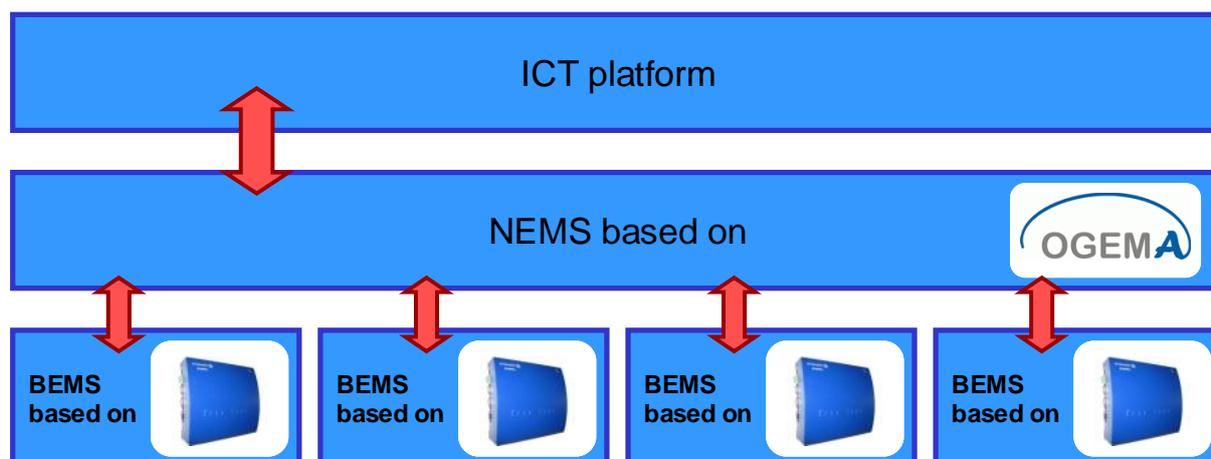

Figure 5. General structure of communication levels in German demonstrator

In addition to the general services and functions offered by the ICT platform we need the following data and services to determine the immediate and future optimum actions:

Service/Data	Source	Description
customer preferences (continuously)	End-user collaboration tool	preferences for comfort conditions (heating, cooling, ...), sell & buy borders, ...
price signals / forecast (at least 24h forecast, updated continuously)	electricity company / energy broker	used to calculate current optimum and to find out when peaks in demand are expected
weather conditions and weather forecasts (at least 24h forecast, updated continuously)	meteorological office (e.g. DWD)	predict future energy production (in case of local PV and Wind Turbine) and energy needs

Table 3. Summary of the services needed for german demonstration

4.3.3 Use Case #3: Laboratory prototypes

Data communication between NEMS and ICT platforms is required in order (i) to provide NEMS with the needed data for energy management in the neighbourhood and (ii) to provide ICT with a feedback on neighbourhood energy management results and on measured energy consumption/production values.

Addressing the first point, NEMS requires predicted electricity price profile, weather forecast (wind and solar irradiance) profile and demand forecast profile for neighbourhood (further in the text prediction profiles), e.g., for 24 hours in ahead. Price profile is used as an instrument for neighbourhood energy management and should be updated frequently in order to reflect the current situation in electricity market. Weather and neighbourhood demand forecast profiles are additional input data required for efficient energy management in neighbourhood.

Addressing the second point, the feedback to ICT is required in order to inform ICT how efficient was the price profile in load shifting.

Name	**Required data communication NEMS-ICT**
Related scenario	All scenarios in T1.1
Pre-conditions	• NEMS based on OGEMA is installed on embedded computer • NEMS is connected to data communication network • Data communication between NEMS and ICT is provided
Stakeholders	NEMS operator and ICT operator
Systems involved	OGEMA, ICT platform
Trigger	NEMS is running. NEMS is waiting for prediction profiles. Once at least one of prediction profiles is received, NEMS executes one management cycle.
Basic path	After NEMS receives at least one updated prediction profile, it runs one neighbourhood management cycle in order to provide ICT with updated feedback on price influence on load shifting. NEMS executes neighbourhood management algorithm using the updated profile/s. When NEMS receives feedback information from end-energy users/producers, it sends this feedback to ICT. After that NEMS is waiting again for next updated prediction profile and cycle repeats.
Post-condition	Neighbourhood management is performed according the updated prediction profiles
Exception-path	NEMS operator may involve in NEMS management through HCI. In this case feedback to ICT will include also human factor.
Priority	High

Table 4. Summary of the services needed for NEMS-ICT

4.3.4 Use Case #4: Energy Brokering Tool

Energy Brokering Tool (EBT) uses ICT platform resources through web interface. Needed resources include sensors and meters from Finnish Demonstration (T5.1) through Automation Platform (T2.1) using Ogema or Jace as a platform.

