



## FI-ICT-2011-285135 FINSENY

### D4.3

#### Smart Building Functional Architecture

**Contractual Date of Delivery to the CEC:** 31.03.2013 (Month 24)

**Actual Date of Delivery to the CEC:**

**Author(s):** see list below

**Participant(s):** ABB, Acciona, EDF, Engineering, Grenoble-INP, Orange Labs, Synelixis, TID, Telecom Italia

**Workpackage:** WP4 Smart Buildings

**Security:** PU=Restricted to a group specified by the consortium (including the Commission Services), here FI-PPP projects

**Nature:** (R = Report)

**Version:** 1.0

**Total number of pages:** 160

#### Abstract:

This deliverable presents the final description of the functional architecture for the Smart Buildings domain. This architecture is viewed in the FI-PPP way as a common denominator shared infrastructure, corresponding to the functional and information layers of the Smart Grid Architecture Model and aimed at supporting all smart building ICT applications. Instances of building energy management systems that operate the building as a smart grid endpoint are presented as the intended lead users of this infrastructure. This document describes the decomposition of this architecture into finer-grain functional building blocks and how they are mapped to FI-WARE. Communication and component layers in building domain of the SGAM model are further described.

#### Keyword list:

Building Energy Management System, SGAM, Functional Architecture, Smart Grid, Microgrid, Smart Energy, FI-WARE, Generic Enablers, Building Infrastructure, Building Operating System

## Executive Summary

We present in this deliverable a final description of a Smart Building ICT architecture aimed at supporting advanced energy management as well as other smart building applications, across different building domains comprising homes/stand-alone houses, residential apartment buildings, office/public buildings, data centers and hotels. To elaborate this architecture, we started from a drill-down of a select few relevant high-level and detailed use cases previously identified that delineate the functionality we are aiming for the designed system.

We complement this top-down analysis with a complementary approach starting from a general vision of a layered Building Operating System that is aimed at supporting all smart building applications, much like the overall FI-WARE platform itself, only at the building scale.

This architecture framework is then refined into a more fine-grain decomposition as a set of functional building blocks linked with the results of the top-down use case functional breakdown and with the generic enablers provided by the FI-WARE project.

The application layer is described as a set of alternative or complementary solutions for home or building energy management and their relationships with the lower layers of the architecture.

The Energy@Home solution is a step towards the development of the home as the end point of the smart grid that, in the future, will allow continuous real-time two-way information exchange between utilities and appliances in the houses to enable customer to “self-manage” their energy behaviors according to power supply and prices. From an application point of view, Energy@Home envisions a system that can provide users with information on their household consumption directly on the display of the appliance itself, on the smart phone or on their computer. It is expected that, through easy access to information on consumption and through the possibility of downloading custom applications, consumers will be able to use their appliances in a “smart” way by enhancing the energy efficiency of the entire house system. For instance, Smart Appliances can start functioning at non-peak (and therefore less expensive) times of day as well as they can cooperate to avoid overloads by automatically balancing consumption without jeopardizing the proper execution of cycles.

The BeyWatch energy management applications uses an Appliances Management layer (on a par with entity management in the FINSENY framework) allows the management of appliances independently from both manufacturers and communication technologies. The BeyWatch agent is in charge of controlling and monitoring the appliances, using the Appliance’s APIs, which provide simplicity and consistency and a data model for each type of appliance along with the methods of data interchange. This agent connects with the manufacturers which have the control of the appliances, and schedules the operation of the devices. One of the objectives is to allow different manufacturers to connect to the HAN, for that it is used OSGi technology that provides a smooth integration of the different modules composing the framework.

The ReActivHome solution is a comprehensive home energy management system that uses a hybrid optimization engine (with a mixed analytical multi-agent-system based approach) to optimize the allocation of resources corresponding to virtual entities in the services layer.

By mapping the resulting units (functional building blocks) of the top-down functional breakdown onto components and layers of the proposed architecture we demonstrate the relevance of the design. By mapping elements of the proposed architecture onto FI-WARE enablers we show concrete ways to leverage the infrastructure provided by FI-WARE.

We complete this functional architecture description with a more complete specification of the corresponding lower Communication and Component SGAM layers

**Authors**

<b>Partner</b>	<b>Name</b>	<b>Phone /e-mail</b>
<b>Orange Labs</b>	<b>Gilles Privat (Editor)</b>	Phone: +33 476764330 e-mail: gilles.privat@orange.com
<b>ABB</b>	<b>Dirk John</b>	Phone: +49 6203 71 6281 e-mail: dirk.john@de.abb.com
<b>Acciona</b>	<b>Rafael Socorro</b>	Phone: + e-mail: rafaelclaret.socorro.hernandez@acciona.com
<b>EDF</b>	<b>Yves Dherbecourt</b>	Phone: +33 1 47 65 37 90 e-mail: yves.dherbecourt@edf.fr
<b>Grenoble INP</b>	<b>Didier Boëda, Stephane Ploix</b>	Phone: +33 4 76 82 62 92 e-mail: didier.boeda@g2elab.grenoble-inp.fr
<b>Synelixis</b>	<b>Fotis Chatzipapadopoulos</b>	Phone: +30 22210 61309 e-mail: fhatz@synelixis.com
<b>Synelixis</b>	<b>Eleftherios Lefkolikos</b>	Phone: +30 22210 61309 e-mail: elefkolikos@synelixis.com

**Synelixis                      Menelaos Perdikeas**

Phone: +30 22210 61309

e-mail: mperdikeas@synelixis.com

**Telecom Italia S.p.A.      Valter Bella**

Phone: +39 011 228 5643

e-mail: valter.bella@telecomitalia.it

**TID                                      Javier Lucio**

Phone: +34 914832756

e-mail: lucio@tid.es

## Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>9</b>
<b>2</b>	<b>Methodology.....</b>	<b>10</b>
2.1	The SGAM framework .....	10
2.2	Approach for this deliverable .....	12
2.3	Analysis Steps.....	12
<b>3</b>	<b>Use cases and ICT requirements of the Smart Buildings domain .....</b>	<b>14</b>
<b>4</b>	<b>Smart Building Functional Architecture High-level Building Blocks ...</b>	<b>19</b>
4.1	ICT Perimeter of the Smart Building and external interfaces.....	19
4.1.1	Interfaces to Energy marketplace.....	19
4.1.2	Interfaces to External applications .....	21
4.1.3	Interfaces to Microgrid management system & distribution network.....	22
4.2	Overall view of the functional architecture .....	24
4.2.1	Building Operating System.....	24
4.2.2	Generic functional building blocks .....	26
4.3	Application layer .....	28
4.3.1	Energy@home Application Layer .....	29
4.3.1.1	Control modes.....	29
4.3.1.2	Start-up and discovery .....	30
4.3.1.3	Customer Awareness .....	30
4.3.1.4	Appliance regulation.....	32
4.3.1.5	E@H control disabled.....	32
4.3.1.6	E@H control enabled.....	35
4.3.1.7	Self-Production and Primary meters in Energy@home application layer .	38
4.3.2	BeyWatch Application Layer.....	41
4.3.2.1	Appliance Management Framework.....	41
4.3.2.2	Application layer main characteristics in BeyWatch.....	44
4.3.2.3	Application layer design principles in BeyWatch.....	44
4.3.3	Data Center application layer.....	45
4.3.3.1	Data Center Infrastructure Management (DCIM).....	45
4.3.3.2	Data Center Application Management components .....	46
4.3.3.3	Open Data Center Infrastructure Management application .....	48
4.3.4	ReActivHome project application layer.....	48

---

4.3.4.1	Principle of control mechanism .....	49
4.3.4.2	Principle of regular/centralized solving approach .....	51
4.3.4.3	Principle of mixed solving approach .....	53
4.4	Shared services layer .....	60
4.4.1	Virtual entity services .....	61
4.4.2	Building services in the ReActivHome project.....	62
4.4.3	Building state maintainer .....	64
4.4.4	Historization.....	65
4.4.5	Supervisory control services .....	66
4.4.6	User interface services .....	67
4.5	Building Entity Abstraction Layer.....	68
4.5.1	Models for target entities .....	69
4.5.2	Identifying the target subsystems.....	69
4.5.3	Defining the relevant state of target subsystems.....	69
4.5.4	Self-configuration/reconfiguration.....	70
4.5.5	Interface to services/applications .....	70
4.5.6	REST interface.....	70
4.6	Sensor & actuator Interface layer .....	70
4.6.1	TEDS interface layers .....	71
4.6.1.1	TEDS model .....	72
4.6.1.2	TEDS applications through networking .....	72
4.6.1.3	TEDS standard interfaces .....	73
4.6.2	AS-Interface .....	76
4.6.3	EEBUS interface layers .....	78
4.6.4	WSAN interface layers .....	81
<b>5</b>	<b>Mapping onto FI-WARE Generic Enablers .....</b>	<b>87</b>
5.1	FI-WARE Framework .....	87
5.2	Description of FI-WARE Generic Enablers .....	88
5.3	Mapping Smart Building High-level Building Blocks to FI-WARE GEs.....	92
5.3.1	Covering Networking Requirements.....	95
5.3.1.1	IoT Service Stack.....	95
5.3.1.2	FI-WARE mapped design for Device Layer components .....	98
5.3.1.3	FI-WARE - Device Front-end GE.....	99
5.3.2	Event & Data Processing .....	100

---

5.3.3	Interface with Marketplace and Grid .....	102
5.3.4	Building Abstraction .....	103
5.3.4.1	Composite abstraction definitions for data and control .....	103
<b>6</b>	<b>Information models .....</b>	<b>109</b>
6.1	Ontology of building entities .....	109
6.2	Discrete-event models of individual building entities .....	110
<b>7</b>	<b>Communication layer .....</b>	<b>112</b>
7.1	OSGP and ISO/OSI communication in the smart buildings .....	114
7.2	Topological Segmentation of the communication protocols for Smart Buildings.....	115
7.2.1	WAN communication protocols .....	117
7.2.2	NAN communication protocols .....	118
7.2.3	HAN/BAN/IAN communication protocols.....	119
7.3	Communication layers inside the different kind of smart buildings.....	119
7.3.1	Smart Home communication layer.....	120
7.3.2	Residential building communication layers .....	122
7.3.3	Offices/public buildings communication layers.....	123
7.3.4	Data Center communication layers .....	125
7.3.5	Hotels communication layers.....	126
7.4	Characteristics of the communication protocols in Smart Buildings.....	127
7.5	Communications protocol constraints .....	129
<b>8</b>	<b>Components &amp; Communication Infrastructure .....</b>	<b>132</b>
8.1	Components .....	132
8.2	Communication infrastructure .....	135
8.2.1	Generic infrastructure (TI) .....	135
8.2.2	Domain-specific infrastructure (TI) .....	137
8.2.2.1	Home domain communication infrastructure .....	137
8.2.2.2	Residential domain communication infrastructure .....	139
8.2.2.3	Office/Public Building domain communication infrastructure.....	140
8.2.2.4	Data Center communication infrastructure.....	142
8.2.2.5	Hotel communication infrastructure .....	143
8.2.3	Virtualization and simplification of the communication infrastructure .....	144
8.2.4	Communication Infrastructure Requirements .....	145
<b>9</b>	<b>Security .....</b>	<b>147</b>
9.1	General approach.....	147

---

---

9.2	Applied Security Technology .....	147
9.3	Relevance of security requirements to WP4.....	148
9.3.1	Authentication and authorization .....	148
9.3.2	Data confidentiality.....	149
9.3.3	Data integrity.....	149
9.3.4	Non-repudiation .....	149
9.3.5	Data backup and recovery.....	149
9.3.6	System protection components .....	149
9.3.7	Secure SW/FW Updates.....	149
9.3.8	Secure Network Design .....	150
9.3.9	Security Management .....	150
9.3.10	Logging and Audit .....	150
9.3.11	Time Synchronization .....	150
9.3.12	Observation of Policies & Laws .....	150
9.3.13	Transaction Security .....	150
9.4	Specific requirements for data center buildings security .....	150
<b>9</b>	<b>Conclusion .....</b>	<b>153</b>
	<b>References .....</b>	<b>154</b>
	<b>Index of Figures.....</b>	<b>155</b>
	<b>Index of Tables .....</b>	<b>158</b>
	<b>Acronyms and Abbreviations.....</b>	<b>159</b>

## 1 Introduction

It is a pivotal tenet of the Future Internet programme that a set of common-denominator enablers should make up a shared software platform, a foundation for an internet that will be much more than a network, providing transversal services to applications in all relevant environments at their own scale. Buildings are one such environment, where a generic ICT infrastructure is warranted by the development of new “smart buildings” applications, among which buildings energy management for the smart grid can be a prime mover and a loss leader...

The present situation in building automation is very far from this, as each specialized system (such as HVAC or security management) may be entirely closed and vertically integrated with its own networking and its own sensors and actuators. Going beyond these present-day siloed solutions, energy management for Smart Buildings should make it possible to monitor and control all energy-relevant building subsystems, appliances and other physical entities in a non-ad hoc way, operating on top of a shared platform à la Future Internet. It is towards this broad outlook of the architecture that we propose to set course.

This deliverable is D4.3 "Smart Buildings Functional Architecture Description". It describes the detailed ICT architecture of the proposed Smart Buildings energy management system and expands on the previous deliverable (D4.2) "Interim results: Coarse-grain Architecture" which was meant to summarize interim results of the architecture

The architecture is informed by the use case scenarios identified in D4.1 "Smart Buildings" scenario definition" and so we proceed to define the architecture using the use cases of D4.1 as the starting point and elaborating them according to the SGAM methodology. SGAM defines a three-dimensional layered plane and enables a top-down approach whereby the architecture is progressively refined as the use cases are decomposed into constituents residing in the business, function, information, communication and component planes. In this deliverable we focus mainly on the business and functional and, to an extent, information planes, since the communication and component planes are best explored when a concrete technical architecture can be suggested. We refer to this process of use case concretization as drill-down since we are starting at the use case level and are, in a sense, drilling down through successive SGAM planes till we arrive at a concrete, fully specified technical architecture. This exercise is undertaken and reported in Chapter 3. However, we complement this top-down analysis with a bottom-up approach. This is necessary to ensure that our results are rooted in practical considerations. In the same sense that the starting point of the top-down elaboration were the D4.1 use cases, so for the bottom-up elaboration we are using an IoT-inspired vision for a Building Operating System (described in Section 4.2) Chapter 4 goes further into the functional description of the architecture by analyzing the functional building blocks that make up the architecture and providing them as a basis to mapping FI-ware enablers (discussed in Chapter 5) as well as transversal requirements such as security in section 9.

The SGAM information layer is described as the set on implicit and explicit ontologies that underlie the device and entity layers of the FINSENY smart building architecture

The deliverable proceeds with a description of those parts of the Smart buildings architecture corresponding to the Communication and Component layers of SGAM (Chapters 7 and 8), with the provision that the physical layers of the communication infrastructure are, as per SGAM described together with the components layer.

## 2 Methodology

The goal of this deliverable is to outline the ICT architecture of the proposed building energy management system, comprising functional view, infrastructure view and the mapping of functional components onto the generic enablers provided by FIWARE. We also describe the ICT interfaces between the targeted building energy management systems and other systems being addressed in the project: smart distribution grid, smart microgrids and electronic marketplace.

The main top-down analysis tool we are using is the SGAM framework.

### 2.1 The SGAM framework

The Smart Grid Architecture Model (SGAM) had been specified as a joint effort from CEN, CENELEC and ETSI in the framework of the M490 mandate. SGAM provides a useful methodology so as to analyse Smart Grid use cases, from an architectural point of view. It guarantees a consistent mapping on the main architectural layers, while being neutral with the Smart Grid system.

A Smart Building can be considered as a system that is composed of a number of sub-systems that interact with each other, utilising various components. Information and communication technologies act as an enabler for the interactions between the various sub-systems.

The SGAM framework consists of the following layers, as shown in Figure 1, below:

- Function Layer, which represents use cases, functions and services in a way that is independent from their physical implementation.
- Information Layer, which provides the necessary information regarding objects or data models, which are needed by use cases, functions or services.
- Communication Layer, which describes all the communication and connectivity requirements.
- Component Layer, which represents the physical distribution of all participating components in the relevant domain. This includes power system equipment, protection and control devices, network infrastructure, as well as any kind of computers.

Each SGAM layer covers the smart grid plane, which is spanned by smart grid domains and zones as depicted in Figure 1.

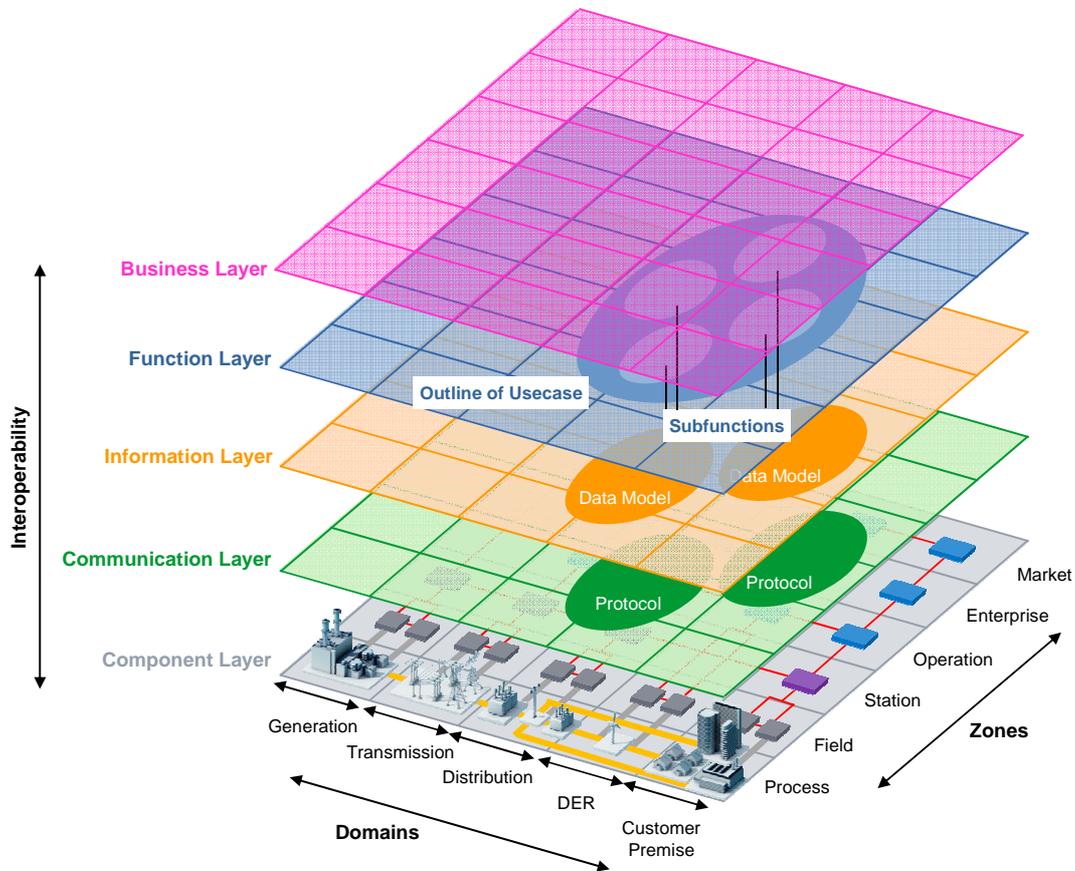


Figure 1 : Layers, domains and zones of the SGAM Framework

SGAM cover the following type of zones, which we will take in the following acceptations:

- *Process* corresponds to the building energy-relevant “hardware”: power equipment (loads, sources, storage, wiring), appliances, but also possibly the walls and roof of the building inasmuch as they have an energy impact
- *Field* corresponds to the sensors and actuators used to monitor and control this hardware
- *Station* corresponds to a level of consolidated data from several sensors and actuators, possibly corresponding to the same entities, including concentrators that perform fusion and aggregation of the sensor data, and conversely “fission” of the concentrator data, as well as modules that maintain this data. This aggregation may itself be at several nested levels
- *Operation* will be taken to correspond for our purposes to a complete building energy management system comprising a “building operating system”, as defined below; encompassing the scope of a single target home or building in the sense addressed here
- *Enterprise*. The SGAM definition is not directly applicable for the DER and Customer Premise zones so we've redefined it to denote energy management systems, solutions and related services / applications involving more than one economic agent or more than one building. For instance, an Aggregator Energy Management system or an Energy Management System for a building comprising multiple subordinate Energy Management Systems for the various appartments inside the building

- *Market* : as defined in SGAM with the additional nuance that it can comprise aspects of the broader environment (to which the energy marketplace is but a part), e.g. publicly accessible web services reporting on weather

## 2.2 Approach for this deliverable

To arrive at the functional architecture for smart buildings we are using the use cases described in D4.1 and the SGAM methodology as the starting point. Then we consolidate the interim design presented in D4.2 and finally provide an analysis of the Smart Building functional blocks and detail functional primitives, dependencies and communication requirements. In Chapter 3 we present the key functional primitives of the most relevant use cases, identified in D4.1 [1][1]. This drill-down allows us to identify the high-level functional building blocks of the proposed smart building architecture in Chapter 4. To organize the building blocks in a proper layered architecture we recognize three separate layers: application layer, shared services layer and building abstraction layer, which are discussed independently in Chapter 4. Chapter 5 then maps these functional building blocks on FI-WARE generic enablers, whereas Chapters 5.3, 7 and 8 surveys the lower SGAM layers: information models, communication layer and components and communication infrastructure.

## 2.3 Analysis Steps

This deliverable is part of WP4, which addresses the interim results of Task 4.3.

Consequently, the activities performed to produce the results available in this deliverable can be summarized as follows:

1. Start from the specification of high- and mid-level Smart Building scenarios and Use Cases of D4.1 [1]
2. Develop a method to drill them down using a layered approach in order to detail the ICT Requirements on the Information and Communication layer.
3. Identify first results on data models, interfaces and key building blocks.
4. Consolidate the design in a final functional architecture, analysing its key functional blocks and primitives.
5. Address and assess FI-WARE integration, communication and security issues.

The steps taken to consolidate the results of this work package were the following

- **Specification of the Smart Building scenario, the high- and mid-level Use Cases and their ICT requirements.**

This step has been described in detail in D4.1, IR4.1 and D7.1. It is summarized in section 3.

- **Categorize and consolidate the Smart Building ICT Requirements.**
- **Identify ICT-based Smart Building Use Cases for a subsequent detailed analysis**

The following two criteria for this selection process are

- relevance according to the work package scope, i.e. the Smart Building scenario in case of WP4,
  - Relevance with respect to ICT beyond the state-of-the-art.
- **Develop a suitable method to drill-down the selected Use Cases to identify the prominent functional building blocks, leading finally to the Smart Building functional architecture.**

- **Detail relevant ICT Requirements dealing with ICT technologies to specify the relevant mechanism for the information and communication layers.**
- **Attempt a mapping to FI-WARE Generic Enablers, where this is possible (Chapter 5)**
- **Revisit interworking, communication and security aspects from the point of view of the consolidated architecture and the results of the other FINSENY work packages (Chapters 6-9).**

### **3 Use cases and ICT requirements of the Smart Buildings domain**

The intention of this section is to summarize the findings of the functional decomposition of the smart building domain use cases carried out in D4.1.

As described in section 2.3, we are following the SGAM methodology. Therefore, to arrive at functional building block decomposition we are performing a use-case drill-down through the Functional layer of SGAM. In a previous step, we studied the Business and Function layers, combining them into a single grid representation. Most of the functions that were identified are not evenly distributed across all SGAM zones and domains because they come from the building use cases, so they are all located in the Customer Premise domain.

For the purposes of this deliverable which focuses on smart buildings, we took the slightly adapted SGAM zone definitions presented in section 2.1

We are going to briefly highlight the results of the use cases analysis, taking into account the five different topologies that we decided to be more representative when we built the use cases in D4.1. For that purpose, the most relevant high level use cases have been selected.

Domain	High Level Use Cases	Analysis Summary
Home domain	Monitor and manually Control Energy Use	<ul style="list-style-type: none"> <li>• Hierarchy of functions to collect, store, aggregate and display energy consumption global and detailed information to successively higher echelons spanning all the way from the Process zone to the Operation zone.</li> <li>• One different function to forecast the energy consumption located at the Operation zone (BEMS), and another one to generate alerts to the final user at Field and Station zones (smart meter and gateway).</li> </ul>
	Globally optimize home energy use	<ul style="list-style-type: none"> <li>• At the Distribution domain, a set of functions are required to provide the BEMSs at the Operation zone with needed information about the electric tariffs, load reduction signals, peak loads, emergency control signals elaborated at the Market zone, in order to be able to perform their operation for optimizing energy use. Also, the information about the level of energy generated at the house is used in this domain for the global home energy optimization.</li> <li>• At the Customer domain a set of functions are required to provide the BEMSs at the Operation zone with quantitative energy information and at the other side (the BEMSs at the customer premises) with notifications of load controls and other control signals.</li> </ul>
	Locally optimize home energy use	<ul style="list-style-type: none"> <li>• The use case function decomposition is practically the same, except that the function “Aggregate home energy generated info” is located at the Customer domain because the optimization is done at local level, that is, the generated energy in the house is used for internal consumption.</li> </ul>
	Generate and Store Energy locally	<ul style="list-style-type: none"> <li>• At the Customer domain. Both generation and storage of energy in the house are used for internal consumption, not transferred to the grid.</li> </ul>
Residential domain	Monitor Energy Use	<ul style="list-style-type: none"> <li>• Hierarchy of functions to collect, store, aggregate and report energy consumption readings to successively higher echelons spanning all the way from the Process zone to the Operation zone.</li> <li>• Functions to publish readings at either home or building level at the Operation and Enterprise zones respectively.</li> </ul>
	Support Online Community	<ul style="list-style-type: none"> <li>• Automatically publish aggregate readings to web social-media applications or gamification platforms as well as rich GUIs to support the interaction, the view of historical information or trends, and the current consumptions.</li> </ul>
	Generate Locally Energy	<ul style="list-style-type: none"> <li>• Generate electricity, monitor and control electricity generation sources,</li> <li>• Make sell decisions (e.g. to decide whether and when to sell the locally generated electricity to the grid or market or whether and when to consume it or store it locally), as well as functions to obtain market information on prices, etc. which inform that kind of decisions.</li> </ul>

	Provide Emergency Electricity	Diagnose that an emergency exists and functions that allow the Building Energy Management System to activate the use of emergency electricity reserves.
	Store Energy Locally	Comprises the generation and storage of electricity, APIs to monitor and control the storage units, functions to display the status of energy reserves, an optimization function to decide whether and when to buy energy from the market for local storage and a function to buy the energy from the market.
	Optimize Energy Use	Comprises the generation and storage of electricity, monitoring and controlling the electricity generation and smart appliances consumption.
Office/Public Building	Check Energy Use	The BEMS provides regularly updated historic, real-time and/or forecast energy usage data of the office building via displays/information screens/web browsers to the end-users with the goal to motivate the staff to use energy conservatively.
	Check acute Alerts	Information about the energy use of the office building is available remotely. If an anomalous situation is detected, an alert is sent out. The information exchanged between the actors is the energy used by the devices inside the building and the alerts in abnormal situations.
	Allow Real-time DR events in the service center	Real-time Demand Response (DR) events are received by the energy manager in the building. The energy manager can then decide on the best strategies to respond to or exploit these events.
	Check DR period in the office room	The BEMS should provide to the end users / office workers information about the DR actions that have been taken. Means should be granted to end-users to override these actions, especially in shared spaces such as meeting rooms. Information about DER actions is always accompanied with advices to the end-users for using energy conservatively.
	Energy Coaching	Bidirectional exchange of information between the end users and the BEMS. The BEMS provides the end users information on the amount of energy consumption and on the costs associated to each energy hungry device that they normally use. The BEMS learns from the ordinary behavior of the end-users in order to adapt the operation of the devices.
	Check benchmarking in districts	The energy consumption data from buildings in the district is obtained and provided to the end-user to enable benchmarking.

Data Centre	Optimize the data center air conditioning	As the temperature is not uniformly distributed in the local data center, a set of temperature sensors is used to ensure reliable detection of temperature. These sensors send wirelessly information to a collection point. The information gathered can activate in real time, wireless-actuators in order to turn on or off the air conditioners.
	Optimize the free-cooling in the data centers	The sensors detect, in real time, the temperature values at different points in the data center. A monitoring system compares the values collected for each humidity sensor with those defined as acceptable operating margins. If the humidity values recorded are above or below the acceptable operating margins, the monitoring system sends radio commands to the actuators that change the air flow or temporarily activates the air conditioners for the dehumidification.
	Optimize server power usage	A set of smart info power meters controls in real time the server power consumption and sends, via radio, the measured data to a gateway point. Then the real time power consumption of each server is compared with the nominal one and, if this last is greater, then it will be possible to increase the number of servers without extra energy from the systems of power distribution.
	Manage business continuity	When a power outage or a cooling system failure occurs, HW/SW commands are sent to reduce the power consumption of the servers without penalizing too much the quality of their services (QoS).
	Optimize power workload while maintaining a high quality of services	Power optimization requires a table with various workload profiles and a performance loss target not to be exceeded. Developers perform a series of experiments to characterize how much capping can be applied before the performance target is hit. Afterwards, during normal operations, the applications engineer sets power capping targets based on the prior measurements.
	Choose between multiple service classes in function of the workload priority	High priority workloads run on unconstrained servers whereas the medium priority workloads are assigned to power capped servers. The financial manager presents to the customer a tariff that depends on the expected level of the quality of services (QoS).  The real-time power consumption of the server and the pre-characterized applicable energy capping are sent to the facility manager that through HW/SW commands optimizes the power consumption of the servers without penalizing the quality of their services (QoS).
Hotels	Shed Load	<i>The strategy includes energy consumption and generation. In collaboration with the utility a hotel can participate at the energy exchange and offer positive and negative reserve energy. A strategy includes a tolerance range in temperature (swimming pool, room) which has an effect on the possible energy reserve that can be offered. In addition to that, energy storage can be installed in order to optimize against a set of KPIs defined by Hotel Management.</i>

	Execute Strategy	<ul style="list-style-type: none"> <li>The strategy includes energy consumption and generation. In collaboration with the utility a hotel can participate at the energy exchange and offer positive and negative reserve energy.</li> </ul>
	Store energy locally	<ul style="list-style-type: none"> <li>The execution of the strategy may be based on static information such as occupancy of a room (number of persons, wake-up time if set) as well as dynamic information such as motion detectors or occupancy sensors.</li> <li>By dynamically setting the price for the different energy forms, the utility influences the execution of the strategy.</li> </ul>
	Generate energy locally, Store energy and Request energy from the grid	<ul style="list-style-type: none"> <li>The Energy Manager decides to generate some energy locally. Especially for renewable energy sources the decision to generate energy locally may also be triggered by external factors such as wind speed, intensity of sunlight, etc.</li> </ul>
	Define and Monitor KPIs	<ul style="list-style-type: none"> <li>The KPIs are a guideline and optimization criterion for the energy manager. Usually the fulfillment of the KPIs is closely coupled to the compensation of the person or organization responsible for the Energy Management, which is why the KPIs are monitored to create a feedback.</li> </ul>

## 4 Smart Building Functional Architecture High-level Building Blocks

### 4.1 ICT Perimeter of the Smart Building and external interfaces

The overall FINSENY high-level architecture diagram was derived in preparation of this task in the FINSENY Architecture Group to show the interrelation between the various scenarios in FINSENY and to ground the study of the various external interfaces.

FINSENY High Level Architecture

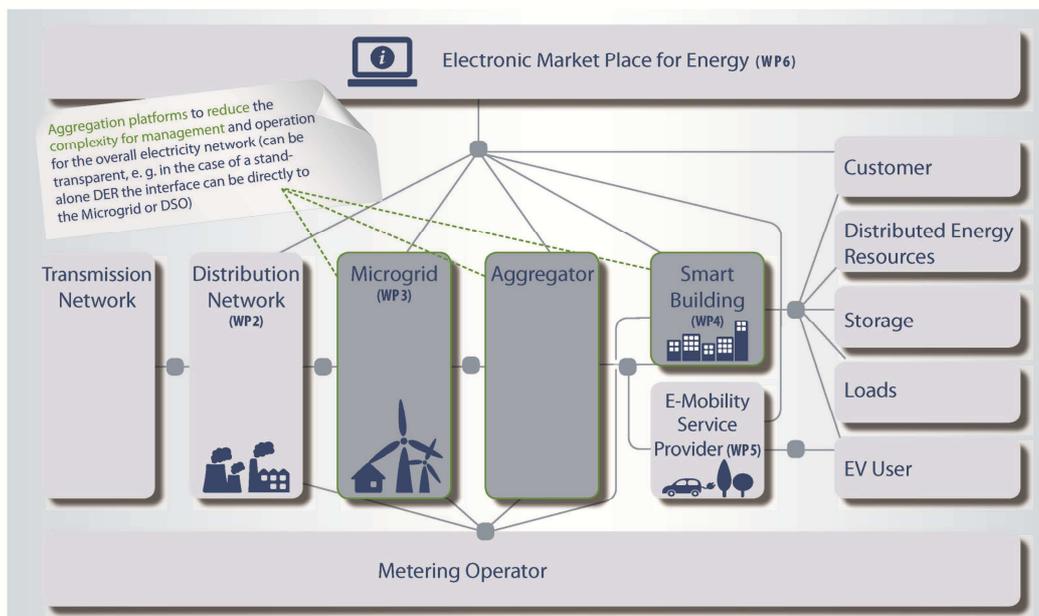


Figure 2: Smart Buildings external interfaces with other FINSENY domains and the Energy ecosystem.

This section delineates the perimeter of the Smart Building, and more precisely the ICT perimeter since this document deals with ICT architecture. The boundary is defined by describing (at a high level) the semantic content of the interactions of the smart building ICT system with its environment as well as the information models of the data flows exchanged (also at a high level). When defining an ICT architecture this approach usually yields a clear separation between the functions that are implemented inside a system and the functions or services that lie outside the system and on which the system relies. In defining the boundary we take a black box approach of the internal Smart Building's architecture. Once the boundary is defined, we begin to elaborate an architecture that can support the various "contracts" implicit in the external interactions we have defined and also, one that can host the various functional building blocks we identified in Chapter 3 as part of the use cases functional decomposition exercise. This latter point is treated in Section 4.2.

#### 4.1.1 Interfaces to Energy marketplace

The major European wholesale power markets see energy traders, retail companies and large consumers as the major actors. In this marketplace, electricity is exchanged through bilateral contracts or directly on the spot power exchange market (e.g. IPEX in Italy) where traders aggregate demand and resell the power to retail customers. With the introduction of smart buildings and smart building energy management systems, the energy marketplace we know today could radically change. Indeed, with the advent of smart buildings one is able to have an explicit view of customer consumptions but more importantly of their historic trends. Knowing

the consumption trends of a customer or group of customers, one is able to profile the consumption and allow energy managers/management systems to fit the customer needs with a variety of energy contracts. For example, standard bi-lateral contracts could be used to respond to the customer’s base load throughout standard consumption hours. Moreover the missing or excess energy demand could be met by buying or selling electricity on the spot market.

In the event the consumer also has the means to produce his own energy via distributed generation means (thus being a prosumer), then the energy manager should also be able to analyze the energy market place in order to assess which is the best way to use the energy being produced. If energy storage devices are available or if one is also able to modulate the production of the distributed generation plant, the energy manager will have an even broader spectrum of possible actions at his disposal.

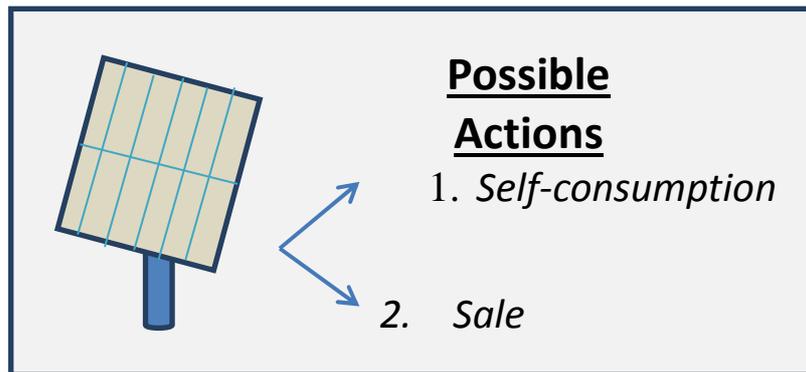


Figure 3: Prosumer - Weather dependent DG (e.g. PV)

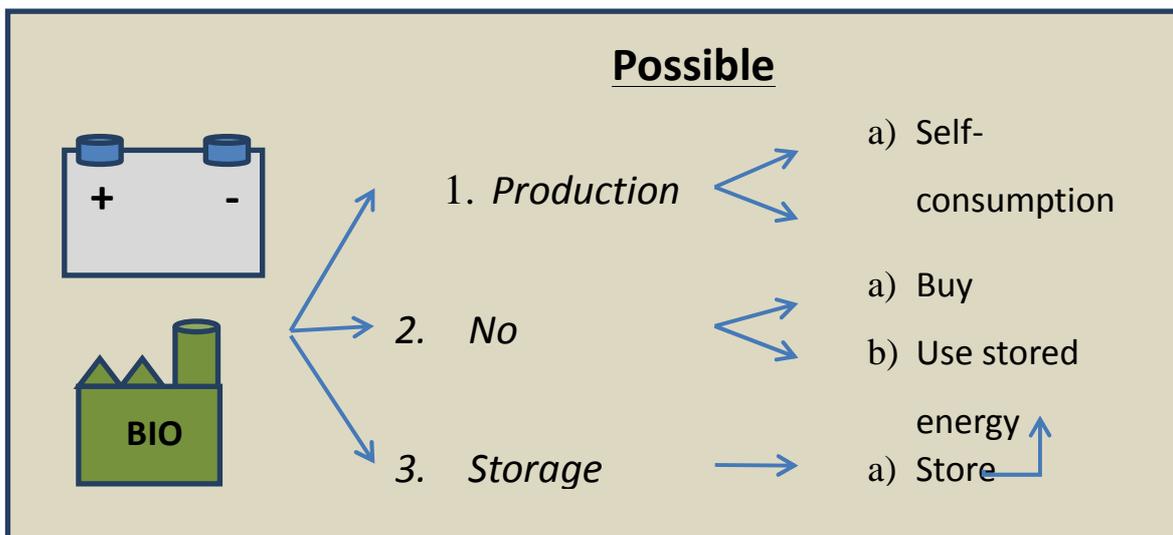


Figure 4: Prosumer – Weather Independent DG & Storage (e.g. Batteries & Small Scale Bio mass Units)

Development and improvement of cost-effective energy storage systems will also be important in order to facilitate a larger penetration of distributed generation units by decoupling generation and energy consumption.

The information which has to be exchanged between smart buildings or aggregators and the energy marketplace is the real time energy consumption and production of the consumers / prosumers. The interfaces which allow one to obtain this information are smart meters. However meters must also be able to talk with the smart building energy management system for this information to be used appropriately. Indeed, data exchanges with dedicated ICT platforms controlling the information flows between the different players may strengthen the capabilities of real-time trading, generation control and participation on the demand side.

A large number of prosumers will cause a considerable amount of data to be shared as the flow of information will no longer be bilateral (Energy Provider → Customer) but multilateral (Energy Provider ↔ Energy service providers ↔ Prosumer). The management of this information will shape new roles for the traditional players who will have to share not only information related to the sale of electricity, but also information concerning transmission costs and generation revenues which will be shared amongst different types of actors in the energy marketplace. In this way, installation of smart meters along with smart building energy management systems may rationalize energy usage and stabilize electricity prices throughout different hours in the day.

The advent of smart buildings and the relative instruments which adorn them might drive the energy market as we know it to become a flexible and dynamic electronic marketplace for energy. Moreover, with the aid of dedicated ICT platforms and energy management systems, consumers may follow the trend of becoming vertically integrated prosumers.

**4.1.2 Interfaces to External applications**

We are anticipating the following developments in the medium-term:

- New business models may encourage a move from hierarchical command and control operations to symmetrical peer to peer negotiations on the power grid
- Renewable energy sources will increase reliability
- Distributed generation will create more power sources not under the control of traditional utilities
- Energy buildings will make each end-node both a buyer and seller of power.

Starting from the top level conceptual model shown in Figure 2, it is clear that the prosumer is not only connected to the distribution domain via the meter, but also to the markets, grid operations, and customer service provider domains via communication networks.

To successfully unleash the potential of buildings in the Smart Grid, the facility interface must constitute a clear demarcation point between grid operations and facility operations. To successfully enable markets, motivate customers, optimize assets and enable efficient grid operations it is likely needed to exhibit the properties identified in Table 1 below:

Principle	Action
Loose coupling	Describes a resilient relationship where each end of a transaction makes its requirements explicit with minimum knowledge of the other side of the interface.
Composition	The building of complex interfaces from simpler interfaces enables diversity. Composition also means that the base, simpler services are available, and, hence, can be repurposed and recomposed.
Layering	Denotes separation of function and loose coupling between them. A layer has a general function and provides services to the layer above while receiving services from the layer below. A communication stack is composed of layers, just as a protocol standard is composed of simpler component standards.
Scalability	The Smart Grid applications, components, and participants are expected to grow rapidly as standards mature and infrastructure is modified or added. System performance should not be detrimentally affected as components and capabilities are added.
Security	This mandatory point is of vital importance and should be implemented

	at all hierarchical levels of the SGAM described here. In particular, every HW / SW interface element interacting with the outside world must take on board the appropriate security techniques. However, at the same time the security approach must allow the transactional transparency for the marketplace operation in terms of traceability of the events
--	---

Table 1: Architectural external interface principles

The facility interface must conform to these architectural principles to meet the goals of the Smart Grid to enable innovations, ensure interoperability and grid reliability.

As shown in Figure 2, the Smart Building has two primary gateways: the “Metering Operator” and the “Electronic Marketplace for Energy”, with its distribution in the domain communications. Then there are the interfaces, whose properties listed in Table 1 must be met, aimed at microgrids and aggregators, and internal building points such as load, home customers, storage etc. For this reason, an Aggregation Platform is expected to simplify the complexity of the entire electricity grid interconnection and, at the same time, to manage the security aspects in more efficient way.

The Aggregation Platform shown in Figure 2 attempts to summarize the energy interoperation communications that involve both the grid and the building. The Aggregation Platform I/O interaction interfaces, also enable the prediction of future energy use (expected demand in kW) for future time intervals, and may be generated by analysis of past use, facility schedules, weather, sub-system status and known user plans. Rather than deal directly with a market, the facility may send these forward demand estimates (or supply estimates in the case of potential demand reductions or generation/storage resources) to an aggregator who then bids this demand or supply resource into a wholesale market. Accurate estimation of sub-system demand may rely in part on sub-system energy profiles. Energy profiles may serve not only for configuration purposes (e.g., identifying sub-system load shed capabilities) but also as a resource for dynamic status: operational mode, faults, power level, storage status, etc. The subject of energy profiles is a topic of ongoing research.

In conclusion, interactions with the grid should occur at a secure interface with external applications that also serves as a demarcation point of ownership at the domain boundary. Communications across the interface should be collaborative in nature, with simple data exchanges that require minimal knowledge of how that information is used or what protocols exist on the other side of the interface.

**4.1.3 Interfaces to Microgrid management system & distribution network**

A building or a home is likely to have two types of physical network interfaces with either the microgrid or the distribution network management system:

- *On the one hand*, an interface provided by the metering operator, namely the smart meter, connected to a full AMI (Advanced Metering Infrastructure). The smart meter has long been used for the billing of the energy supply. As such, it also supports dynamic pricing of energy, as it is mandatory to measure the energy consumed in different tariff periods, and, depending on the regulations of different countries and on the commercial offerings of the utilities, it may also be used to deliver to the end customer various services : consumption information services and basic energy management services, to help the customer to better manage his consumption and, in particular, to allow him to better cope with dynamic prices that are one of the means for DSM (Demand Side Management). Additionally, the locally generated energy produced by DER that is increasingly installed in homes and buildings also have to be measured by smart meters, so that the DER owner can be paid for the electricity exported to the grid.
- *on the other hand*, connection with the Internet is delivering high speed connectivity with all kinds of applications and services and, as such, will allow the implementation

of the most advanced use cases, in particular those that can't be supported through the AMI.

As this sub-chapter is related to the interface of the home or building with the Microgrid and/or distribution system, the use cases described in [1], [5] and [6] were examined in order to identify the type of data exchanges using either one or the other of this two network interfaces. It must be noted that the data exchanges related to DSM (Demand Side Management), typically leading to exchanges with an aggregator, have mainly been described in the subchapters above and will not be described extensively here again; however, as this aggregator role could in fact be endorsed by a DSO or by a Microgrid Operator, these data exchanges could be potentially added to those described below.

The description of these data flows is given in the list below; as can be seen, they can be related to the electrical network operation – either Distribution Network or Microgrid – or to deliver services to the customer, and on another hand, the direction of the data exchange may vary: going upstream from the home or building to the grid operator or service provider, or going downstream, for example, for control purposes.

Traditional energy supply requires periodic meter reading of the measurements made in the different tariff periods, now completed with the load curve (with intervals generally of 10mn, 15mn, 30mn or 1hour), so that the billing of the consumed energy can be made. The period of these readings may vary between once every 2 or 3 months and once a day. Together with the basic consumption data, other important characteristics of the consumption may also be read such as: reactive energy, data related to the quality of the energy delivery at the customer end point (for example minimum and maximum voltage, frequency), frequency and duration of power outage at the end point, etc.

Additionally, the same kind of data, but even more frequent (for example several times a day) can be read, either remotely or locally, completed with alerts, in order to provide the customer with consumption information services, in the frame of the high level use case “Monitor and manually Control Energy Use” (see D4.1).

The above data flows may also apply for electricity generation by Distributed Energy Resources owned by the customer (in the capacity of a prosumer in this case). The electricity exported to the grid has to be measured so that the bill can be established and the prosumer paid. It is conceivable that, here also, dynamic pricing may apply, so that several readings, related to different tariff periods, are to be made. And here also, more frequent readings may allow the system to provide the prosumer with monitoring services so that he may ensure an optimal functioning of his DER installation. But, as we'll see below, DER also leads to other data flows.

In order for the metering to be comprehensive, either for consumed or for generated energy, the right configuration has to be downloaded in the metering devices. Such a configuration generally includes tariff periods, but also subscribed maximum power and possibly other critical contract parameters.

Providing the customer /prosumer with the adequate tariff information (periods, prices of energy in each of these periods) is also important, so he can have all the monitoring information, but also to meet the expectations described in the “Optimize home energy use locally” high level use case (see D4.10). In particular, they should allow a Home or Building Energy Management System to perform its optimization task. On the other hand, the design of these dynamic tariffs and the incentives that they provide to the customer/prosumer to adapt his consumption/exported-consumed generation accordingly, can be exploited by the utility or by the Microgrid operator towards a global optimization of the grid. This information may either be provided through the AMI, as the metering system already holds some of that information for the purposes of billing, or through the Internet. Additional information, not always energy-related, may also be required to perform this optimization: the weather and temperature data forecast are a typical example.