Additional information needed from weather service (FMI in Finland) and energy prices from Nordpool (Finland). Interfaces those services uses will be researched and implemented if possible. Also the total energy consumption information from energy companies is needed.
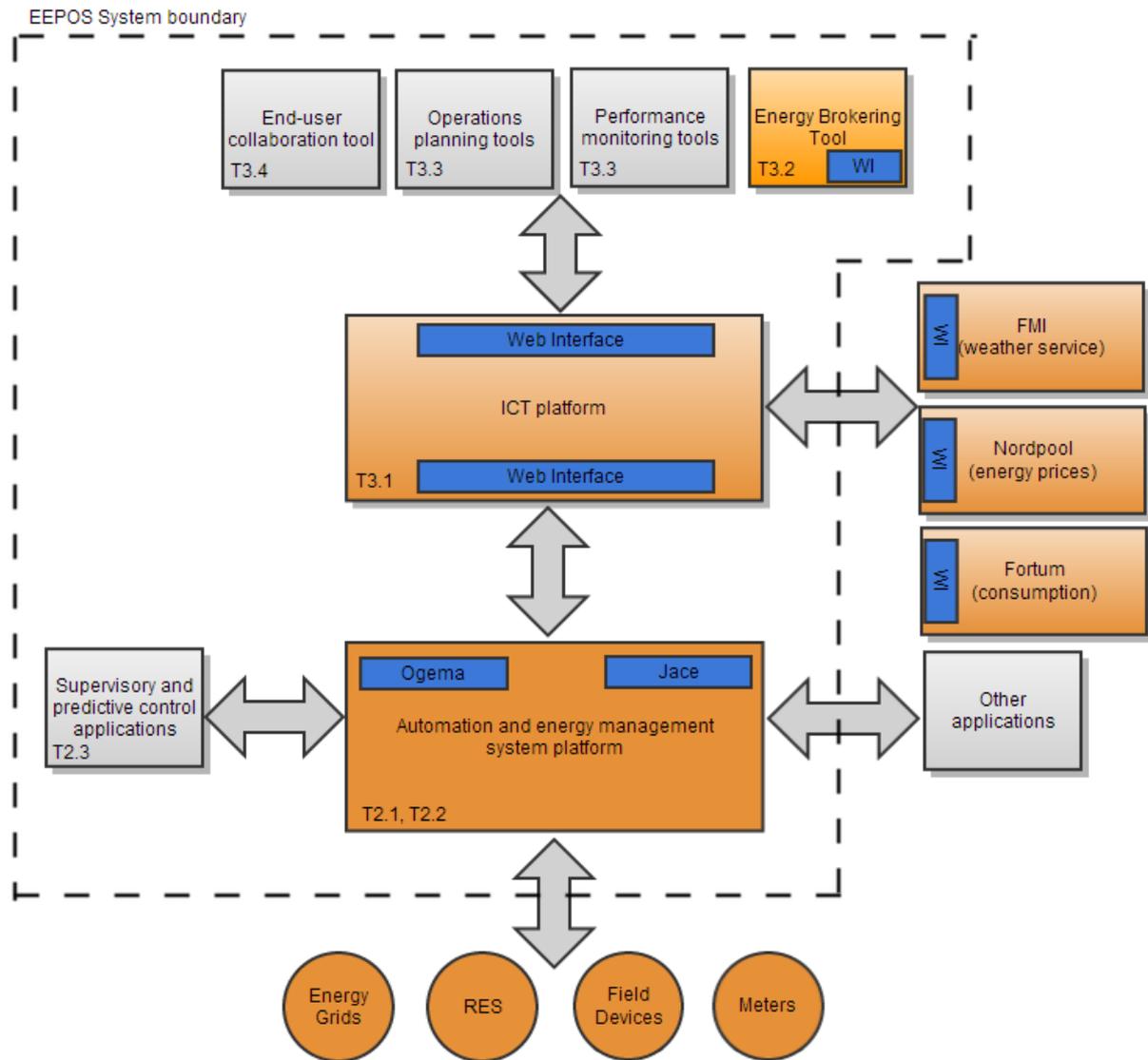
Figure 6. Energy Brokering Tool diagram