More direct control information may be sent to the home or building energy management systems. In particular, as stated in the FLIR (Fault Location, Isolation and Service Restoration)

use case for Distribution Networks in [5], DER modulation and control may be necessary in order to automatically shed/reduce/increase generation and loads to preserve load balance and system stability (load curtailment and source support to diminish a given congestion). The same kind of requirement also applies to Microgrids, with even more accuracy.

Planning data collection is also important, specifically for “Balancing supply and demand” use cases of Microgrids (see [6]), but they may also help a lot the Distribution Network, typically in high congestion risk areas. This involves collecting information from the customers, DER and storage owners. This may for example involve the collection by the BEMS of data from the various smart appliances in the home and their aggregation at the local level.

Microgrids may also lead to increased needs of monitoring and control of loads and generation means.

The Optimal Power Flow (Intelligent loss minimization) is a specific use case for Microgrids (see [6]) that also involves direct load and generation control in order to ensure that generation and consumption points are geographically close from each other in order to minimize losses.

The management of the islanding mode is also a situation specific to Microgrids where load and generation control plays an important role.

It should be noticed that, in such cases, the monitoring and control is a continuous process. As stated for example in the DSM use case for Microgrids, it leads to the continuous determination of available controlling power range on different timescales. And what is true for Demand Side Management also applies in that case for “Supply Side Management”.

## **4.2 Overall view of the functional architecture**

We present the proposed smart buildings functional ICT architecture within a generic framework that is in line with the overall FI-PPP approach: raising the level of the shared infrastructure, beyond the basic hardware and connectivity. This framework is “horizontalized” in a way that is also in accordance with FI-PPP principles: infrastructure should not be dedicated and vertically integrated with applications; it should be as generic as possible and available to other applications operating upon the same physical plant. The Smart Buildings architecture is thus meant to support not only Building Energy Management but also other applications needing access to the physical plant of the building for monitoring and control.

### **4.2.1 Building Operating System**

This genericity of the proposed infrastructure is highlighted here with its consolidation as a “Building Operating System”, itself decomposed in to three levels, as represented in Figure 5

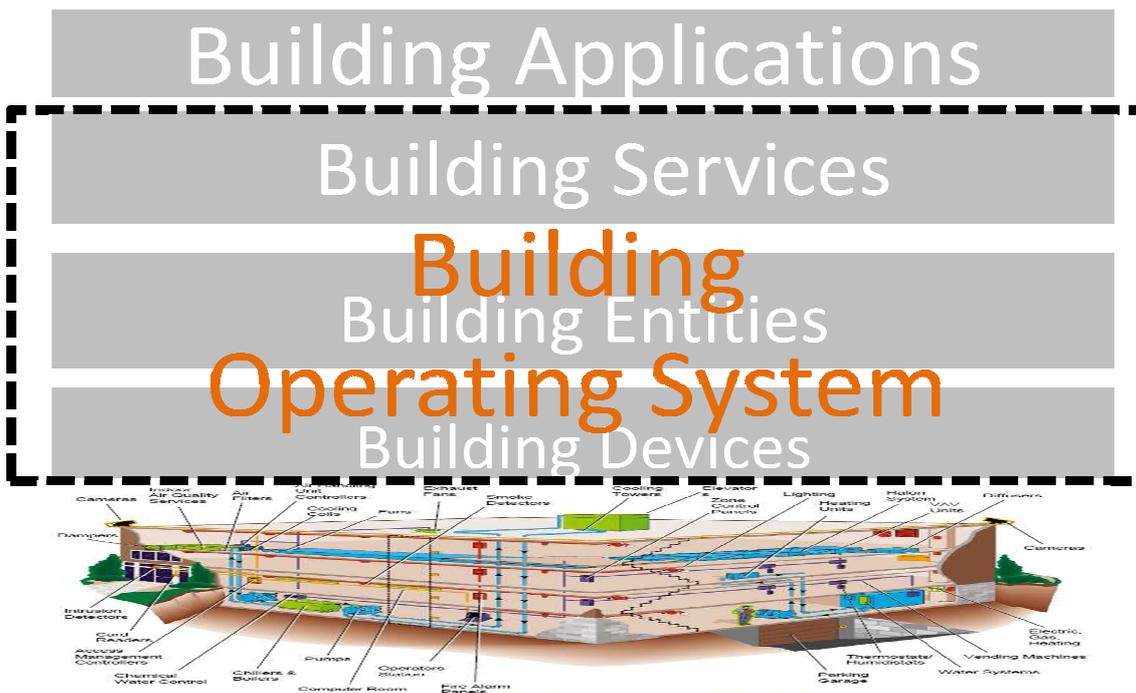


Figure 5: The FINSENY smart building architecture framework

The matching of this framework with SGAM is represented in the following figure.

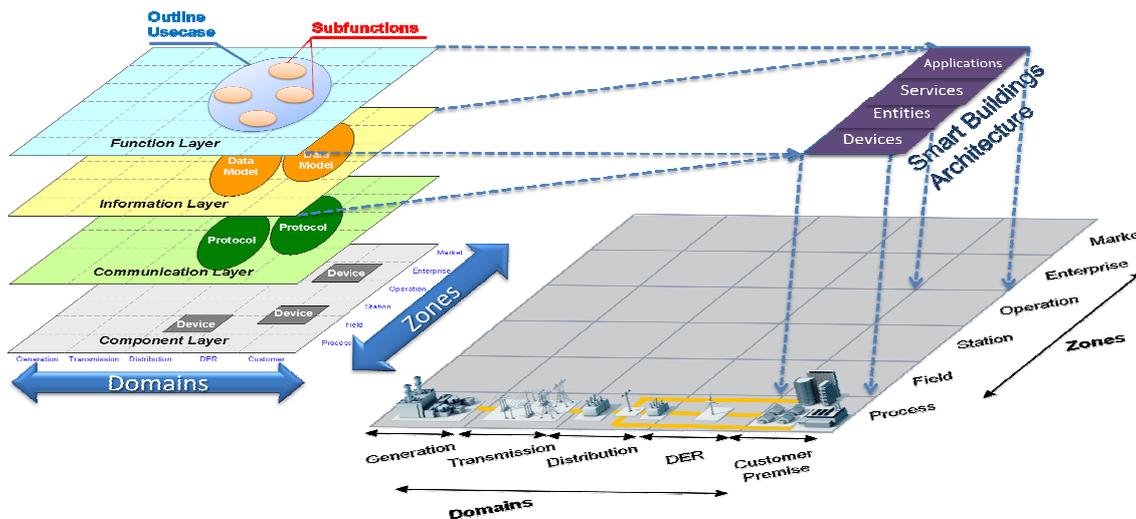


Figure 6 : Matching of Smart Building Architecture to SGAM framework

The architecture describes information and function layers, while the matching with zones could be taken roughly as follows

- **Field Zone** → Devices
- **Station zone** → Entities & virtual entities
- **Operation zone** → High-level (building-wide) services & applications

The motivation for the various levels is discussed below, while the following sections describe each layer in more detail.

Sensors and actuators are supposed to be shared between all building applications and made available as a pool when they are individually identifiable and addressable (right-hand side of

the diagram of Figure 5). Building legacy systems (such as dedicated HVAC management systems) that do not give direct access to their individual sensors and actuators will be dealt with through their own interfaces (left-hand side of the diagram). They will be considered as black boxes, but integrated within the perimeter of the system nonetheless.

Building automation applications are not interested in sensors and actuators themselves, but in what is being sensed by the sensors, or acted upon by actuators. The relevant level of abstraction for information pooling should thus be at the level of the physical entities that are being sensed by sensors and acted upon by actuators, which can be pieces of equipment, appliances, people, rooms of a building, or more generally any relevant self-contained subsystems of the building. These entities are generic, intrinsic to the building environment and not tied to any specific building automation application. A set of models and corresponding software components for these entities make up a “Building abstraction layer”, in a way similar to a hardware abstraction layer for a computer platform.

Taking a room of a building as an example such entity, the state of a room could be whether it is occupied, the type of activity going on, and the corresponding attributes could be its temperature, the number of persons present, etc. For control purposes, an application can change the state of an entity to another state, if admissible, or change associated attributes. In the examples given for supervisory control in section 4.4, the state of a room could be changed to dark by sending coordinated commands to individual actuators, such as those controlling shades and light fixtures.

An additional service layer, corresponding to software enablers that span several entities or entity categories, may be provided to applications on top of the building abstraction layer to make up the building operating system. Absent such services, the interfaces that are exposed to applications from this building operating system may correspond directly to the states and associated attributes of relevant physical entities of the building. Further, absent the models representing such entities, interfaces exposed to applications from this building operating system may correspond directly to sensor readings and actuator control parameters

The use of these stacked layers of information abstraction is in line with the Internet of Things and Context Management enablers provided for the Future Internet platform by the FI-WARE project [4].

#### **4.2.2 Generic functional building blocks**

A more detailed decomposition of this architecture is presented in the following figure: the main building blocks of the architecture are represented through examples of their instances and example instances of their mutual relationship.

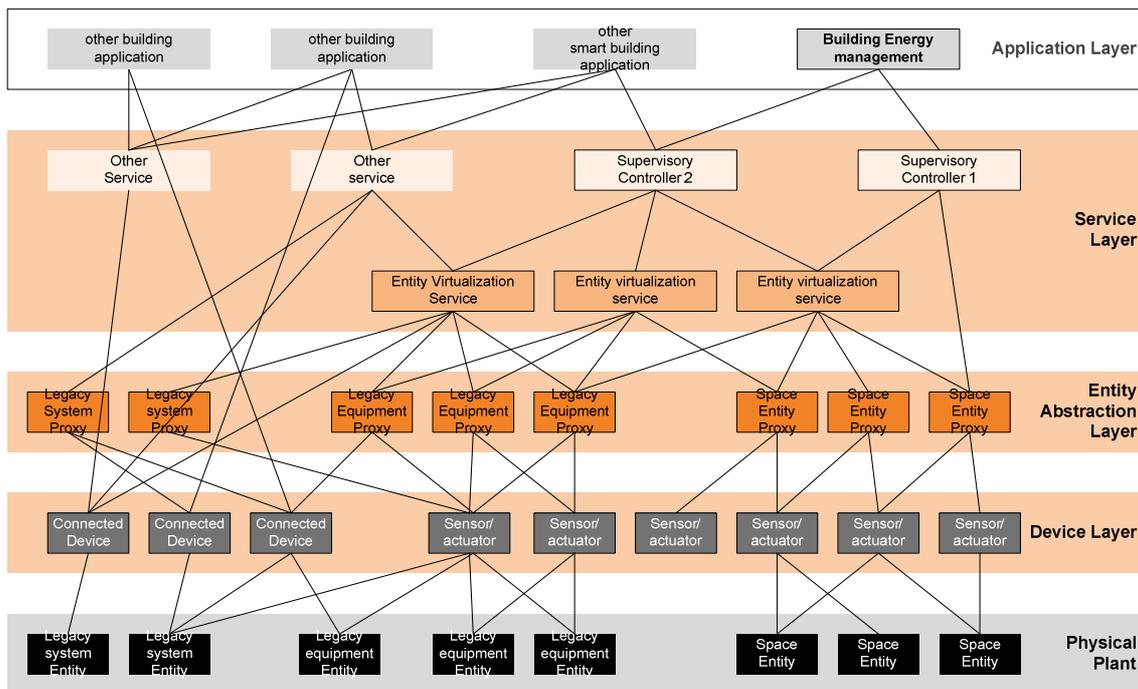


Figure 7 : Generic functional building blocks of FINSENY Smart Building Architecture

The following table recapitulates the main high-level building blocks of the FINSENY architecture pictured above, together with their external interfaces:

	<b>Building block</b>	<b>External interfaces</b>
1	(Application-level) Building energy management (see sec 4.3)	Microgrid Market Distribution network
2	(Service-level) Supervisory control services (section4.4.5)	Microgrid or Distribution network, by exception to separation of concerns principle
3	Entity virtualization services (see section 4.4.1)	Microgrid or Distribution network, by exception to separation of concerns principle
4	Building space Entity Proxies (section 4.5)	Building occupants
5	Building equipment Entity Proxies (section4.5)	Microgrid or Distribution network, by exception to separation of concerns principle
6	Building Legacy Systems proxies (section 4.5)	Operators, facilities managers Microgrid or Distribution network, by exception to separation of concerns

		principle
7	Building connected devices	
8	Building Sensors (section4.6)	Building occupants External environment
9	Building Actuators (section 4.6)	Operators, facilities managers

Table 2 : Main Building block of the FINSENY Smart Building Architecture

Entity virtualization services represent groups of entities through shared functionalities or properties and provide a common interface to these entities for applications. This may be for example lighting or heating services that regroup all entities that provide lighting or heating, either as their primary function, as their secondary function, or even as a side effect of their unrelated primary function.

Subsets of space represented by a proxy are those that are relevant for more than one application and that have general significance with regard to the building itself, such as individual cubicles in an office space, rooms or floors in any building.

Pieces of equipment represented by a proxy in the building abstraction layer are any piece of equipment that is individually monitored and controlled through their individual generic model.

Building users/occupants may be modeled separately as entities, as this is relevant for other applications that the ones we are primarily interested in

Legacy systems are integrated systems such as HVAC systems that are dealt with as black boxes rather than representing their individual components as legacy equipment in the building abstraction layer: this is clearly relevant for large buildings rather than for individual homes.

Connected devices are, beyond sensors and actuators, regular networked devices that may or may not be represented separately as “entities” in the building abstraction layer. If they are not represented as entities, this may be because they are not relevant for their energy impact, or because they already have specific models of their own through a regular service infrastructure such as UPnP. Yet even for devices that are represented through specific models in such infrastructures, it may be useful to mirror them through more generic models in the building abstraction layer, which means the specific model has to be associated with the generic one.

Basic sensors and actuators, whether they are standalone or embedded in such connected devices, will usually not be dealt with separately as entities. They are merely intermediaries.

### 4.3 Application layer

According to the horizontalization principle highlighted before, the application layer is limited to software that is dedicated and interfaced directly with external actors. Software that can be shared between different applications, such as a generic supervisory controller, is supposed to belong in the service layer rather than the application layer. The smart building application layer comprises classical applications such as security management, safety management, building automation. We focus here on energy management but it should be clear by now that all other applications are to be supported from the same interfaces to the underlying infrastructure.

We describe here different application-level solutions for home or building energy management and their relationships with the lower layers of the architecture described above. These solutions are, between one another, partially alternative solutions for dealing with overall building energy management and partially applications complementary solutions that do slightly different things and should work concurrently.

### 4.3.1 Energy@home Application Layer

Energy@Home is a step towards the development of the home as the end point of the smart grid that, in the future, will allow continuous real-time two-way information exchange between utilities and appliances in the houses to enable each customer to “self-manage” his/her energy behaviors depending on power supply and prices.

From an application point of view, Energy@Home envisions a system that can provide users with information on their household consumption directly on the display of the appliance itself, on the smart phone or on their computer. It is expected that, through easy access to information on consumption and through the possibility of downloading custom applications, consumers will be able to use their appliances in a “smart” way by enhancing the energy efficiency of the entire house system. For instance, Smart Appliances can start functioning at non-peak (and therefore less expensive) times of day as well as they can cooperate to avoid overloads by automatically balancing consumption without jeopardizing the proper execution of cycles.

End users can remotely monitor and control the house and all connected home devices and appliances from anywhere at anytime through a commercial Smart Phone. The system can be configured to send text messages in case of alarms or warnings (e.g. power overload, local blackout).

Moreover, Energy@home pro-actively schedules the user loads and the smart appliances when green energy from renewables is available, according also to the weather forecast. By synchronizing consumption with local micro-generation the consumer increases the economic incentives and the power grid works better.

All these applications are governed by the residential broadband gateway that coordinates the Home Area Network and enables the seamless integration with the Internet. It provides function for remote monitoring and control and for integration with energy tariff information. Its Java execution environment allows installing multiple applications and provides a variety of new Value Added Services.

This section describes in detail the application logic in Energy@home through a set of sequence diagrams that show the possible interaction between the gateway and all the involved devices.

For more details about the features and the experimental results obtained with the system Energy @ home, please refer to the FINSENY deliverable D8.2 “*Experiments and evaluation*” [8] and D8.3 “*Selected domain specific enablers specification*” [9].

#### 4.3.1.1 Control modes

The interactions between the Energy@Home devices can be operated in two different control modes, depending on how each device is willing to participate to the overall system control operation:

- Operating mode without E@H control (**Energy@home control disabled**): In this case the awareness scenario is covered but the devices in the E@H network shall not be scheduled and controlled by the Home Gateway or energy Management System;
- Operating mode with E@H control (**Energy@home control enabled**): this represents the full set of Energy@home features: the appliances can be automatically scheduled according to the needs of the user and pre-emptive and reactive control on the devices is allowed.

Selection of the control mode has to be harmonized by the functional controller (e.g. the Home Gateway or Energy Management System), how that is done and how it is selected by the user is implementation specific and is outside the scope of these specifications: for instance a special button on the appliance might be used or, alternatively, a special function on the Central User Interface, or some other mechanism, may be adopted depending on the implementation.

### 4.3.1.2 Start-up and discovery

The device association and discovery procedures are dependent on the underlying protocol used. However, the general Start-up procedure shall follow the steps listed below:

**Home Gateway present:**

- The Home Gateway opens the network (i.e. enable other device joining the HAN);
- The Home Gateway manages the authorization and authentication of the new HAN devices willing to join the E@H network;
- The services offered by the HAN devices shall be automatically discovered using the underlying protocol service discovery procedures: the E@H devices shall then detect the addresses of the devices they are required to communicate to;
- an auxiliary mechanism for enabling the configuration of the HAN by using an interface exposed by the Home gateway should be supported as well;

**Home Gateway NOT present:**

Since the Home Gateway is not available, the admission procedure should be managed by another device, responsible for the authorization and authentication of the new HAN devices willing to join, which shall provide user with a user-friendly interface; alternatively, if no user interface would be supported by this device, a pairing mechanism with the other HAN devices shall be enabled (such as button pressed or other peering techniques).

An example of sequence diagram is reported in **Figure 8**, where a Smart Info and a Smart Appliance send a command to the Home Gateway for the network joining and association.

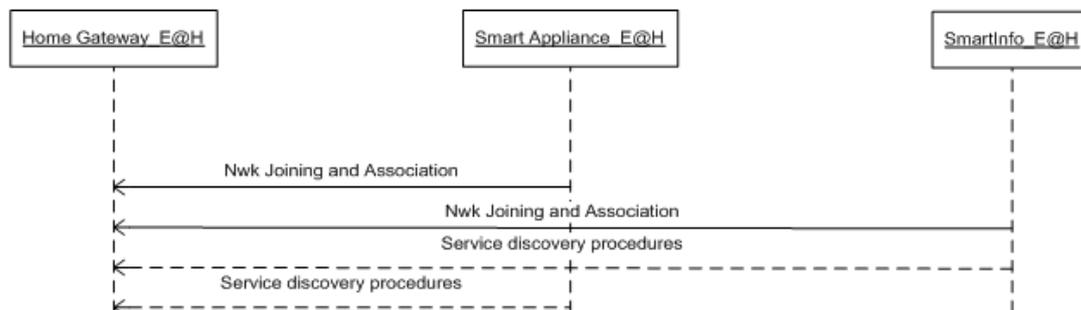


Figure 8 : Start-up and discovery procedure.

Then the service discovery procedure shall leverage on the procedures defined by the communication protocol.

### 4.3.1.3 Customer Awareness

#### 4.3.1.3.1 Visualization of current energy, power and price data

The energy, power and cost information should be distributed on the E@H network using the procedures depicted in **Figure 9**. In case the Home Gateway is operating in the E@H network it shall acts as a mirror for the information to the other devices on the HAN: that means that the Home Gateway shall maintain up to date data related to energy, power, and energy cost (if required), associated to each device as well as metering data from the Smart Info related to home global consumption.

The devices willing to access this information should access the mirrored information in the Home Gateway. That mechanism provides the following main advantages:

- it enables sleeping devices in the network: since devices may sleep in the network, the Home Gateway (always-on device) should buffer the data to be retrieved by the other devices in the HAN;
- it reduces the need of broadcast messages enhancing the performance in case of wireless E@H network: the mirroring feature on the Home Gateway enables the other devices to communicate in unicast to the gateway itself, reducing the need of the broadcast messages in the HAN (typically considered unreliable mechanism for the wireless HAN).

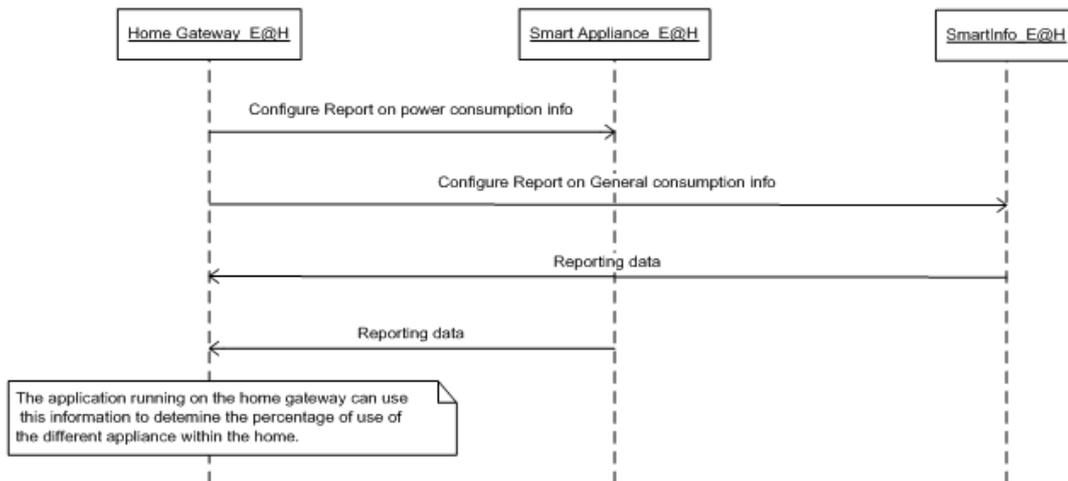


Figure 9 : Configuration of energy, power, and price reporting procedure.

**Figure 10** shows the configuration of instantaneous power reporting on appliances: the Home Gateway send to the Whitegood the power reporting and this accept it. Then the whitegood reports the instantaneous power at every pre-defined time interval.

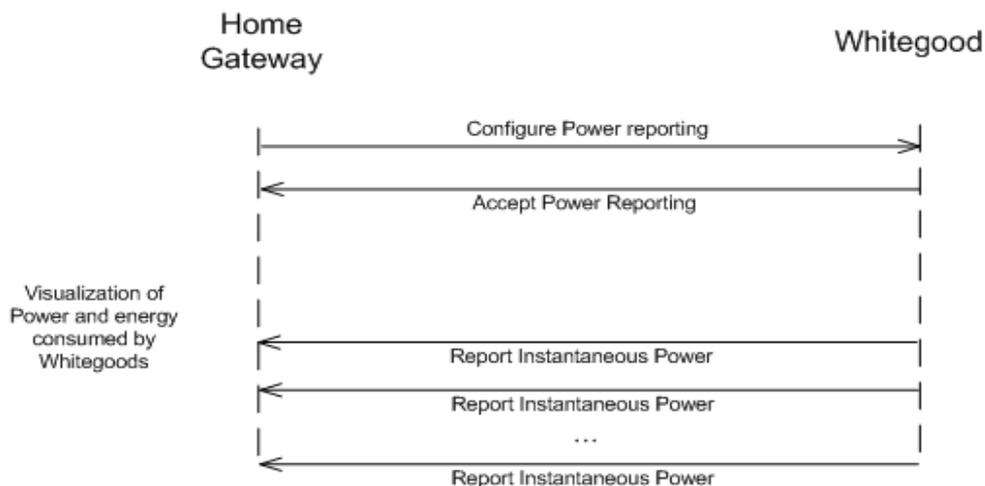


Figure 10 : Configuration of instantaneous power reporting on appliances.

In the **Figure 11** are shown the sequences diagram for the visualization of price associated to a power profile. Here, the Whitegood sends to the Home Gateway a power profile notification; moreover the home user can check the price at a specific time. When the gateway receives the power profile notification and the “get price” command, it calculates the price and sends back it to the Whitegood.

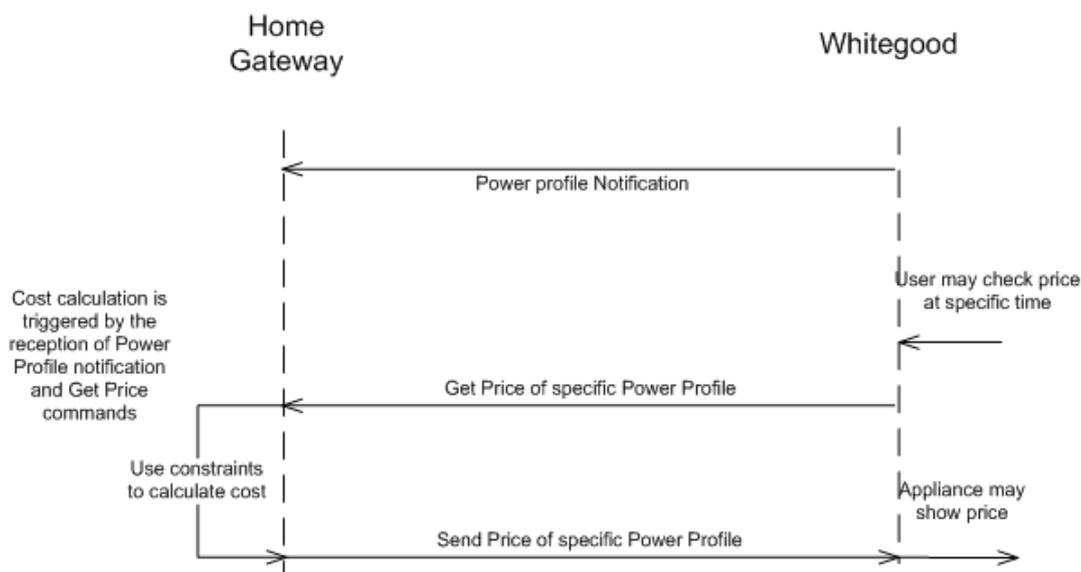


Figure 11 : Visualization of price associated to a power profile

#### 4.3.1.4 Appliance regulation

The description of these interactions is reported in different examples of sequence diagrams: the interactions and the message flows represent only example of possible interactions: please notice that they might be different according to implementations and feature support.

#### 4.3.1.5 E@H control disabled

In this paragraph, it will be shown how an interaction occurs between the user and the smart appliance when the Energy@home control is disabled.

In this case, the smart appliance will not be scheduled and controlled by the Home Gateway, even if the user will have the awareness of what occurs.

In the example of **Figure 12** it is described both a possible interaction with the user and the expected messages exchanged between the smart appliances and the Home gateway.

In the first section of the sequence diagram (*each section in figure is separated by a green line*), the smart appliance is already programmed and the user changes an application setting (*i.e. select a cycle*).

Due to this change, the smart appliance sends to the home gateway the power profile notification and the “get price” command.

The home gateway answers sending the power price to the smart appliance that shows the price on the display to the user.

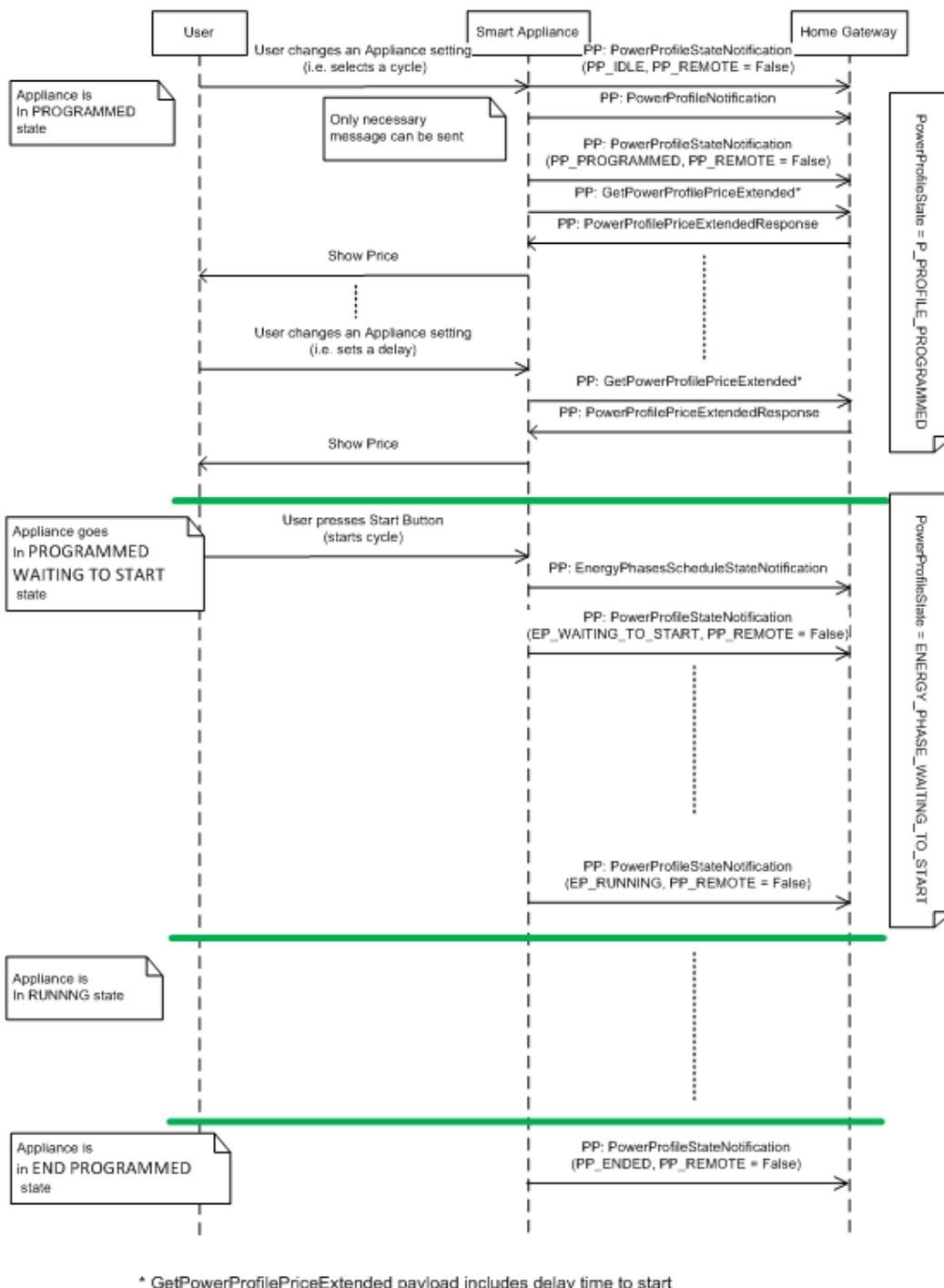


Figure 12 : E@H control disabled: example of sequence diagram with user interaction.

Another user action shown in the first section of the sequence diagram is related to the “delay insertion” for a cycle of the smart appliance; again, the “get price” procedure above described is repeated.

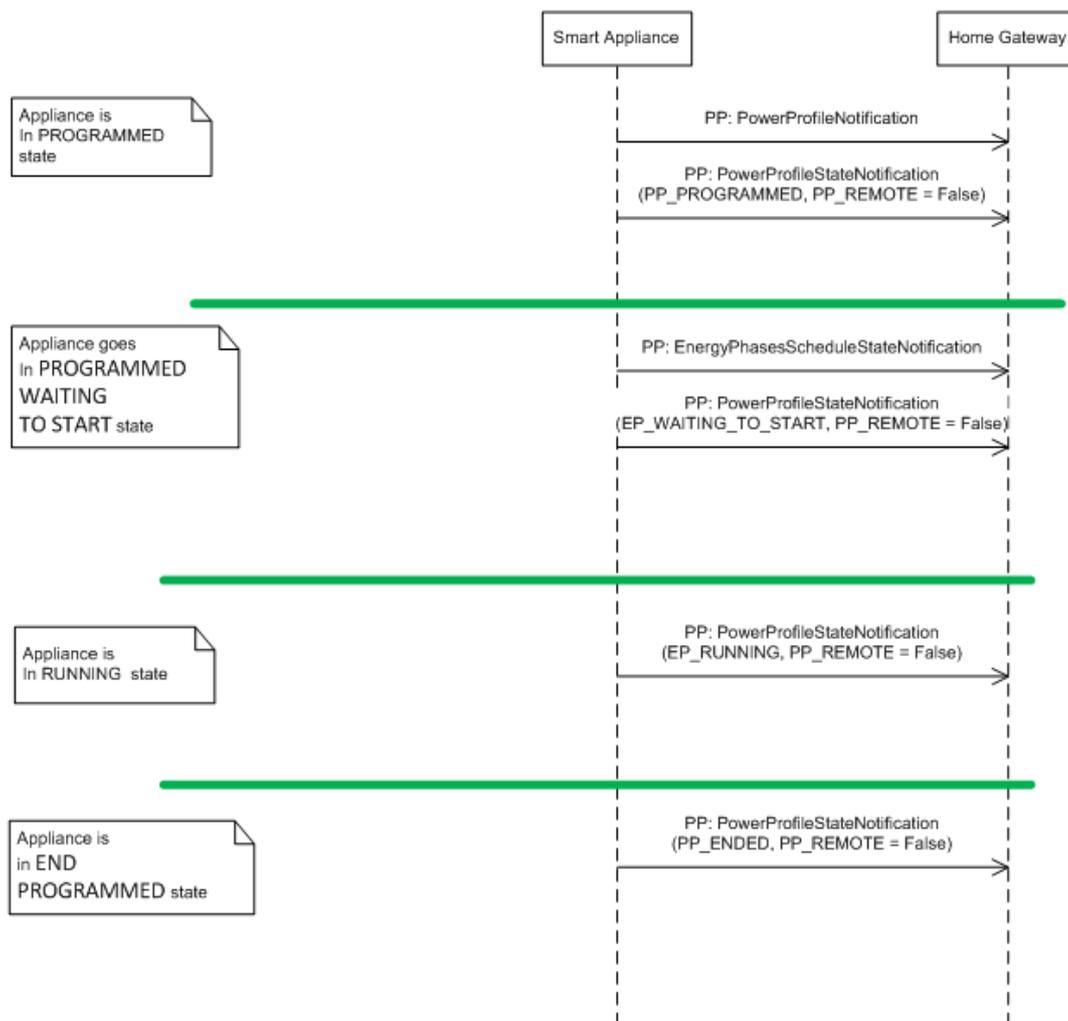
In the second section of **Figure 12** the smart appliance is in “waiting to start” state and the user press the “start button”.

Then the smart appliance sends to the gateway the notification about its new state change (*from waiting to running*).

The third section simply shows that the smart appliance cycle is in progress.

Finally, in the fourth section the smart appliance has finished its cycle and communicates to the home gateway this state.

The **Figure 13** shows an example of sequence diagram with E@H control disabled. The sequence diagram is very similar to the one before, with the difference that here there is not the user interaction and the smart appliance simply communicates its various states to the home gateway.



\* GetPowerProfilePriceExtended can be generated any time by SA if a PP is active

Figure 13 : E@H control disabled: example of sequence diagram.

4.3.1.5.1 Overload condition.

When the total instantaneous power used by the house (measured in kW and described by the attribute *InstantaneousDemand* in case of ZigBee) exceeds the contractual limit (described by the attribute *DemandLimit*), the home gateway starts to send periodically to the smart appliances (e.g. every 60 seconds) an “Overload Warning” alarm (**Figure 14**).

This alarm will be reset by sending once the “End of Overload Warning” message when the total instantaneous power returns below the limit.

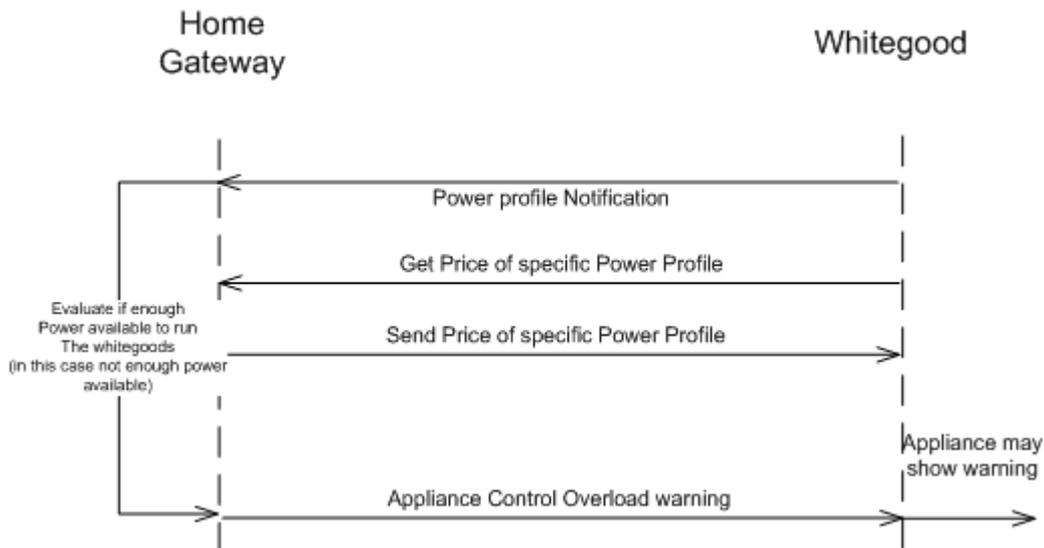


Figure 14 : E@H control disabled: Overload warning.

**4.3.1.6 E@H control enabled**

In this paragraph it will be analyzed the cases in which the Energy@home control is enabled that is the case where the full set of Energy@home features is running: the appliances can be automatically scheduled according to the needs of the user and pre-emptive and reactive control on the devices is allowed.

*4.3.1.6.1 Pre-emptive control (scheduling)*

In **Figure 15** is reported an example of sequence diagram with user interaction with the E@H control enabled. With the smart appliance in “Programmed” state the user changes a setting and the appliance notifies this change at the Home Gateway which answers showing to the user the optimal start. Then the appliance requests the related price at the Home Gateway which answers showing it to the user.

In the second section (*each section in figure is separated by a green line*) the smart appliance is in “waiting to start” state and the user press the “start button”, with the option to choose between the “*immediately start (Forced by user)*” or “*accept the E@H scheduling*” (delegating control).

Then the smart appliance sends to the gateway the notification about the user choice (*if and how switch from waiting to running state*). In the third section, when the appliance starts its running state, it sends to the Home Gateway its power profile and this replies giving the “*acknowledge to proceed*” or “*change the schedule*”.

Finally, in the fourth section, when the appliance finishes its work, it notifies at the Home Gateway the “*job concluded*” state in order to make available this “requested energy” for other loads inside the home.

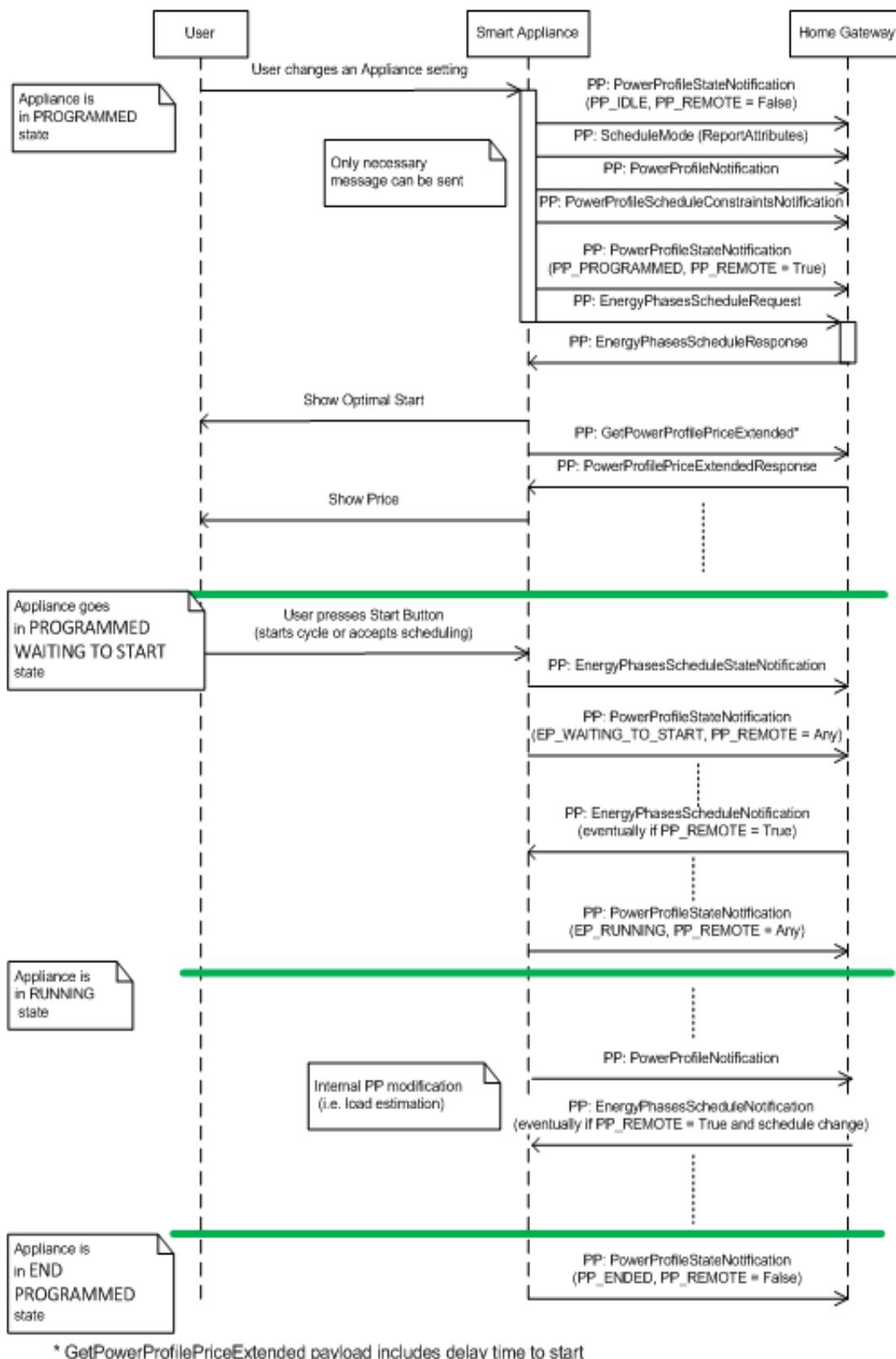
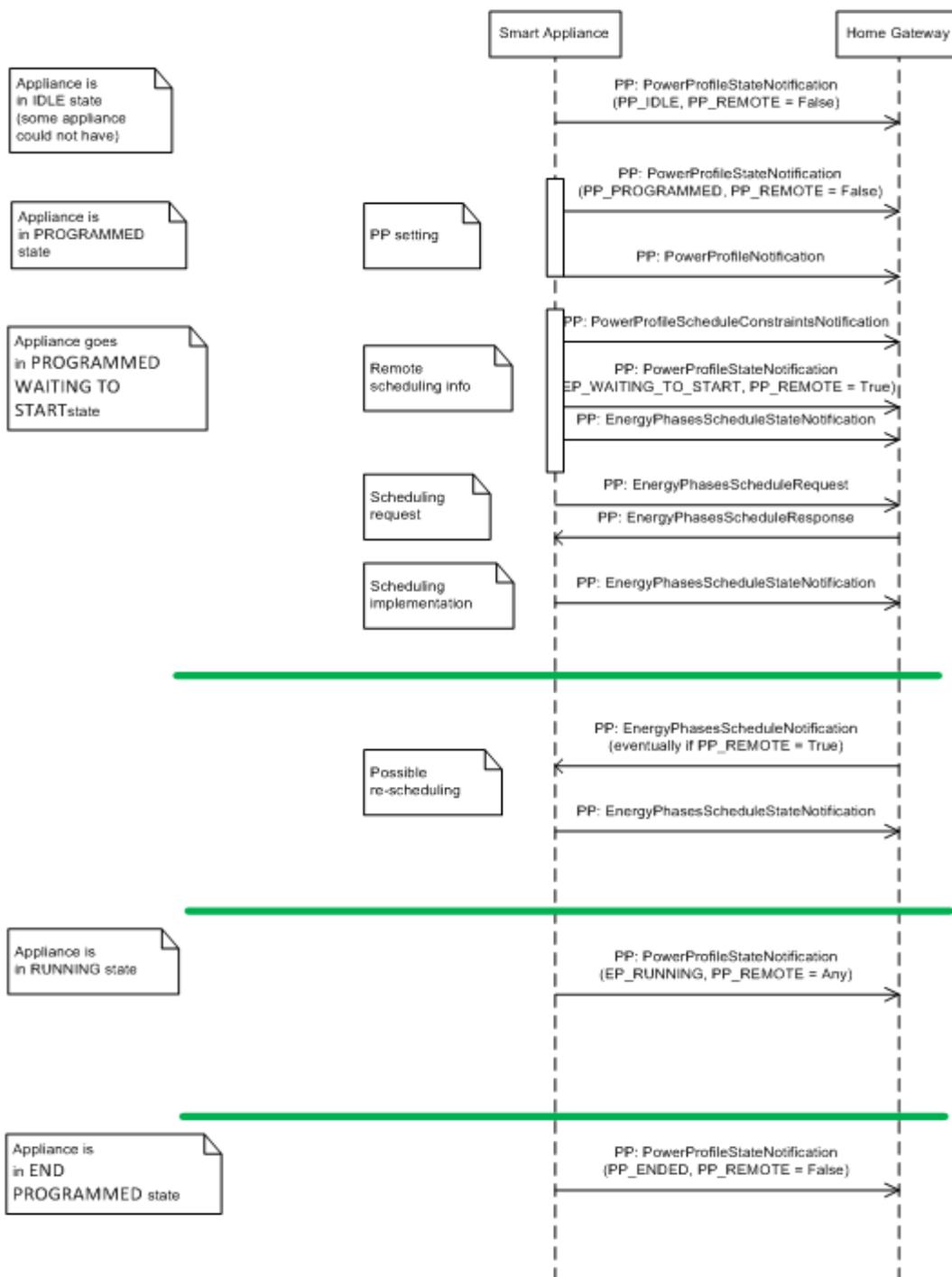


Figure 15: E@H control enabled: example of sequence diagram with user interaction.

In **Figure 16** is reported another example of sequence diagram of the smart appliance interface. The sequence diagram is very similar to the one before, with the difference that here there is not the user interaction and the smart appliance communicates and negotiates its various states with the home gateway. This is the classic example of machine-to-machine communication, where

the optimization of the power consumption is completely delegated to a communication protocol between machines.



\* GetPowerProfilePriceExtended can be generated any time by SA if a PP is active

Figure 16 : E@H control enabled: sequence diagram without user interaction

There are also special cases in which the appliance, although under the control of Energy@home system, can decide whether to stop or not its cycle. A classic example is the washing machine which could not accept an interruption because the clothes in the wash may be damaged.