Service	What is needed	Interface
Ogema	Sensors, Meters	Ogema API
Jace	Sensors, Meters (if not possible to connect Jace through Oogema)	oBIX
Weather	Weather information (FMI or other provider)	Web Service (SOAP)
Energy Prices	Energy prices (Nordpool or other provider)	?
Energy Consumption	Energy consumption (Fortum or other provider)	?

Table 5. Summary of the services needed for EBT

4.3.5 Use Case #5: Performance monitoring and operations planning tools

Both the performance monitoring and the operational planning tools will have to connect to the automation and energy management system platform. For proper performance monitoring, the energy consumption data has to be connected to the variables which are controlled by the HVAC components in the individual buildings. Therefore the performance monitoring needs read access to this data. Furthermore, auxiliary data (weather, prices) have to be taken into account as well, which have to be accessed also through the ICT platform. To be able to assess the impact of the other modules on the performance, their influence on the actual consumption should be included as well. It can be concluded that the performance monitoring tool needs read access to almost of the systems components. On the other hand, the end user tool and the operation planning tool may need access to the performance monitoring tool, information included there is needed, in order to reduce duplicated functionality.

The operational planning tool will need access to the supervisory and predictive control applications, which should relay the desired operation to the systems. Furthermore, it should be able to communicate with the energy brokering tool which may give input, as well as the auxiliary data (weather and prices), in case the energy brokering tool either is not present or has no preferences. Furthermore, it should access the performance of the components through the monitoring tool.

4.3.6 Use Case #6: End-user collaboration tool

End-user collaboration tool (ECT) uses ICT platform resources through web service interface shown in *Figure 9*. Needed resources includes

- Apartment level consumption (electricity, heat, gas, water) data.

- Apartment level consumption (electricity, heat, gas, water) data related to other same kind of apartments (reference data).

- Apartment level load shifting data and related costs savings.

- Apartment level load shifting data and related costs savings related to other same kind of apartments (reference data).

- Energy demand and price forecasts.

- Weather data and weather forecast.

- RES related energy production and related impact on CO_2 emissions.

- Apartment level purchased, generated (apartment's RES) and sold energy and related cost savings and impact on CO_2 emissions.

- Apartment level purchased, generated (apartment's RES) and sold energy and related cost savings and impact on CO_2 emissions related to other same kind of apartments (reference data).

- Other available apartment level BAS measurements like inside temperatures, relative humidity, CO_2, etc.

- Apartment level average inside temperatures, relative humidity and CO_2 related to other same kind of apartments (reference data).