4.3.1.6.2 Reactive control (overload management)

In the condition of E@H enabled, the **Figure 17** shows the sequence diagram of reactive control (overload management). A whitegood is running a cycle and, at a certain instant, the user activates a no-smart device.

As a result of this action, the home gateway detects that the total power consumption exceeded the total available one, and it sends a first warning. If the total power consumption is still exceeding the total available one, the home gateway sends a pause command to the whitegood.

The whitegood checks if the pause can damage something (e.g. the clothes in the washing machine may be damaged) and it acts with the following decisions:

- If the pause can damage something, it continues the cycle and it will pause it as soon as possible.
- If the pause doesn't damage anything, it accepts the pause command.

Then, when the home gateway will detect that the total power consumption is less than the total available one, it will send a "resume" command and the whitegood will continue with its cycle.

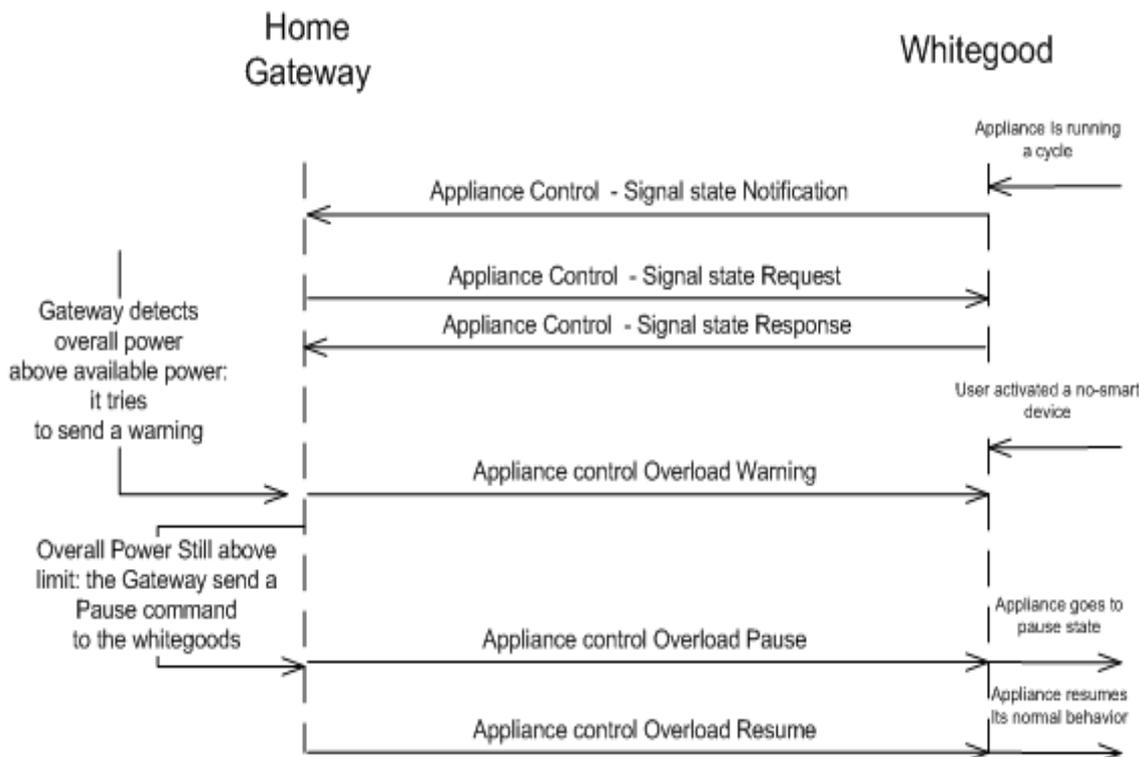


Figure 17 : E@H control enabled: sequence diagram of reactive control (overload management).

4.3.1.7 Self-Production and Primary meters in Energy@home application layer

This chapter describes the standard configuration of residential on-site generation plant (i.e. photovoltaic panel, mini wind turbine...).

As well known, the Feed-In tariff (FIT) schemes have been successful in many countries around the globe. Such schemes are based on private power producers feeding all electricity they generate into the public grid against payment of a pre-determined price that is guaranteed for a set period of time.

However, local utility companies are concerned that a large amount of privately generated renewable energy could have a negative impact on the grid. In fact, there is no need to feed any self-generated electricity back into the grid. It makes more sense for households to themselves use the generated energy.

Instead of providing incentives for energy being fed into the grid, it therefore makes more sense to incentivize self-consumption of self-generated electricity. So, today many local utility companies offer a higher energy price if the prosumer self-consumes its self-produced energy instead of sell it to the grid.

Given the above, in the **Figure 18** is shown the use Primary and Self-Production meters in E@H. The energy production of any on-site generation plant is monitored and recorded by a smart meter (it is marked in **Figure 18** with the label M2 and the self-produced power with the vector P).



Figure 18 : Use Primary and Self-Production meters in E@H.

In such case the primary smart meter (M1) monitors and records both the energy picked-up from the power distribution network (vector E) and the energy put into it (vector U).

The home consumption of energy (vector C) is calculated as the contribution of both a part from the on-site generation plant and from the power distribution network.

The vector C is so calculated:  $C = E + (P - U)$ .

**E:** Primary meter M1      *CurrentSummationDelivered*

**U:** Primary meter M1      *CurrentSummationReceived*

**P:** Self-Production meter M2      *CurrentSummationReceived*

Instantaneous power data may be different according to the **Table 3**, where the “Data Quality ID” identifies the data quality.

<i>Device</i>	<i>Data Quality ID</i>
All Data Certified	0x0000

Only Instantaneous Power not Certified	0x0001
Only Cumulated Consumption not Certified	0x0002
Not Certified data	0x0003

Table 3: Data Quality identification

In the case "All Certified Data", instantaneous power data is monitored from the meter M1 or M2 and the refresh rate will not be in real time.

In the case “*Only Instantaneous Power not Certified*”, instantaneous power data is measured by an additional meter installed on the power line that supplies the user of the client, measuring the vector **C** in a real-time frequency.

In **Figure 19** is shown the sequence diagram about the management of the process for Self-Production and primary meter in Energy@home.

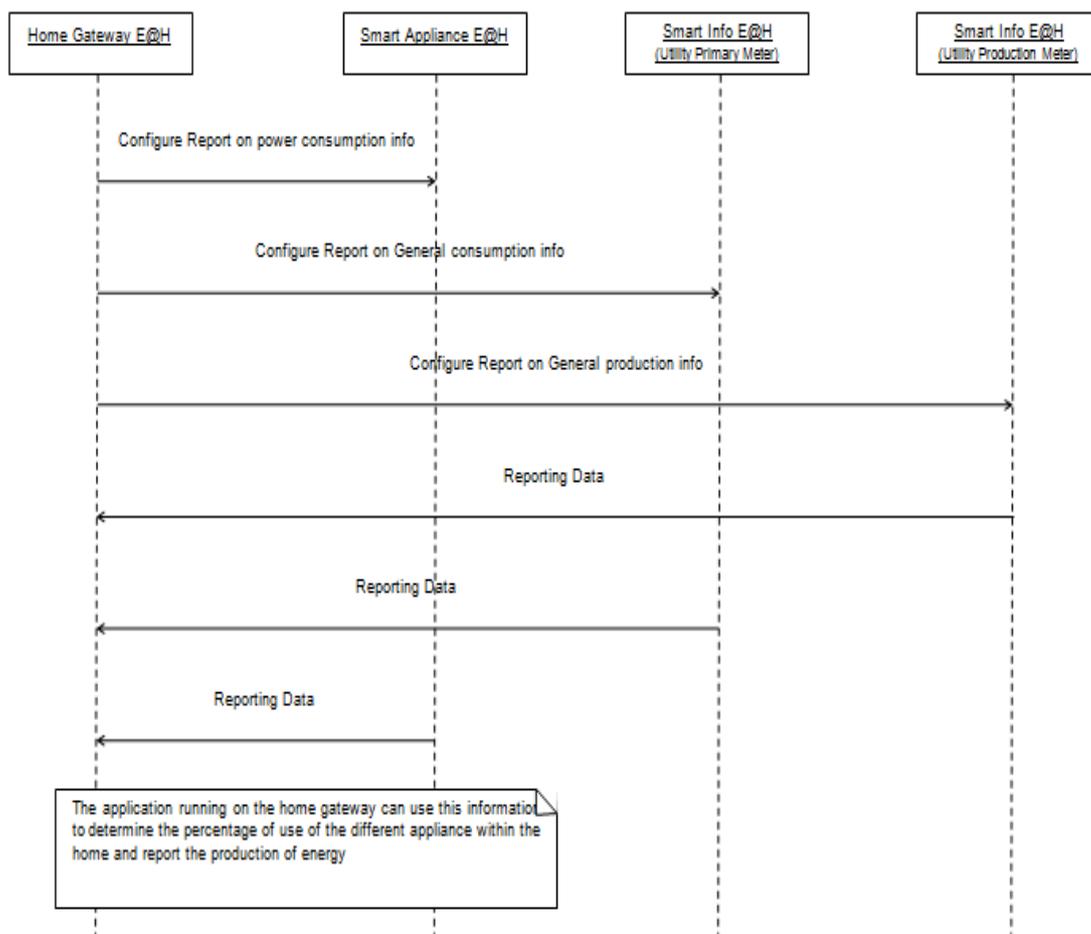


Figure 19 : Management of Self-Production and primary meter in Energy@home.

As first step, the home gateway requests a specific report on power consumption info at a specific smart appliance and requests a report on the general consumption info at the Smart Info M1. Finally, the home gateway requests a report about the general power self-production info at the Production meter M2.

Each of the entities questioned by the home gateway responds to requests by sending a reporting data. The application running on the home gateway uses these report information to determine the percentage of use of the different appliances within the home and also report the self-production of energy.

### 4.3.2 BeyWatch Application Layer

This section describes the application layer as it has been defined in BeyWatch [10].

BeyWatch uses an Appliances Management Framework that allows the management of appliances independently from the manufacturers and the communication technologies. In this way a user is provided with a wider range of possible appliances to purchase, not tied to only one manufacturer for all the appliances that he needs.

In BeyWatch, the communication between the home area network and the appliances use M2M communications, and as the Appliance's Management Framework allows the appliances to be independent from the communications technology, although they are mainly ZigBee and Wi-Fi, it is possible to use others.

It is used an interface with the agent. The agent is in charge of controlling and monitoring the appliances, for that it uses Appliance's APIs, which provide simplicity and consistency and a data model for each type of appliance along with the methods of data interchange. This agent connects with the manufacturers which have the control of the appliances, and schedules the operation of the devices.

One of the objectives is to allow different manufacturers to connect to the HAN, for that it is used OSGi technology that provides a smooth integration of the different modules composing the framework.

For FINSENY's layered architecture, the appliances' control was done independent from higher levels. This makes it easier for a user to buy a new appliance with the manufacturer that suits his necessities better, he doesn't have to worry about the integration of the appliance because is the service provider who installs it in the framework with the correct module. Then the user is able to access the information about the appliances status.

For more details about the features and the experimental results obtained with the system BeyWatch, please refer to the FINSENY deliverable D8.2 "*Experiments and evaluation*" [8] and D8.3 "*Selected domain specific enablers specification*" [9].

#### 4.3.2.1 Appliance Management Framework

The main objective of the Appliance's Management Framework is to make appliances management independent from the manufacturer and M2M communications technology they use, facilitating monitoring and control of the appliances from the application layer (normally developed by service providers).

In BeyWatch, the Appliance's Management Framework abstracts monitoring and control procedures of real appliances from the **Agent Controller** (the module in charge of the scheduling and actuation over the devices belonging to the Building Energy Management System – BEMS) depicted in Figure 20, which communicates with the Appliance's management framework through a set of well-defined APIs.

A basic representation of the Appliance's Management Framework layered architecture is presented in the picture below:

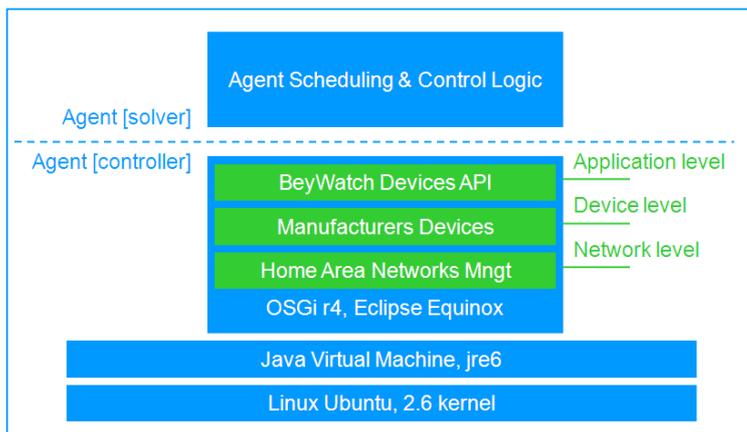


Figure 20: Appliance Management Framework – Global view

The monitoring and control of each type of appliance is implemented in 3 different layers:

- Base driver layer:** This is the layer in charge of communicating through the corresponding hardware interface of the Agent towards the home area network the appliance is connected to. Over the corresponding network, M2M communications between the framework and the physical appliance use the corresponding protocol (either public or proprietary, depending on the appliance manufacturer). Appliances in BeyWatch communicate through Wi-Fi and ZigBee but other communication protocols could be easily added by including the necessary base drivers.

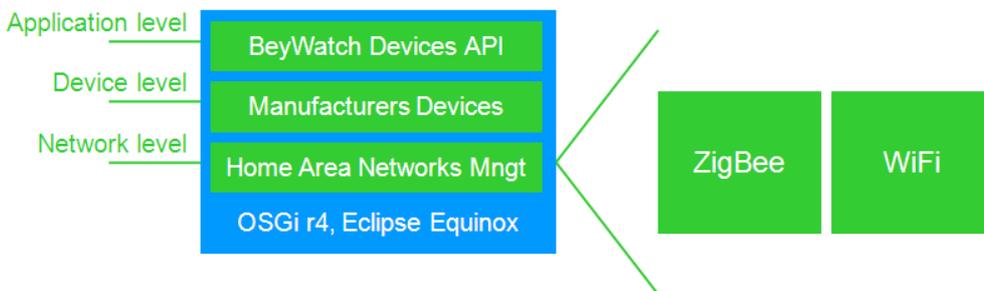


Figure 21: Appliance Management Framework – Base drivers

- Manufacturer driver layer:** This layer provides the manufacturer implementation for accessing the various functionalities the appliance offers for its integration in more complex systems. In BeyWatch there were 3 different manufacturer’s drivers: Gorenje, Fagor and UniPa-EDF, which provide the implementation for the monitoring and control of the Refrigerator, Washing machine and Dishwasher, the Combined Photovoltaic System (CPS), the smart meter from EDF and external watchers (energy aware smart plugs) respectively. Other device manufacturers could be easily added by integrating the corresponding plug-ins at the manufacturer level layer.

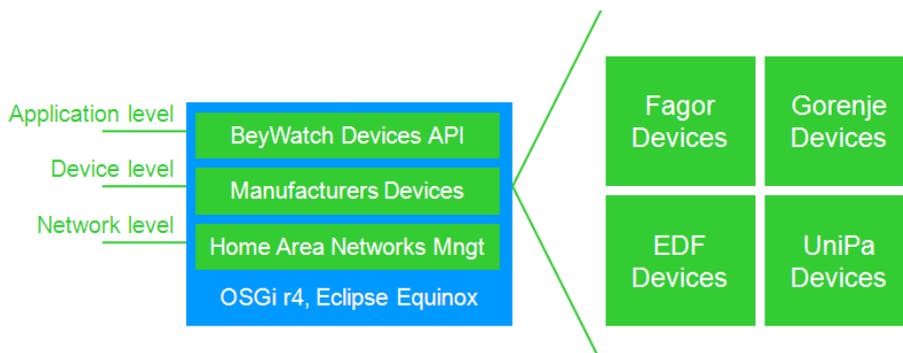


Figure 22: Appliance Management Framework – Manufacturers’ drivers

- BeyWatch appliance driver layer:** At the top of the framework it is the BeyWatch appliance driver layer, where the implementation of the BeyWatch appliance’s APIs (interface with the Agent Controller in the BEMS) is provided. There is one driver per appliance independently of its manufacturer. Other devices could be easily added by integrating the corresponding plug-ins at the device API layer.

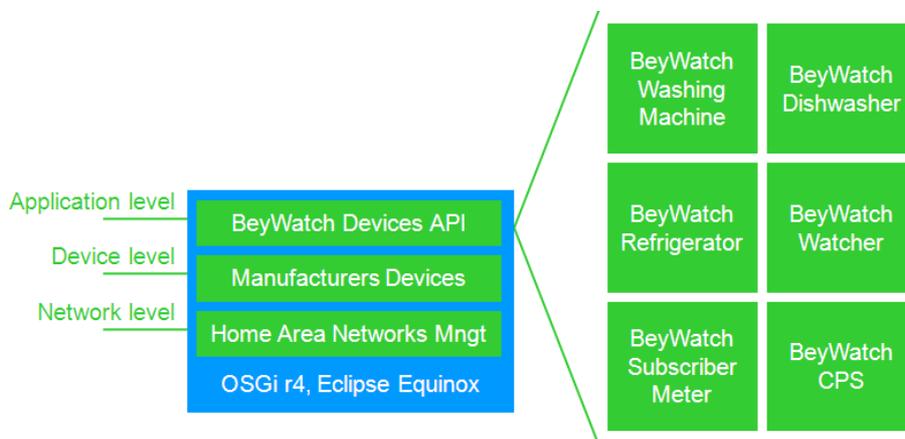


Figure 23: Appliance Management Framework – BeyWatch appliances’ drivers

Appliances’ APIs are intended for monitoring and control of the various appliances from the BEMS. I.e., the APIs, which the BeyWatch Agent uses to monitor the status, get information about instant power demand (and energy spent in some cases) and control of the appliances.

APIs definition has been guided by four principles:

- Simplicity** - the APIs necessary to fulfil devices monitoring and control from the Agent were defined. Extra functionality the devices might offer was not included.
- Clarity** - method and parameter names are sometimes lengthy but clearly describe the semantics and are consistent with standard Java language coding conventions.
- Consistency** - the same approach is used in all the interfaces and uniform naming conventions have been used for methods and parameters.
- Generalizations when applicable** - use of interface inheritance to denote associations between interfaces.

The Appliance’s Management Framework uses OSGi technology. The modularity provided by OSGi has allowed integrating smoothly every module developed for BeyWatch appliance’s management without interfering with the rest of modules already deployed in the framework.

Modules in OSGi are composed of bundles, each bundle providing a specific functionality (or ‘service’, according to OSGi terminology). The Appliance’s Management framework makes use of some of the basic services provided by the equinox implementation ([11]) of OSGi R4 specifications (such as the http, configuration admin, event handler and device manager services).

#### 4.3.2.2 Application layer main characteristics in BeyWatch

The main technical characteristics of this application layer in BeyWatch are:

- **Specification of functional APIs for BeyWatch appliances:** BeyWatch appliances APIs specify a data model for each type of appliance (washing machine, dishwasher, refrigerator, CPS, smart meter and watcher) and the methods for data interchange (related to energy and control) between the service provider and actual appliances.
- **Development and integration of the various appliances' bundles.** The management of each appliance involves the use of several bundles. Some of them are specific but some other implement standard functionalities that can be reused afterwards. For example, M2M communications with the CPS, the smart meter and the watchers make use of ZigBee standard application profiles Home Automation and Smart Metering. This is done by using the standard methods in ZCL library, what is modelled in OSGi by the use of a single service (ZCS) that can be used by any manufacturer bundle that needs to communicate over ZigBee with standard application profiles. This is the case in BeyWatch for EDF and UniPa manufacturer bundles.
- **Successful integration with the Building Energy Management System (BEMS),** thanks to the correct implementation of appliances APIs.
- **Test OSGi bundles,** which are fed with BeyWatch appliances APIs to run unitary tests at a parameterized frequency and log the results in text files. This test bundles can be reused in the future for testing new appliances from different manufacturers that implement BeyWatch appliances APIs.
- **Scheduled control bundle.** This bundle is meant for scheduling the operation of devices of the 6 types supported right now in BeyWatch. The appliances can be controlled (with the needed parameters) at the desired time (in intervals of 15 minutes).
- **Energy aware Home Gateway Standardization.** BeyWatch has share its view on energy management services with the Home Gateway Initiative (HGI) [12] and OSGi forums [13] with the bundles for managing the intelligent metering device, the networked, energy-aware white goods and the RES/CPS system, while at the HGI it has contribute to the design and specification of the energy management functionalities to be added in the next version of the standard for Telco's home gateways.

#### 4.3.2.3 Application layer design principles in BeyWatch

The integration of appliances from different manufacturers and different M2M communications technologies is always hard. A layered architecture like the one proposed in FINSENY for smart buildings and also used by the Appliance's Management Framework is very useful to make appliances control independent from higher layers (service logic and provisioning), and it is even something necessary until M2M communications are standardized. The main design principles supporting this layered architecture used in BeyWatch, and applicable also to the proposed architecture in smart buildings in FINSENY, are:

- **Modularity.** It is a key point for developing configurable products. In the home area, it is the user at the end who will chose which are the appliances and devices that better suits his needs and thus, a framework like the one used in BeyWatch can be a good approach for FINSENY: The user buys a new appliance, the service provider installs in the framework the module needed to monitor and control it, remotely and without disturbing the user with technology integration.
- **Reliability.** The appliance management framework needs to be reliable and stable: the user must have at any time reliable information about appliances status and consumption. If communications are lost, he has to be aware of it.

- **Scalability.** The appliance management framework needs to be extensible to integrate as many manufacturers as possible and, given the heterogeneity that exists in M2M communications technologies, integrating as many technologies as needed.
- **Openness.** Clear and complete functional APIs must be provided to open the framework to manufacturers. The implementation of the API may be private but they can extend this APIs with much other functionality to differentiate from their competitors.
- **Platform independent.** There are several trends regarding the hosting of the energy management system. The framework could run in an ‘energy box’, in a general purpose box (for the provision of other services apart from energy management), in a home gateway (including broadband connectivity), in a module coupled to the utility smart meter... all these considerations make platform independence a good point.

### 4.3.3 Data Center application layer

In the data centers the application layer has to manage a constant tradeoff between QoS (*Quality of Service*), security and power efficiency (**Figure 24**).

In the FINSENY D4.1 “*Smart Buildings ‘scenario’ definition*”, chapter 6, was proposed six use cases about the data center, where this constant tradeoff was highlighted.

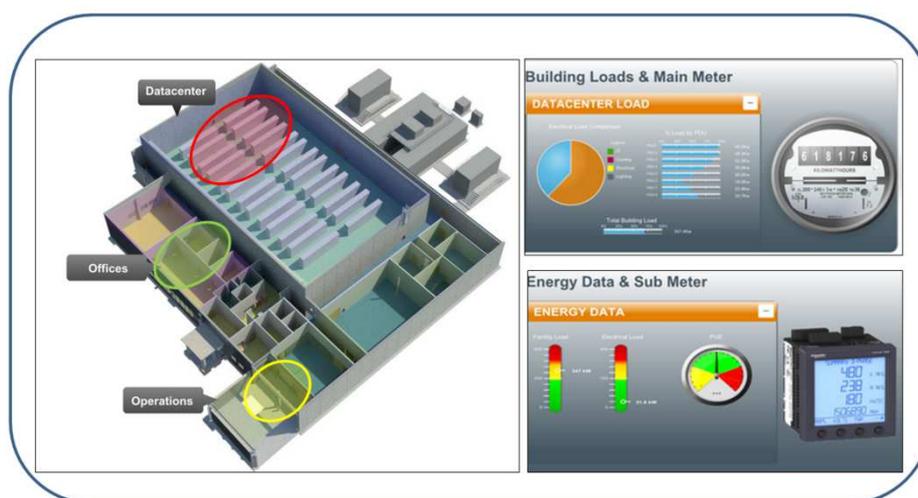


Figure 24: Data Center dashboard application

In this paragraph, the data center is seen by its own internal, i.e. what actions at the application level must be made in order to optimize the power consumption without compromising the quality of service offered to the outside.

#### 4.3.3.1 Data Center Infrastructure Management (DCIM)

Data Center Infrastructure Management (DCIM) is an emerging (2012) new form of data center management which extends the more traditional systems and network management approaches to now include the physical and asset-level components.

DCIM unifies systems onto a single common platform, enabling data collection from not only different systems, but completely different silos of effort.

DCIM tools bridge the gap between IT and Facilities by providing a real time monitoring and management platform for all interdependent components within the Data Center through use of hardware, software, and various sensors/actuators.

DCIM provides the following benefits:

- Access to accurate, actionable data about the current state and future needs of the data center,
- Standard procedures for equipment changes,
- Single source of truth for asset management,
- Better predictability for space, power and cooling capacity means increased time to plan
- Enhanced understanding of the present state of the power and cooling infrastructure and environment increases the overall availability of the data center,
- Reduced operating cost from energy usage effectiveness and efficiency.

There are three DCIM application categories of real-time monitoring systems in the data center:

Building Management System (BMS): a BMS is typically a hardware-based system utilizing Modbus, BACnet, OPC, LonWorks or Simple Network Management Protocol (SNMP) to monitor and control the building mechanical and electrical equipment. These are often custom-built systems priced on the number of individual data points being monitored (a data point might be the output load on a UPS or the return temperature on a computer room air conditioner unit). In some cases, the BMS system is extended into the data center to monitor and control power and cooling equipment.

Network Management System (NMS): an NMS is typically a software-based system utilizing SNMP to monitor the network devices in the data center. Network devices can usually be auto-discovered, so installation can be automated to some degree.

Data Center Monitoring System (DCMS): a DCMS can be hardware-based and/or software-based and is used to monitor a data center or computer room. Device communication is typically done using SNMP, although some data center monitoring systems can also communicate using Modbus, IPMI or other protocols.

In the next paragraph these three real-time application categories will be analyzed in their Management components.

#### **4.3.3.2 Data Center Application Management components**

As shown in the **Figure 25**, the Application Management Layer for the data center must find ways to tie together at least seven different needs from different sources into an overarching monitoring and analytics environment that lets them make better decisions, optimize the use of their capital spending, and increase the overall efficiency related not only to power but also to capacity, utilization, and operational aspects.

The application management layer itself can also provide redundancy across multiple data centers, freeing data center managers to use spare capacity in their data centers for lower-priority applications when they are not in use for disaster recovery (which is most of the time), eliminating the need for costly redundant data center configurations, large UPS devices and reliability build-outs and decreasing the need for each data center's redundancy.

The seven application components shown in **Figure 25** have the following functionality:

Asset Management: identifies all assets within the IT and facilities domains, shows their location, calculates power and cooling needs, performs "what-if" scenarios when adding, moving or changing servers.

Building Management: manages and controls the cooling system, physical security, fire protection, leak detection, and CCTV monitoring.

Power Management: sometimes included as part of Facilities Management, this discipline meters and manages the delivery of power to the servers from primary and alternate sources.

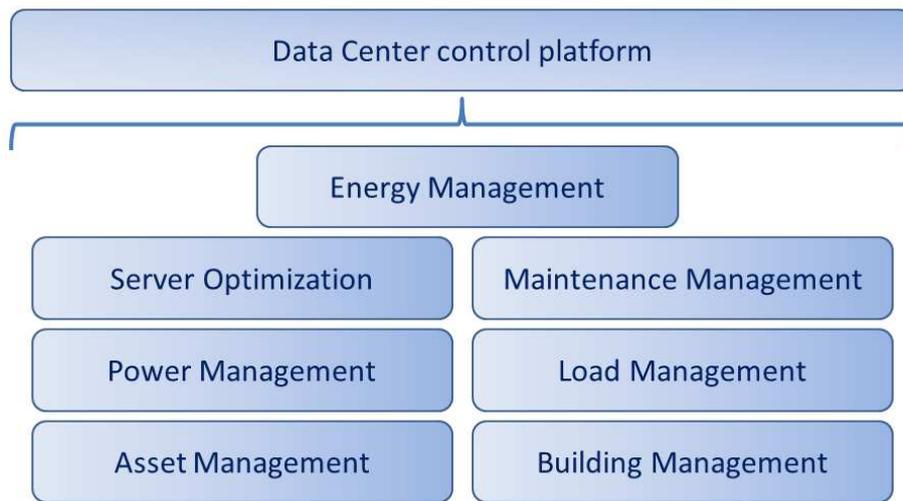


Figure 25: Application Management components

**Load Management:** analyzes the current and forecasted server, power and cooling demand along with utility contract rates to balance the load across multiple data center sites.

**Server Optimization:** analyzes CPU utilization and application criticality in conjunction with server temperatures to adjust candidate servers to operate more efficiently through reduced power draw.

**Maintenance Management:** handles work order ticketing using prescribed work flows for submittal, creation, tracking, expenditures, lessons learned, and spare parts availability.

**Energy Management:** optimizes the energy usage profile against forecasted demand, contracted rates, and alternate energy source rates.

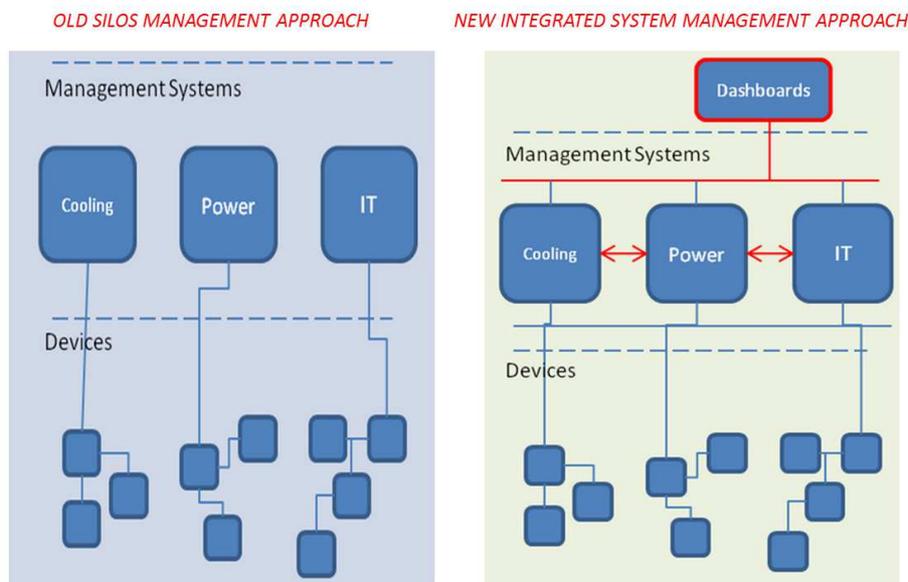


Figure 26: Old and new DCIM approach

For most data centers, the current state of what could be classified as data center infrastructure management are a collections of independent and “closed” monitoring, planning and control systems. Typically, there is at least one software tool for each of the physical sub-systems: electrical, cooling and mechanical, IT infrastructure, IT assets, etc.

In these DCIM approach (**Figure 26**) the information is kept in discrete silos to meet the needs of a specific expert user group but this would require different software to application level that often were incompatible to communicate with each other, forcing the supervisors to a limited vision of the whole situation in the data center.

A new and more effective approach is to leverage open communication protocols and to architect the communication pathways so data can be easily gathered into any expert system and detailed information can be exchanged between systems.

#### 4.3.3.3 Open Data Center Infrastructure Management application

Today there are many companies that develop DCIM commercial products. But recently (2012) is available “openDCIM” (<http://www.opendcim.org/>), a web based Data Center Infrastructure Management application it means many different things to many different people, and there is a multitude of commercial applications available.

openDCIM does not contend to be a function by function replacement for commercial applications. Instead, openDCIM covers the majority of features needed by the developers - as is often the case of open source software. The software is released under the GPL v3 license, so one can take it, modify it, and share it with other project partners.

These the most important features of openDCIM:

- Provide complete physical inventory (asset tracking) of the data center
- Support for Multiple Rooms (Data Centers)
- Management of the three key elements of capacity management - space, power, and cooling
- Basic contact management and integration into existing business directory via UserID
- Fault Tolerance Tracking - run a power outage simulation to see what would be affected as each source goes down
- Computation of Center of Gravity for each cabinet
- Template management for devices, with ability to override per device
- Optional tracking of cable connections within each cabinet, and for each switch device
- Archival functions for equipment sent to salvage/disposal
- Integration with intelligent power strips and UPS devices - APC, Geist Manufacturing, Liebert, and Server Technologies. Easy to update with OIDs for other manufacturers.
- Open Architecture - All built on a MySQL database for easy report building, or export to other applications

The system requirements for openDCIM are

- Web host running Apache 2.x (or higher) with an SSL Enabled site.
- MySQL 5.x (or higher) database
- PHP 5.3 (or higher)
- User Authentication
- Web Based Client

#### 4.3.4 ReActivHome project application layer

The ReActivHome project [14] developed a software system supporting energy management applications adaptable to any home environment. It focuses on the adjustment of the electric energy consumption and production in order to maximize energy usage efficiency, which is seen as a compromise between energy cost and overall comfort. It dealt with the followings issues:

- Integration within a generic home automation architecture

- User interfaces and social acceptability
- Automatic recognition of equipment by sensors, including an ammeter, in a “plug&play” mechanism
- Anticipative/reactive energy management system based on an optimization solver

The ReActivHome system architecture introduced the abstraction layer of physical home entities (rooms, electrical equipment) called “Home Abstraction Layer (HAL)” that has been further developed and generalized in the FINSNEY project. The physical coupling between the ReActivHome system and the home is supported by shared sensors and actuators. Among these is a smart plug for monitoring and controlling mains-connected devices, connected by Zigbee. Supported by a hybrid optimization engine programmed in Java using ILOG CPLEX, an energy control plan is generated 24 hours in advance. This schedule can then be actuated directly through HAL by the ReActivHome system or be presented as energy management guidelines via adapted user interfaces.

The ReActivHome energy management system of has been installed in different demonstrators to validate its performances and demonstrate it as a proof of concept.

#### **4.3.4.1 Principle of control mechanism**

An important issue in building energy management problems is the uncertainties in the model data. For instance, solar radiation, outdoor temperature or services requested by inhabitants may not be predicted with accuracy. In order to solve this issue, a 3-layer architecture is proposed, composed with a local layer, a reactive layer and an anticipative layer.

The figure 2 gives a description of these layers and the mechanisms which link each one to the other.

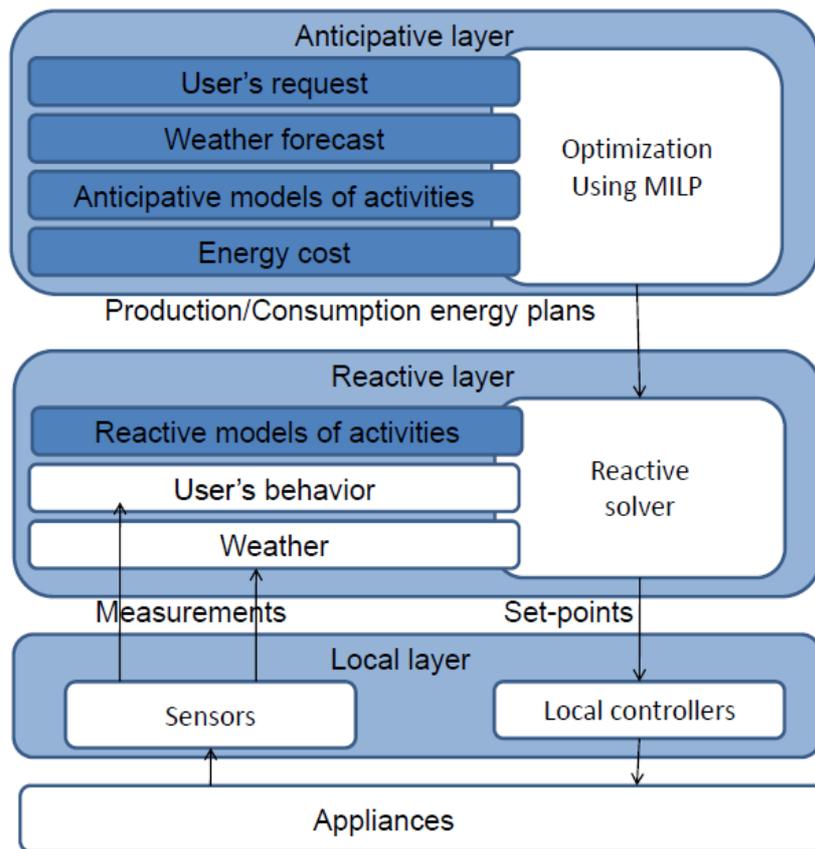


Figure 27: Control Layers description

The anticipative layer is responsible for scheduling end-user, intermediate and support services taking into account predicted events and costs in order to avoid as much as possible the use of the reactive layer. The prediction procedure forecasts information about future user requests but also about available power resources and costs. Therefore, it uses information from predictable services and manages continuously modifiable and displaceable services. This layer has slow dynamics (e.g. a 1h sampling time) comparing to other layers and includes predictive models with learning mechanisms, including models dealing with inhabitant behaviors. This layer also contains a predictive control mechanism that schedules energy consumption and production of end-user services several hours in advance. This layer computes plans according to available predictions. The sampling period of the anticipative layer is further noted  $\Delta$ . This layer relies on the most abstract models.

The objective of the reactive layer is to manage adjustments of energy assignment in order to follow up a plan computed by the upper anticipative layer in spite of unpredicted events and perturbations. Therefore, this layer manages modifiable services and uses information from observable services. This layer is responsible for decision-making in case of violation of predefined constraints dealing with energy and inhabitant comfort expectations. The set-points determined by the plan computed by the upper anticipative layer are dynamically adjusted in order to avoid user dissatisfaction. The control actions may be dichotomous in enabling/disabling services or more gradual in adjusting set-points such as reducing temperature set point in room heating services or delaying a temporary service. Actions of the reactive layer have to remain transparent for the plan computed by the anticipative layer: it can be considered as a fast dynamic unbalancing system taking into account actual building state, including unpredicted disturbances, to satisfy energy, comfort and cost constraints. If the current state is too far from the computed plan, the anticipative layer has to re-compute it.

The local layer is composed of devices together with their existing local control systems generally embedded into appliances. It is responsible for adjusting device controls in order to

reach given set points in spite of perturbations. This layer abstracts devices and services for upper layers: fast dynamics are hidden by the controllers of this level. This layer is considered as embedded into devices and is not part of the application layer proper as described here.

This section mainly deals with the scheduling mechanism of the anticipative layer that computes anticipative plans for the building energy management problem.

#### 4.3.4.2 Principle of regular/centralized solving approach

##### 4.3.4.2.1 Services Modeling

Services are understood in the sense given in section 4.4, as generic functionalities of the building that are targets for being controlled by the application layer.

The services modeling can be decomposed in two aspects: the modeling of the behaviors with operational constraints, which depends on the types of involved models, and the modeling of the service performances, which depends on the types of service. Whatever the model, it has to be defined on over a time horizon  $K \times \Delta$  for anticipative problem solving composed of  $K$  sampling periods lasting  $\Delta$  each.

##### 4.3.4.2.1.1 Modeling behavior of services

In order to model the behavior of the different kinds of services, three different types of models have been used: discrete events are modeled by finite state machines, continuous behaviors are modeled by differential equations and mixed discrete and continuous evolutions are modeled by hybrid models that combine the two previous ones. The case of finite state machines is then presented.

A Finite State Machine (FSM) dedicated to a service, denoted  $SRV$ , is composed with a finite number of states  $\{\mathcal{L}_m; m \in \{1, \dots, M\}\}$  and a set of transitions between those states  $\{\tau_{p,q} \in \{0,1\}; (p,q) \in S \subset \{1, \dots, M\}^2\}$ . Each state of a service  $SRV$  is linked to a phase characterized by a maximal power production or consumption.

A transition triggers a state change. It is described by a condition that has to be satisfied to be enabled. The condition can be a change of a state variable measured by a sensor, a decision of the anticipative mechanism or an elapsed time for phase transition. If it exists a transition between the state  $\mathcal{L}_m$  and  $\mathcal{L}_{m'}$ , then  $\tau_{m,qm'} = 1$ , otherwise  $\tau_{m,qm'} = 0$ . An action can be associated to each state: it may be a modification of a set-point or an on/off switching.

##### 4.3.4.2.1.2 Modeling the performance of services

Depending on the type of service, the quality of the service achievement may be assessed in different ways. End-user services provide comfort to inhabitants, intermediate services provide autonomy and support services provide power that can be assessed by its cost and its impact on the environment. In order to evaluate these qualities different types of criteria have been introduced. Here is presented the case of End-user services.

The global function of comfort is very complex to compute. This function not only depends on the satisfaction regarding each service (heating, cooking, washing...) taken on its own but also on psychological complex factors. Let's try to specify how this global satisfaction function is.

Let  $\sigma$  be the global function of comfort or the global function of satisfaction in a living space. Leaving implicit psychological factors, it can be stated:  $\sigma = \sigma(\sigma_1, \dots, \sigma_n)$  where  $\sigma_i$  represents the satisfaction related to a service  $SRV_i$ .

The satisfaction functions  $\sigma$  or  $\sigma_i$  takes values in the interval  $[0, 1]$ .

Let's now consider indicators to assess the performance of some services. In the following, indicators have to be considered as proposals but alternative indicators coming from further researches could also be used provided that they can be reformulated with a MILP formalism.

Generally speaking, modifiable permanent services use to control a physical variable: the user satisfaction depends on the difference between an expected value and an actual one. Let's

consider for example the temperature of a room heating service. A building can usually be split into several heating services related to rooms (or thermal zones) assumed to be independent.

Let's consider the comfort standard 7730 for thermal comfort assessment. According to this standard, three qualitative categories of thermal comfort can be distinguished: A, B and C. In each category, typical value ranges for temperature, air speed and humidity of a thermal zone that depends on the type of environment proposed : office, room,...These categories are based on an aggregated criterion named Predictive Mean Vote (PMV) that models the deviation from a neutral ambience.

The absolute value of this PMV is an interesting index to evaluate the quality of a HVAC service related to a thermal zone because it can be easily transformed into a MILP formalization. In order to simplify the evaluation of the PMV, typical values for humidity and air speed are used. Therefore, only the ambient temperature corresponding to the neutral value of PMV (PMV=0) is dynamically concerned. Under this assumption, an ideal temperature  $T_{opt}$  is obtained. Depending on the environment, an acceptable temperature range coming from the standard leads to an interval  $[T_{min}, T_{max}]$ . For instance, in an individual office in category A, with typical air speed and humidity conditions, the neutral temperature is  $T_{opt} = 22^\circ$  and the acceptable range is  $[21^\circ, 23^\circ]$ .

Therefore, considering the HVAC service  $SRV(i)$ , the discomfort criterion  $D(i, k)$ , is modeled by the following formula where assumptions are depicted by two parameters  $a_1$  and  $a_2$ :

$$D(i, k) = [PMV(T_{int}(i, k))] = \begin{cases} a_1 \times \frac{(T_{opt} - T_{in}(i, k))}{T_{opt} - T_{min}} & \text{if } T_{in}(i, k) \leq T_{opt} \\ a_2 \times \frac{(T_{in}(i, k) - T_{opt})}{T_{max} - T_{opt}} & \text{if } T_{in}(i, k) > T_{opt} \end{cases} \quad (6)$$

Generally speaking, modifiable temporary end-user services do not aim at controlling a physical variable. Temporary services such as washing are expected by inhabitants to finish at a given time. Therefore, the quality of achievement of a temporary service depends on the amount of time it is shifted. Therefore, in the same way as for permanent services, a user dissatisfaction criterion for a service  $SRV(i)$  is defined by:

$$D(i) = \begin{cases} \frac{f(i) - f_{opt}(i)}{f_{max}(i) - f_{opt}(i)} & \text{if } f(i) > f_{opt}(i) \\ \frac{f_{opt}(i) - f(i)}{f_{opt}(i) - f_{min}(i)} & \text{if } f(i) < f_{opt}(i) \end{cases} \quad (7)$$

where  $f_{opt}$  stands for the requested end time and  $f_{min}$  and  $f_{max}$  stand respectively for the minimum and maximum acceptable end time.

#### 4.3.4.2.2 Formulation of the anticipative problem as a linear problem

Formulation of the energy management problem contains both behavioral models with discrete and continuous variables, differential equation and finite state models, and quality models with nonlinearities such as in the PMV model. In order to get mixed linear problems which can be solved by well-known efficient algorithms, transformations have to be done.

The following presents the time shifting formalization.

Temporary services are modeled by finite state machines. The consumption of a state can be shifted such as task in scheduling problems. The starting and ending times of services can be synchronized to an anticipative period. It leads to a discrete-time formulation of the problem. However, this approach is both a restriction of the solution space and an approximation because the length of a time service has to be a multiple of  $\Delta$ . The general case has been considered here.

Figure 3 presents an example of a state in temporary services. Temporary services can be continuously shifted. Let  $DUR(i, j)$ ,  $f(i, j)$  and  $p(i, j)$  be respectively the duration of the state  $j$

of service  $SRV(i)$ , the ending time and the power related to the service  $SRV(i)$  during the state  $j$ .  $f(i, j)$  is defined according to inhabitant comfort models.

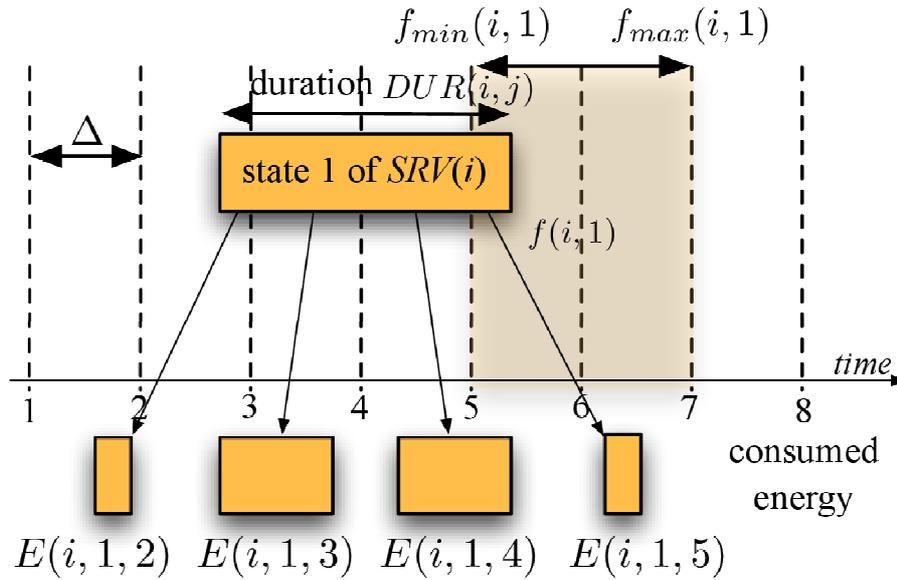


Figure 28: Shift of temporary services

The potential consumption/production duration  $d(i, j, k)$  of a service  $SRV(i)$  in state  $j$  during a sampling period  $[k\Delta, (k + 1)\Delta]$  is given by (see figure 3):

$$d(i, j, k) = \min(f(i, j), (k + 1)\Delta) - \max(f(i, j) - DUR(i, j), k\Delta) \quad (8)$$

Therefore, the consumption/production energy  $E(i, j, k)$  of the service  $SRV(i)$  in state  $j$  during a sampling period  $[k\Delta, (k + 1)\Delta]$  is given by:

$$E(i, j, k) = \begin{cases} d(i, j, k)p(i, j) & \text{if } d(i, j, k) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where  $d(i, j, k)$  stands for the duration of the state  $j$  of the service  $i$  during the period  $k$ . It is null if the state  $j$  of service  $i$  does not intersect the anticipative period  $k$ .  $p(i, j)$  is the power consumed during the state  $j$  of the service  $i$ .