- Easy and secure access to change apartment energy saving mode or set points (for example 1 or 2 hours) in the case of neighbourhood level energy sale.
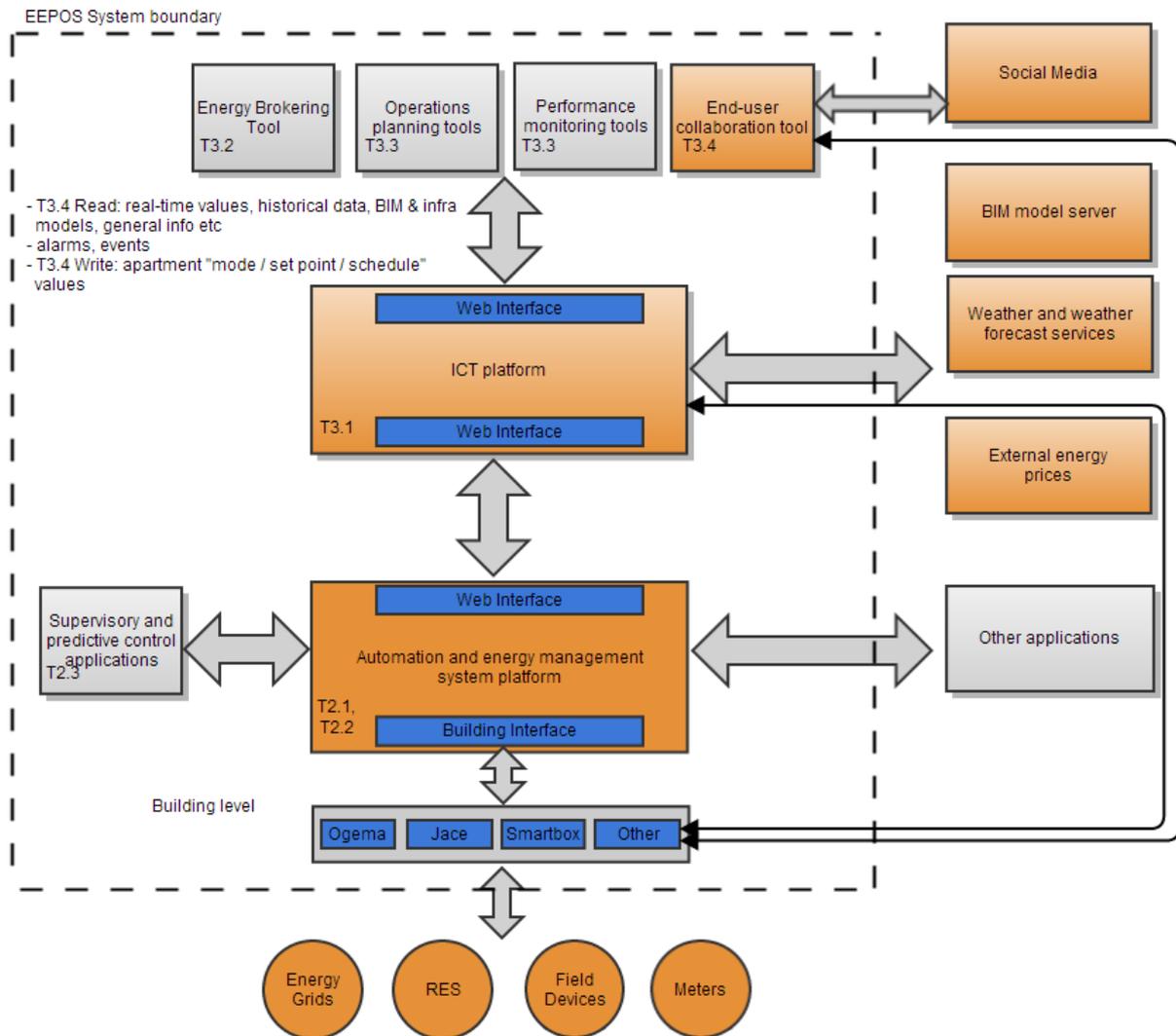
Figure 7. End-user collaboration tool diagram

ECT will utilize also some later selectable well known social media platform like Facebook Groups, Twitter, Google+ Pages, LinkedIn Groups or some more suitable solution for supporting collaboration purposes. Additional information needed from BIM model server, infrastructure models, weather and weather forecast service (FMI in Finland) and energy prices from Nordpool (Finland). The application will most likely utilize also some game engine, GPS information and QR code related links to EEPOS web service interface for EEPOS database information.

4.4 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level.

4.4.1 Performance

The ICT platform will be run as a pilot / demonstration, it should not be considered as production environment. Performance should be in a level that all the aspects of the project

can be demonstrated without problems. Specific details will be updated once the implementation starts and performance can be tested.

4.4.2 Reliability

As we are not building an production environment, there is no need for redundant servers and components. However current planned system infrastructure where the ESB will be installed is built for production servers.

4.4.3 Availability

Service will be ran as best effort model. It will be monitored and maintained during working hours but it will not have the priorities and SLA of an production site.

4.4.4 Security

Site will be published to public web using SSL-certificates (https). Also user interface will require authentication. Any firewall restrictions won't be used as it would be unnecessary layer of security in addition to authentication and SSL.