Some services have been modeled by mixed integer linear form. Other services can be modeled in the same way. Anticipative control in building energy management can be formulated then as a multi-criteria mixed-linear programming problem represented by a set of constraints and optimization criteria.

#### 4.3.4.3 Principle of mixed solving approach

This regular/centralized solving approach has some limits:

- the appliances having a model non-shared by manufacturers: usually, manufacturers keep their appliances models. For the centralized solver, these models cannot be included in the problem solving. The solver can only take into account an unsupervised service reducing accuracy.
- the appliances that need some precision and cannot be included as linear model.
- the appliances having a non-linear model. The local problem solving can be done by using a non-linear optimization method such as Nelder Mead or SQP (Sequential Quadratic Programming). These categories gather appliances with non-linear model and appliances that can be managed by specific solvers.
- the appliances that are managed by user-defined specific heuristic rules. These appliances have some "behavioral rules" which provide the solution without the need of any

optimization. The solver must take into account the chosen solution in the global problem solving.

The following parts present the solution proposed to integrate these types of appliances in the global solving of the problem. As shown on Figure 4, the system consists of three main parts:

- The regular services consist of the services which have a linear model and can be integrated directly into the energy management problem.
- The agents consist of the services that do not have a linear model and should communicate with the solver to give their energetic profiles.
- The solver consists of a regular solver with the ability to communicate with agents. The solver integrates the information sent by the agent’s local solvers with regular service models in order to generate a global problem to solve.

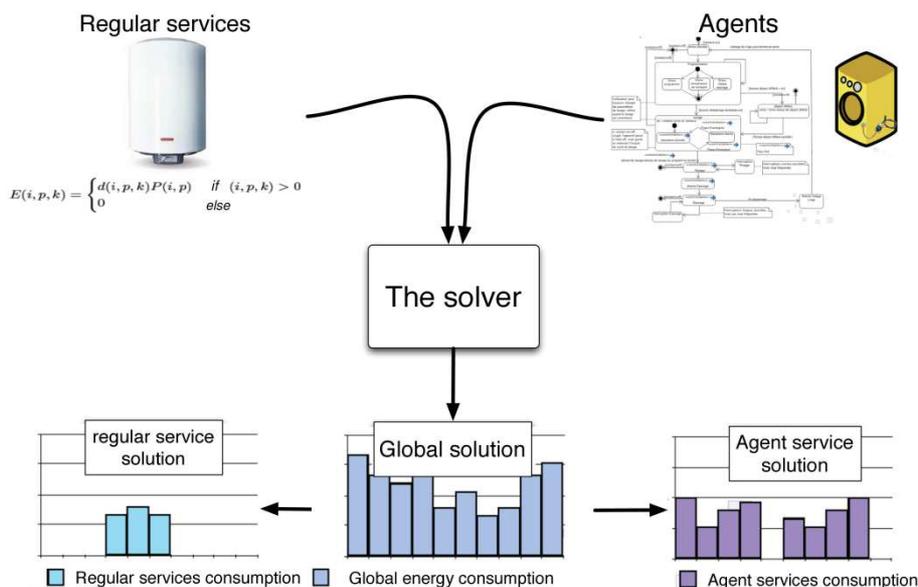


Figure 29: Global architecture of the mixed solving system

Only one communication is needed between this regular services and the solver. At the beginning of the solving process, the solver receives the linear model from the regular services. The models are used all along the solving process.

In the case of agents, some communications are needed. Each exchange between the agents and the solver is considered as a step in the solving process. In each step, an intermediate problem is created by the solver and then computed. The solver decides which information is needed to be sent to the agents in the next step. The agent takes into account the information sent by the solver and sends energetic profiles. The solving process is presented in the following in three parts:

- The progress of the problem solving during one solving step.
- The solver’s behavior during the solving process.
- The agent’s behavior during the solving process.

4.3.4.3.1 One step solving

Figure 5 presents the information exchanged between the solver and the regular services and the agent services during the first step in the solving process.

First, the solver receives the linear models of the regular service. This operation is the initialization of the problem. Once initialization is done, the solver computes the relevance indicator. It is an indicator with the purpose to direct the local solving problem in the agent. When this indicator is computed, it will be sent to all agents.

The agents don't have any information about the environment but they have the ability to solve their own local problem. When agents receive the relevance indicator, they compute their solutions taking into account this indicator serving as information about their environment. They obtain several solutions, which are called energetic profiles. It is the consumption for the concerned agent for each period of the optimization horizon. All these profiles are sent back to the solver. The solver includes them in the problem to be solved at this step. Then the global problem with all the services is solved at this step.

After the first step, the solver begins a new step by computing the relevance indicator. The relevance indicator is computed taking into account the received energetic profiles sent by the agents in order to improve the global solution each step in the solving process.

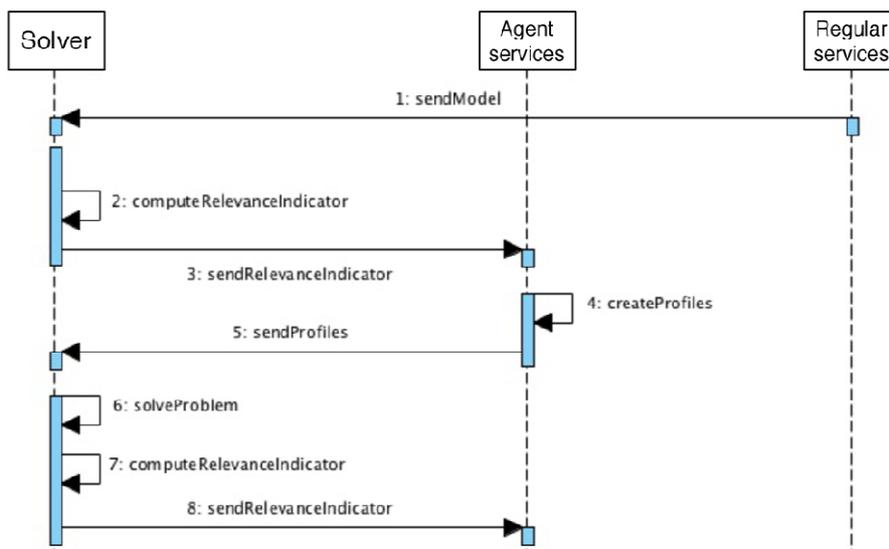


Figure 30: Solving process during one step

4.3.4.3.2 Solver's role

The solver has two tasks to do in each step. In order to formulate these tasks, we introduce some notations:

- $k$  is the index of anticipative period
- $S$  is the set of services
- $S^L$  is the set of *regular services*
- $S^D$  is the set of *agent services*
- $S$  is a service included in  $S$
- $E_k^{max}$  is the available energy during the period  $k$  before any optimisation
- $E_k(S)$  is the consumed energy by the regular service  $S \in S^L$  during the period  $k$
- $E_k(S, i, \mathbb{P}_k)$  is the consumed energy by the agent service  $S \in S^D$  during the period  $k$  for the  $i^e$  profile
- $C_k$  is the cost of energy during the period  $k$
- $v(S)$  is the characteristic of inhabitant request for the service  $S$
- $D(v(S))$  is the dissatisfaction of the *regular service*  $S \in S^L$
- $D(v(S), i, \mathbb{P}_k)$  is the dissatisfaction of the *agent service*  $S \in S^D$  for the  $i^e$  profile
- $\mathbb{P}_{k, \forall k}$  is the relevance indicator for the current step of resolution

4.3.4.3.2.1 Optimization problem

At each step, the solver computes a linear problem to find a solution. This problem is extended by including agent services and so, some equations are added. A new set of variable for each agent service is introduced (see equation 24).  $\zeta_i(S)$  is a binary variable which value is 1 if the profile  $i$  of the agent service  $S$  is chosen by the solver, 0 otherwise.

$$\zeta_i(S) \in \{0, 1\}, \forall i \tag{24}$$

$$\sum_i \zeta_i(S) = 1 \tag{25}$$

The criterion to minimize is modified and becomes a two parts criterion (26).

$$J_{iter} = \sum_{S \in S^L} \left( \sum_k C_k E_k(S, \theta(S)) + \lambda \times D(v(S, \theta(S))) \right) + \sum_{S \in S^D} \sum_I \zeta_i \left( \sum_k C_k E_k(S, i, \mathbb{P}_k) + \lambda \times D(v(S), i, \mathbb{P}_k) \right) \tag{26}$$

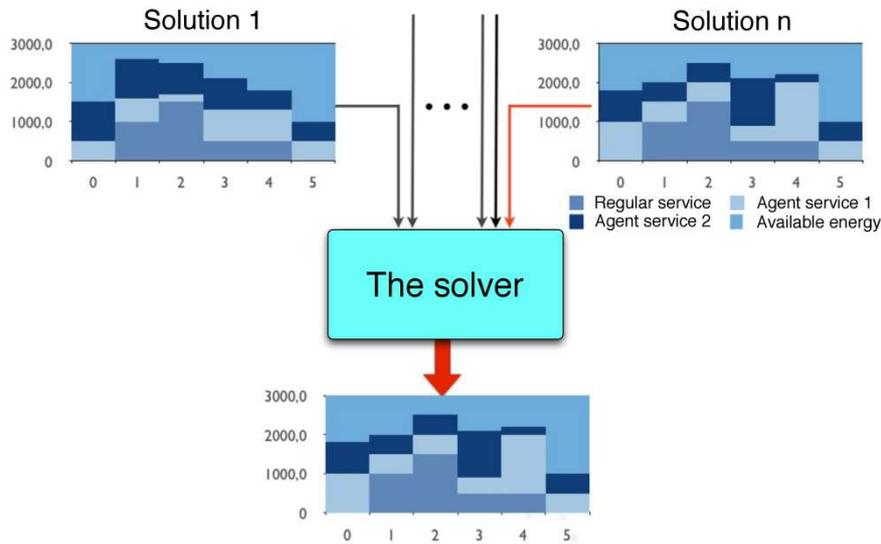


Figure 31: Solution found by the solver

There are two different parts in this criterion, one part concerning regular services and one part for agent services. They are designed on the same scheme to have a standardized criterion. This scheme split into two influences:

- The influence on the cost: the global energy cost must be minimized.
- The influence on the inhabitants: the dissatisfaction of the inhabitants must be minimized.

Those influences can be found in both regular services part and agent services part. Nevertheless, there is a fundamental difference between these two parts which is symbolized by the sum on the index  $i$  in the agent services part. The solver keeps only one profile for each service agents, and for each profile, the solver receives one consumption plan and an associated dissatisfaction. The sum in the criterion with binary variables forces to keep only one profile per agent for the minimization.

Figure 6 shows the complexity of the problem to be solved at each step. Each service agent provides  $n$  profiles and if there are  $m$  singular services, there are  $n^m$  different solutions. But the solver has to minimize the criterion to keep one.

4.3.4.3.2.2 Relevance indicator

The relevance indicator is computed during each solving step to direct the local solving process of service agents for the next step. After the solving step  $j$ , the relevance indicator is

computed with the equation 27. The purpose of this approach is to share the information about the energy consumption and price between solver and service agents. The service agents integrate the received information in their local solving process of the step  $j + 1$ .

$$\mathbb{P}_k^j = \frac{1 + E_k^{max}}{1 + E_k^{max} - \sum_{S \in \mathcal{S}^L} E_i^{j*}(S)} C_k \tag{27}$$

4.3.4.3.3 Role of the agents

An agent is dedicated to a specific entity whose behavioral model cannot be linearized and then taken into account directly by the solver. In this part, the algorithm used by agents is explained using an example of washing machine service agent.

The washing machine service agent has its internal state model. The states are shown by figure 7. They consist in:

- some behavioral states like heating, prewash, washing and spin-drying.
- two states representing the beginning and the end of the service
- some states denoted wait  $i$  represent the waiting time between behavioral states
- some states modeling the interruption within each state, denoted interrupted state

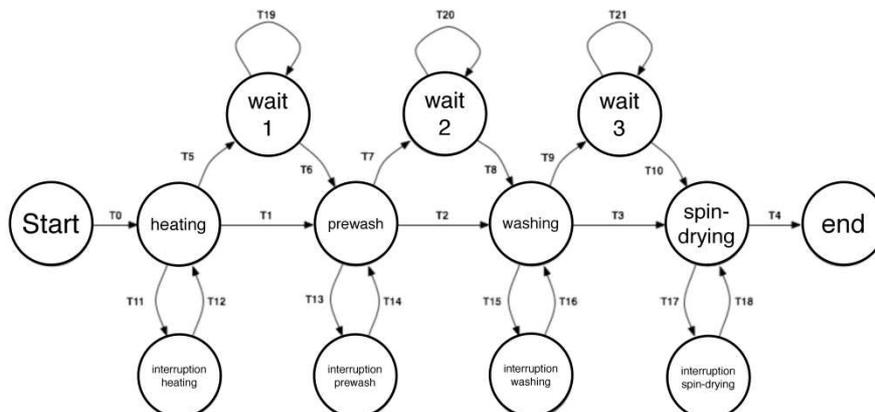


Figure 32: State model of the washing machine service agent

The normal behavior of the washing machine service is given by the state sequence scenario [start, heating, prewash, washing, spy-drying, end]. The other states are only visited when the service agent tries to find some neighboring profiles in order to respond to some criteria sent by the solver.

Each visit to an interrupted state has a fixed time period. It is possible to visit the interrupted state more than once in order to increase the interruption time in a state.

A behavioral profile is the state sequence scenario with the date of each state visit. The behavioral profile is characterized by:

- the starting time of the service
- the number of visits to each interrupted state and the number of visits for each *wait i* state
- the date of each visit to interrupted states and *wait i* states.

These characteristics are denoted in the following parameters of behavioral profile. The energetic profile consists on the energy consumed by the service in each period of the anticipative horizon. The energetic profile is then sent to the solver.

The Agent satisfaction is computed according to the energetic profile. The satisfaction depends on the number of visited interrupted states and on the effective ending time.

The agent solving algorithm is presented in figure 8.

Firstly, the agent receives the relevance indicator which consists of information about the penalization, the energy price during the anticipative horizon and the chosen energetically profile at step  $j$ .

Then the values of relevance indicators have to be normalized (28) in order to obtain  $CA_k$ , the agent coefficient.

$$RI_{k(normalized)} = RI_k / Max(RI_k) \tag{28}$$

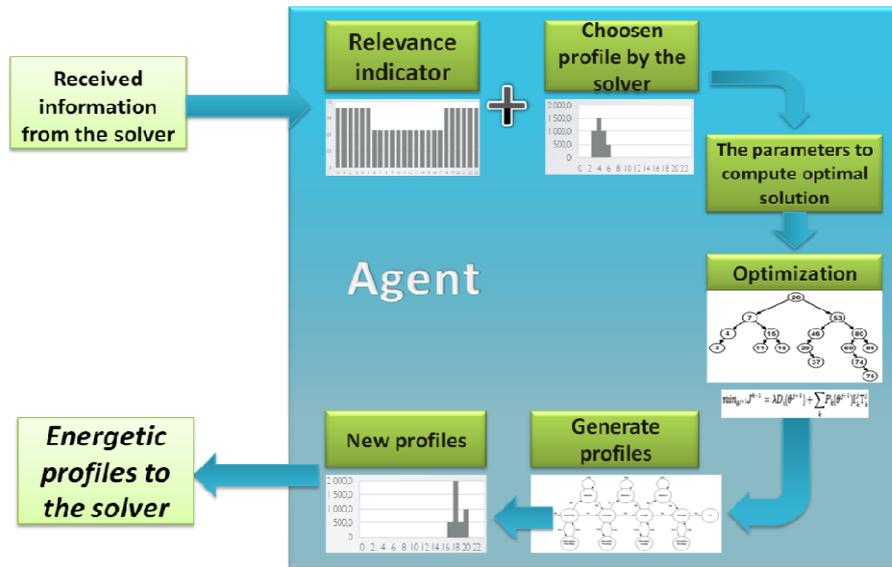


Figure 33: Solving algorithm in the agent

From the  $CA_k$ , it merges the information about the penalization, the energy price and the agent dissatisfaction denoted  $I_k$ (29).

$$CA_k = RI_k + \lambda \times I_k \tag{29}$$

In order to generate an energetic profile, the first step is to compute the behavioral profile. The parameters of the behavioral profile are listed above. The first one is the starting time of the service. The best intervals over 6 periods in the 24 hour horizon according to the values  $CA_k$  are computed; for each interval  $j$ ,  $X_j$  is:

$$X_j = ( \sum_{k \in [j, j+6]} CA_k ) / 6 \tag{30}$$

$X_{j_{min}}$  is the minimum of the list  $X_j$ , and the intervals that have no significant difference with  $X_{j_{min}}$  are evaluated.  $L_{min}$  is the list:

$$L_{min} = \left\{ k / 1 - ( X_{j_{min}} / X_k ) < 1 \right\} \tag{31}$$

The interval  $\chi$  with the maximum variance in  $L_{min}$  is chosen for the optimization.

The parameters of the optimization are presented in figure 9, where  $N_{Si}$  is the number of interruptions in the state  $Si$ .  $W_{Si}$  is a value to select the time for interruption within the state.

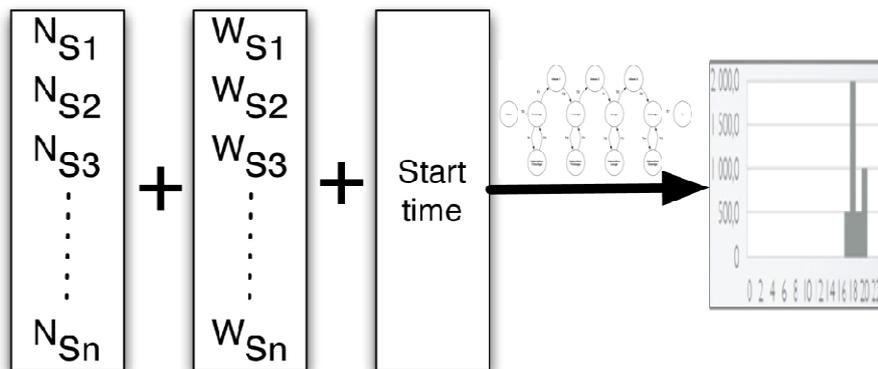


Figure 34: Parameters of a profile

A branch and bound optimization is achieved on this parameter (Figure 10) within the chosen interval  $\chi$ . Each agent solves the optimization problem with this function. It represents the minimization of the energetic cost and dissatisfaction from a local point of view.

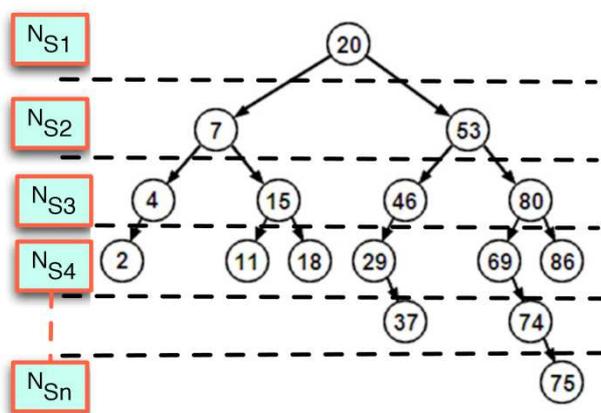


Figure 35: Optimization using branch and bound

The function (32) to be minimized is similar to the one presented for the solver.

$$\min_{\theta^{j+1}} J^{k+1} = \sum (E_k(\theta^{j+1}) \mathbb{P}_k^j \mathbb{T}_k^j + \lambda D_i(\theta^{j+1})) \tag{32}$$

$\theta^{j+1}$  represents the parameters of the user that define the usage conditions. The function is composed of two parts: the first one is the influence of the energetic cost and the second one is the influence of the satisfaction of the agent.

The results of this optimization are a list of parameters required to generate the behavioral profile (parameters of behavioral profile). Then, the energetic profile can be computed and sent to the solver to be integrated in the global problem solving.

#### 4.3.4.3.4 Implementation

The implemented system consists of four components (figure 9):

- the classical regular solver
- the solver that solves global problem composed from regular problem and agent problems
- the broker agent is a communication component that receives all the local problems from service agents and constructs one global service agent problem.
- the service agent with the capabilities to solve a local problem.

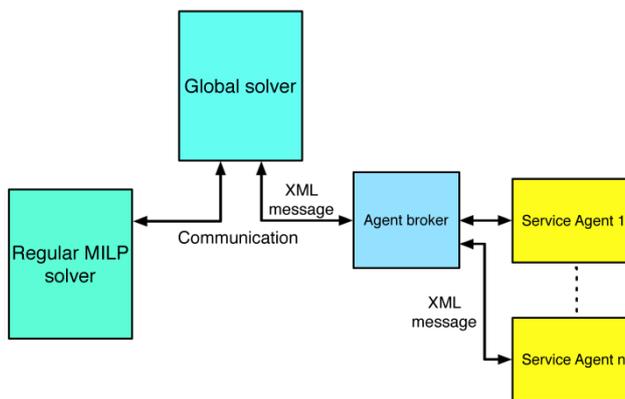


Figure 36: The component of the mixed solving system

### 4.4 Shared services layer

The shared services layer provides common-denominator horizontal services which are separated from application-specific logic and may as such be used by a number of vertical ("vertical" understood here to mean specialized rather than vertically-integrated) applications. The shared services layer serves the following purposes:

- it delineates the environment in which application-layer code operates; it defines the boundaries and APIs of a framework or container architecture for the application layer objects
- it provides ready-made solutions for a number of elementary, clearly-defined problems that applications commonly face in a distributed setting

As such, the shared services layer is a kind of permanent, or at any rate slowly evolving, infrastructure on top of which application-layer code may be built without having to re-implement commonly recurring functionality from scratch. The existence of such a layer therefore brings about all the technical and economic benefits typically associated with:

- *specialization / division of labor*: since the services implementations will likely be maintained by one or more dedicated organizations / providers. The service interfaces themselves will be general purpose but commercial, academic or not-for-profit organizations will likely emerge specializing in implementing a suite of shared services for the new environment.
- *capital formation & GPT effect*: the services created in the shared services layer will outlive the life cycle of any given application and thus represent a permanent contribution to the ecosystem. They provide the accrued benefits of a
- *decomposition*: easier testing and independent evolution of applications disentangled from the services they use.

Shared services layers can typically have two incarnations:

- (a) framework APIs
- (b) web services

In case (a) the shared services are part of the system itself and the code usually executes locally and even in the same process (memory space) as the application that uses it. This is a tight integration model but is also better performing, more efficient and, when interacting with low-level resources such as disks or IO ports, indispensable. In case (b) the shared services are provided at remote locations, usually because the nature of the service itself allows or enforces that. A typical example of a shared service of type (a) would be a logging facility whereas of type (b) would be a naming service. Notice that in either case there is no functional reason why logging may not be provided remotely or why the naming service could not run locally. It is the

considerations of non-functional constraints and pragmatic requirements that dictate one or the other deployment. Put differently, the services of type (a) are more oriented towards fulfilling purpose I described at the beginning of this section whereas services of type (b) cater to purpose II. Finally, it should be clarified that a web-service-based service can also provide client-side libraries for the benefit of the application developers (although not strictly necessary and even frowned-upon as a practice by some practitioners<sup>1</sup>). Such libraries, when provided should not be confused with framework APIs which expose locally available implementations.