4.4.5 Maintainability

Planning first integrations with ESB carefully and testing thoroughly, we can ease the job on the future integrations. ESB system is built that you can re-use end-point implementations and message mediators. Most of the work is done within graphical user interface, so the developer doing the actual integrating work doesn't have to know Linux operating system or go deep into scripting languages.

4.5 Design Constraints

Features that are not possible to implement using WSO2 ESB will be discussed within the task group: if there is something really needed in the project and if it's possible to implement that in some other way.

4.6 Other Requirements

This chapter will include a list of any other requirements and it will be a catchall section for any additional requirements. It will be updated later during the implementation phase.

5. CONCLUSIONS AND FUTURE WORK

The EEPOS ICT platform will enable the EEPOS neighbourhood energy management platform to easily connect to required data sources. This makes integration to current systems already in use and offers great support for future applications. EEPOS tools (T3.2-T3.4) demonstrate the possibilities and defines which kind of functions and features ICT platform needs to fulfill this support. For this platform to be a success we need to prioritize and focus on the following

- Flexible integration interfaces
- Efficient backend services
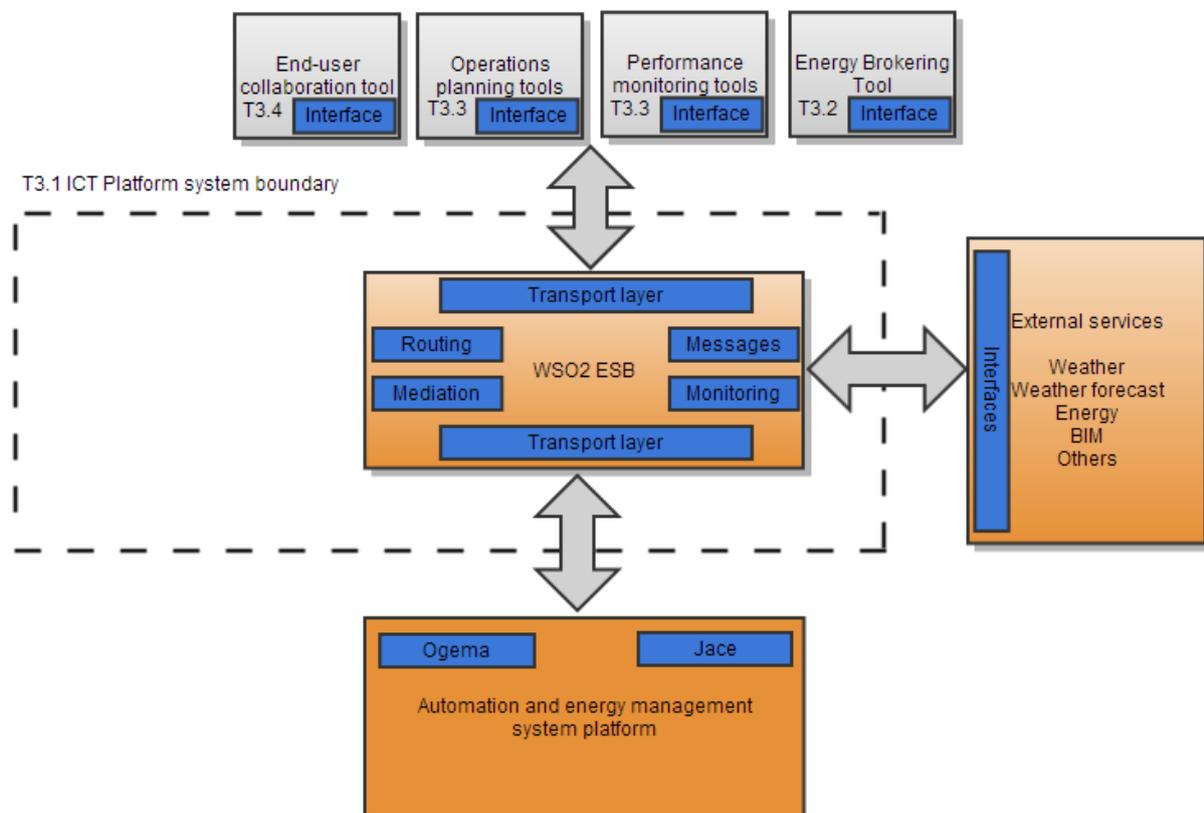- Performance/Quality monitoring of the platform

Figure 8. ICT platform system diagram

As *Figure 8* shows the ICT platform works as a mediator between applications, automation platform and external services. It offers an easy way for EEPOS tools to connect to different data sources.

As stated in DoW objectives of this work package for ICT platform are:

"Specification of interoperable standard based ICT systems architecture and platform covering neighbourhood, building and end users ICT's as well as their internal and external data communication interfaces"

With this specification we can achieve our goals with the given resources. Choosing 100% open source WSO2 ESB was a natural choice. Competing commercial products would be very expensive and most likely can't offer any more functionality.

5.1 Summary of achievements

- EEPOS ICT platform role and main function identified
- Suitable tools tested and chosen for the implementation phase – WSO2 ESB
- External datasource need identified
- Main functions and requirements from EEPOS tools for the ICT platform specified

5.2 Future Work

The tentative schedule for the further work of Task T3.1 is shown in Table 5. The plan will be updated during the project.

Action	Time	Misc
Specification	Jan 2013 – Mar 2013 (m4-m6)	D3.1 Deliverable - Specification
Platform testing and evaluation	Feb 2013 – Aug 2013 (m5-m11)	Testing capabilities of the platform
Implementation	Jul 2013– Dec 2013 (m10-m15)	Implementing required interfaces with the platform
Testing	Nov 2013 – Jan 2014 (m14-m16)	Testing functionalities with other tasks using them
Documentation	Oct 2013 – Feb 2014 (m13 – m17)	D3.1 Deliverable – Technical Documentation
Task completed	Mar 2014 (m18)	D3.1 Specification D3.1 Technical Documentation D3.1 Prototype

Table 6. Task schedule

We must follow closely the progress of the related other tasks to complete our own tasks according to schedule. Other tasks will lean on this task to provide them with the necessary interfaces and data sources. Interfaces must be done in cooperation with other tasks. T3.1 resources only suffice for the ICT platform's side of the interface, the other tasks must implement their task specific interfaces using their own resources.

6. CHANGE MANAGEMENT PROCESS

This document is a first step towards implementation of the ICT platform. It will be updated during the process. We have to identify roles of the different tasks involved and to create a review process for changes regarding interfaces. The interface needs always two sides and all changes must be done unanimously.

This document will be updated to contain starting point information about interfaces. At the moment we don't know details or types of the interfaces. Actual description how the interfaces are implemented are left for the D3.1 – Technical Documentation deliverable due month 18 (March 2014).

Needs might raise during the implementation of the WP3 tools using the ICT platform for new interfaces not specified in this document. If the new interfaces can then be justified, this specification will be updated to cover also those new interfaces.

7. ACRONYMS AND TERMS

BEMS	Building Energy Management System
BIM	Building Information Model
DoW	Description of Work
ECT	End-user collaboration tool
ERP	Enterprise resource planning
ESB	Enterprise Service Bus
FMI	Finnish Meteorological Institute
GPS	Global Positioning System
ICT	Information and Communication Technology
OGEMA	Open Gateway Energy Management Alliance
QR code	Quick Response code
NEMS	Neighbourhood Energy Management System
RES	Renewable Energy Source
SLA	Service Level Agreement
SMS	Short Message Service
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
XML	Extensible Markup Language
WSO2	Company name (www.wso2.com)

8. REFERENCES

[1] http://docs.wso2.org/wiki/display/ESB460/About+ESB

[2] http://docs.wso2.org/wiki/display/ESB460/ESB+Architecture

[3] http://docs.wso2.org/wiki/display/ESB460/Installation+Prerequisites

9. APPENDICES

Additional information is described in related background documents:

Appendix 1: Validation strategy and application scenarios (D1.1; publicly available on the project website).

Appendix 2: Stakeholder requirements (D1.2; publicly available on the project website).

Appendix 3: Overall architecture specification (D1.3; publicly available on the project website).