When discussing shared services the issue of the implementation platform (e.g. programming language and framework / managed environments) inevitably becomes relevant. This is because, depending on the framework chosen, certain services are readily available either as part of the distribution itself or as part of established third party libraries that are typically used by practitioners. Examples are logging, serialization / deserialization, networking, etc. However, it doesn't make sense for us to explore or discuss such domain-agnostic services. We assume that an adequate layer of such services is available and we will only discuss domain-relevant (even though not strictly domain-specific) services which are not provided at the platform layer or for which a more specialized implementation might make sense.

With the above qualification in mind we will discuss, for the remainder of this sub-section a number of appealing shared services that could populate this layer. When appropriate we also describe likely deployment (framework API or external web service).

#### 4.4.1 Virtual entity services

Most building energy management use cases and, more generally, most smart building applications are defined in terms of higher-order abstractions that do not directly correspond to the individual entities represented in the building abstraction layer. I.e. a use case for dimming the lights in the living room uses the abstraction of a lighting service in a room which can involve a number of fixtures and other non-power-related automations that still need to be engaged (e.g. shades). Also a use-case for changing the heating schedule of the hot water tank when it is noticed that there has been no human activity for the last two days calls for fusing sensor input from a variety of sensors and appliances all over the apartment. Although it is of course technically feasible for each application to re-define the relevant composite abstractions in its own logic it appears that there is some value in providing a horizontal facility that allows a shared service to define certain aggregations of entities, and below these, of sensors and actuators in a spatially-aware way and to expose higher order primitives both for obtaining aggregated data from a set of entities and for translating a single higher order function (e.g. "dim lights") into a set of independent directives to individual entities and the corresponding actuators.

In the proposed architecture, controllers are synthesized from generic rules, which as such apply to generic functions (lighting, heating), rather than to individual entities supporting these functions. Entity groups are used as an intermediary abstraction of entities to support the adaptation of generic rules to specific entities.

Also, in a smart environment, the control goals care more about the entire environment as an integrated unit instead of paying attention to individual pieces of entity. Therefore, the controllers concentrate more on the entire environment's properties, such as the global temperature, the luminosity of the space, etc. to which some of the entities existing in the environment make contribution. It is logical to classify the entities by the properties which would have a common effect on the entire environment, e.g. dissipating heat, emitting light, and

---

<sup>1</sup> The rationale being that one of the benefits of using web services is openness and therefore providing vendor-specific client-side libraries, even for the benefit of the client-side application developers, runs contrary to that objective and may introduce inconsistencies or subtle deviations from the behavior foreseen by the HTTP protocol which will not be diagnosed unless somebody tries to consume the web service without employing the vendor libraries.

opening. This classification method is more interesting to the upper level controllers because in most of the cases, such a property corresponds to a general control goal. The group of entities shadows the individual and specific sub models that the upper level controllers have no more than these groups to deal with. The control order applied on one group of entities is transmitted automatically to all its descendants, which implies that children should inherit all properties from its parent or parents.

In the example showed in Figure 4, we put different space entities into 2 categories: “being occupied” makes sense for all ones belonging to “occupiable space” category while all ones belonging to “lock-able space” can be isolated locked from the outside. When a controller applies its rules which depend only on if the space is occupied or not, e.g. “turn off the light if the space is not occupied”, it does not need to know if the concerned space is a room or a street and will turn off the light for both the room and the street if neither is occupied.

Concerning the “thing” entities, the principle is the same. They are arranged into different categories by their intrinsic properties. Obviously, a child has all properties of its parent(s), e.g. “lamp” has “light”, “heat dissipating” and “electrical” as its parents so that it has three properties. If a parent has several properties such as “lamp”, its children will automatically have all of them. If the controller has a rule such as “close all the open things when there is nobody” which applies on the group of entities “opening”, it does not need to worry about which concrete entities it controls because the order will do down layer by layer until it reaches the bottom. We should notice that an entity can be directly under a group, such as “motor” under the group “electrical”, while it is a sub-entity under an entity attributed already to another group, such as “motorized blind” is under “blind” which belongs to the group “opening”.

If there is any conflict between rules on the same entity but from different groups as the target entity belongs to several groups, the principle is the same as in the case where the entity is subjected to several contradictory rules as explained in the next section “controller priority”. Indeed, unlike the relationship between generic and specific entities, the groups are not necessarily considered as intermediate entities. Their existence is mostly for control convenience and described as control rules rather than entities to be controlled. The control rules are actually exercised on the entities directly under the group not on the group itself.

#### 4.4.2 Building services in the ReActivHome project

Components like rooms and appliances are elements of the structural representation of buildings but a functional representation is more relevant for global energy management because it highlights the role of each element. A service oriented representation has been used in the ReActivHome project. Each service is defined by:

- a set of supporting components and appliances of the building
- a time period  $T_i$  where the service may occur: for instance, the time period where a washing machine may consume power to do a specific washing
- a set of actions  $U_i(T_i)$  that may modify the achievement of the service: it may be set-points or controlled variables
- a set of available observations  $Y_i(T_i)$  that provide information about an actual behavior: it has to include consumed, stored or provided powers because energy is focused.
- a set of modeling constraints  $K_{i,j}(U_i(T_i), Y_j(T_j)) = 0, \forall j$  that depict the links between actions and observations: these constraints depend on the supporting components and appliances
- a set of operational constraints  $K'_{i,k}(U_i(T_i), Y_k(T_k)) \llcorner 0, \forall k^2$  that depict the operational limits of the building: these constraints depend on the supporting components and appliances

---

<sup>2</sup>  $\llcorner$  stands for a comparison operator

- a service performance indicator  $S_i(U_i(T_i), Y_i(T_i))$ : it may be an occupant comfort level indicator, a quantity of stored energy or a cost (energy or environmental cost for instance)

Buildings with appliances aim at providing comfort to inhabitants as a final aim. Services can then be decomposed into three kinds: the end-user services that provide directly comfort to inhabitants, the intermediate services that manage energy storage and the support services that produce electrical power to intermediate and to end-user services.

Let's assume a given time range for anticipating the energy needs (typically 24 hours). A service is qualified as permanent if its energetic consumption/production/storage covers the whole time range of energy assignment plan, otherwise, the service is named temporary service. The table 1 gives some examples of services according to this classification, and the figure 1 details their relationships

	temporary services	permanent services
support services	photovoltaic panels	power provider
intermediate services	-	storage
end-user services	washing	room heating

Table 1: Example of temporary and permanent services

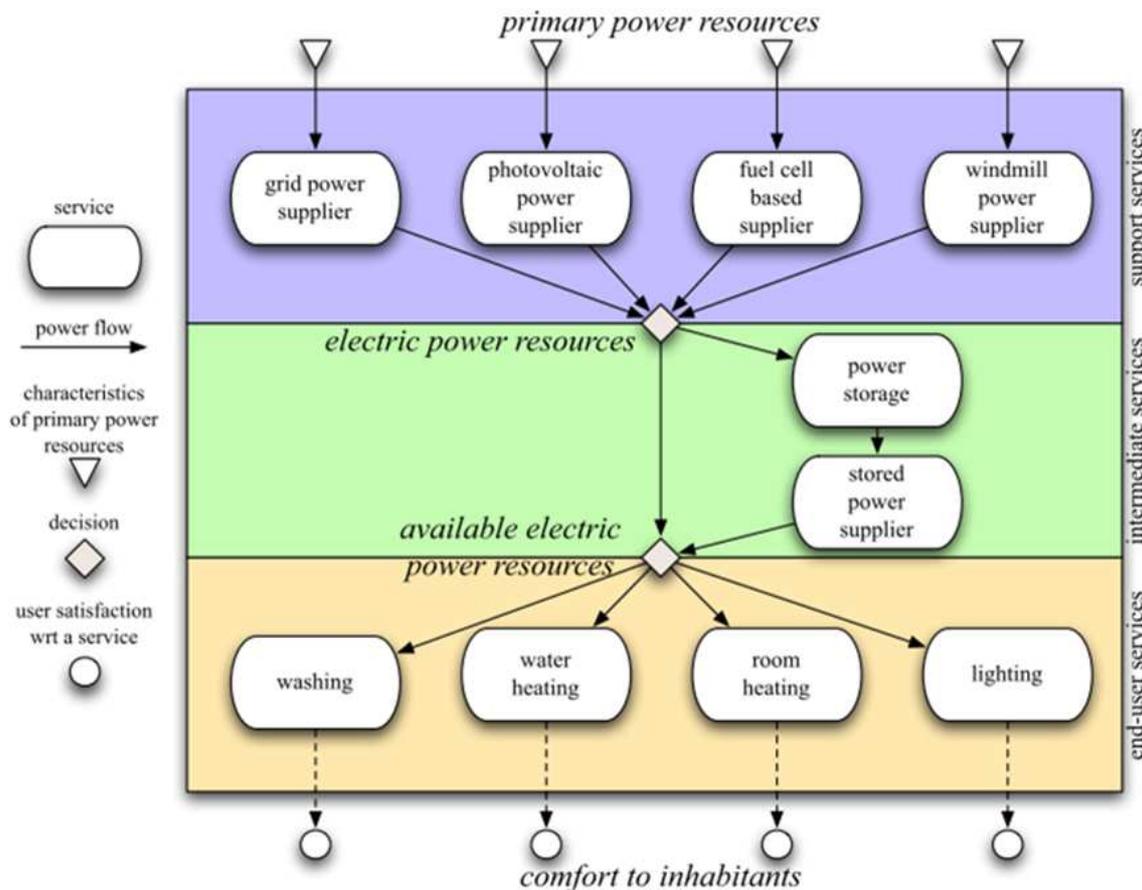


Figure 37: Structure of services

The services can also be classified according to the way their behavior can be modified.

Whatever the service is, an end-user, an intermediate or a support service, it can be modifiable or not. A service is qualified as modifiable by an energy management system if the energy management system is capable to modify its behavior (the starting time for example).

There are different ways of modifying services. Sometimes, modifiable services can be considered as continuously modifiable such as the temperature set points in room heating services or the shift of a washing. Some other services may be modified discretely such as the interruption of a washing service. These ways of modifying services can be combined; a washing service can be considered both as interruptible and as continuously displaceable. A service modeled as discretely modifiable contains discrete decision variables in its model whereas a continuously modifiable service contains continuous decision variables. Furthermore, a service may contain both discrete and continuous decision variables.

A service can also be characterized by the way it is known by an automation system.

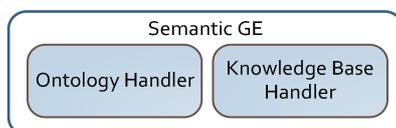
Obviously, a service can be taken into account by an energy management system if it is observable. Some services are indirectly observable. Indeed, all not observable services can be gathered into a virtual non modifiable service whose consumption/production is deduced from a global power meter measurement and from the observable service consumptions and productions. In addition, a service can be taken into account for long term scheduling if it is predictable. In the same way as for observable services, all the unpredictable services can be gathered into a global non modifiable predictable service. A service can be managed by an automation system if it is observable and modifiable. Moreover, it can be long-term managed if it is predictable and modifiable.

**4.4.3 Building state maintainer**

This shared service provides a functional block that collects data from the drivers to provide a representation of the actual state of the building system, and also fuses information from the discovery service (described in 4.5) to provide the current state and structure of the building system. For instance, using the building state maintainer shared service, application layer logic can answer questions of the following types:

- Which parts of the building actually have human activity right now?
- How crowded are certain parts of the building, throughout the day?
- What is the current state of the lifts / elevators / escalators?
- What is the temperature of the water in the hot water tank(s)?
- Has the security system been activated?
- How much time has elapsed since the last observed human activity in a certain part of the building?
- When will the pool, most likely, be used again?

In a FI-WARE based Smart Building environment device instances should be able to publish semantic content. The FI-WARE IoT Chapter specifies in the “IoT Process Automation” section Semantic GE that can provide container of device semantic information.



This information should semantically formulate a comprehensive device content that includes capability, consumption profile and other data or semantic properties.

The Topology Discovery Service can then collect information per device and generate a building ontology. The user can enhance the content by adding annotations relating the device with Real-Life Actions semantics.

A semantically enhanced object mapping function then can correlate semantic annotations to real time meter and sensor data, in order to provide building mapped status and measurement indications, in a structured and meaningful way (e.g. Semantic Hierarchy based faceted display)



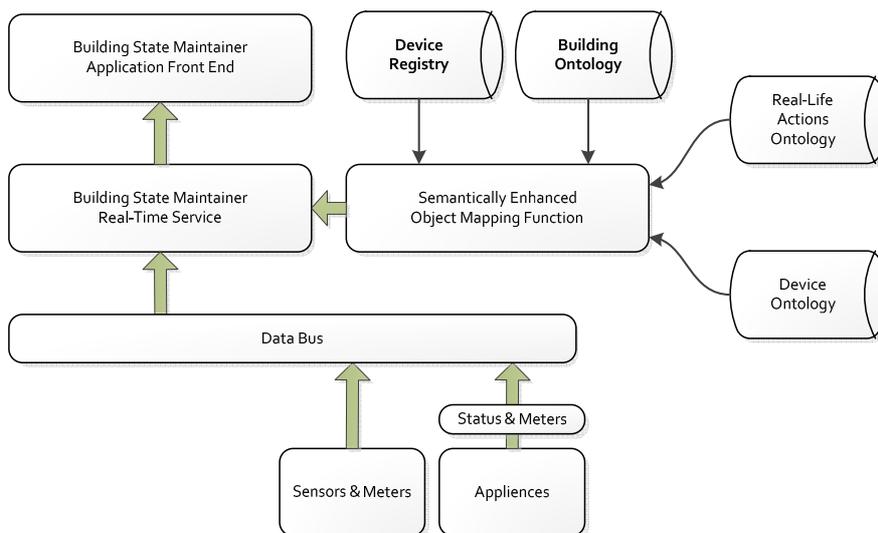
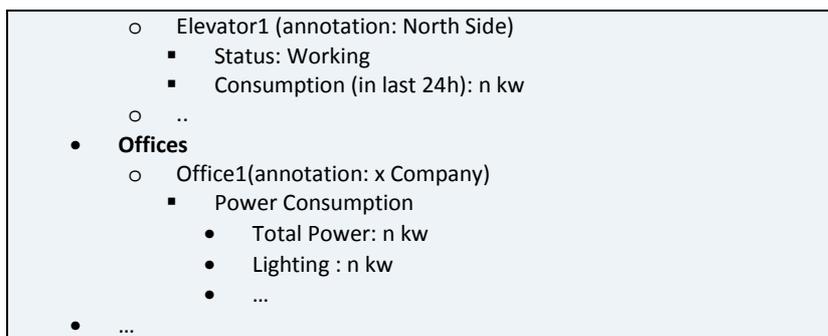


Figure 38: Building State Maintainer service.

#### 4.4.4 Historization

The historization shared service archives building relevant energy data in a database and automatically maintains and phases out content as configured. Using the historization service the application-layer logic can answer such questions as:

- When are showers typically taken in weekdays?
- What is the typical occupancy of a certain part of a building on Friday nights or on specific time periods?
- How often a specific part of the building is being used and for how much time?
- Are there any locations in the building that are more crowded than others?

Such information can be relevant for energy optimization purposes. The historization service allows the building energy optimization logic to "learn" the patterns of the building use and the daily routines of its inhabitants and adapt / optimize accordingly.

In this scenario the service will build on the same principles presented in the previous paragraph (Building State Maintainer service). The building ontology and Real-Life actions semantic annotations can allow the automatic generation of Optimization Profile templates that a user can customize through an application front-end.

The application front-end will produce a semantically annotated optimization profile that will be then uploaded in the Historization service.

The service then will use the optimization profile, along with other semantic information, in order to formulate queries and data mining logic that will be executed periodically from a service scheduler.

The results will produce consumption profiles that can be used to suggest optimal configurations for home/building automation tasks.

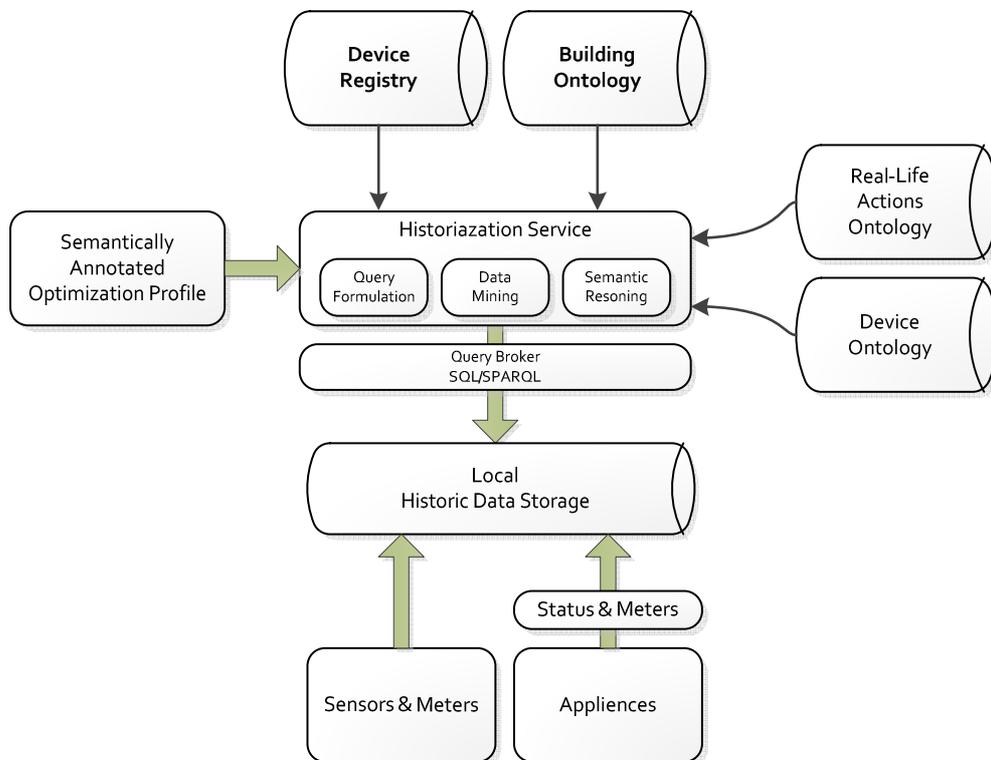


Figure 39: Historization service Data Mining mode

**4.4.5 Supervisory control services**

Supervisory control refers to a lightweight joint control of several individual controlled entities. Its main role is to coordinate entities in order to reach or avoid a prescribed joint state of the system comprising these entities. It is designed to be able to adapt itself to changing context and make decisions based on specific inputs, context events and the current states of each controlled entity.

In the building or even larger scales of the smart grid, all kinds of electrical equipment and energy-relevant physical entities can get integrated in a local energy management system. These entities may thus be considered to directly or indirectly connected to a perimeter of the grid and they are so numerous and heterogeneous that a complexity explosion would easily come up. The Building Abstraction layer described in the following section enables auto-integration of all kinds of entities into the local management system and their self-configuration after the integration.

There may exist a confusion between supervisory control in the sense put forward here and SCADA systems, which have been existing for decades and play a very important role in industrial automation. They monitor a system's behaviour by data coming from distributed sensors and control it by transmitting orders to distributed actuators, possibly with an operator's intervention. As the control interface integrated with sensors and actuators, it is custom-designed for these and cannot benefit from the auto-integration and auto-configuration functionalities provided by the entity abstraction layer which intermediates the sensor and actuator device layer.

WP4 proposes thus here a new supervisory controller considered as a component in the service layer just above the abstraction layer, where it controls only the abstract entities identified which shadow the sensor and actuator devices. Additionally, it will act as an intermediate between entities and applications by providing common functions to applications in the upper layer.

Just like other normal services, there can be several supervisory controllers for different purposes, e.g. building security, comfort, energy consumption optimization. Each of them can work independently or cooperate together. Therefore, a modular concept will be interesting in supervisory controller design in order to reduce the complexity and improve maintainability.

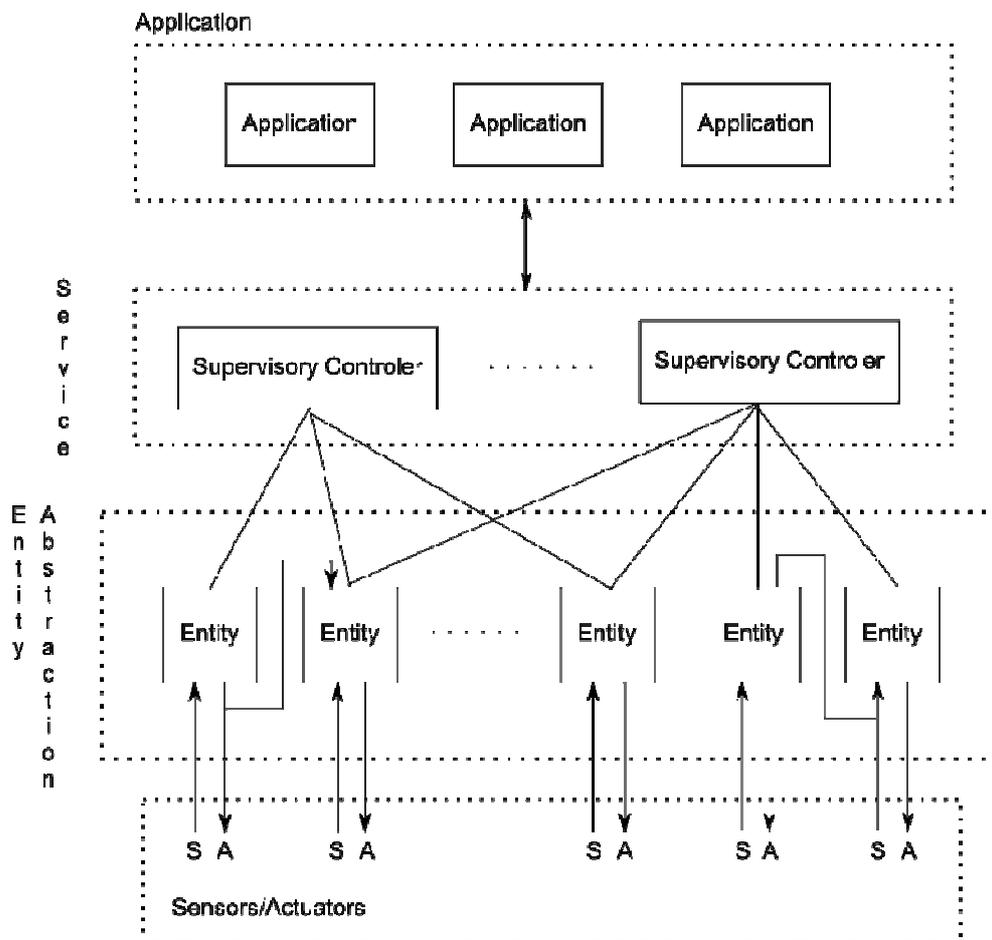


Figure 40: Supervisory control in service layer

Supervisory control in the service layer will not replace the more comprehensive and complex dedicated application-specific control being affected by applications in the upper application layer. They are for different purposes and designed by different approaches. Supervisory control is designed in bottom-up approach, which means it starts from the entity models in the lower level and ends in providing lightweight control functions based on exclusion or sequence criteria, such as general safety and energy-saving constraints.

**4.4.6 User interface services**

This service provides the ability for smart building applications and services to interface with one or several human users, be they end users of the building (typically in residential or office buildings), or members of a technical staff. The actual implementation of this service is obviously conditioned by the requirements of a particular application, but common support may be provided nonetheless, through e.g.; access to a number of non-dedicated interface devices like dashboards that may exist in a room. The most unobtrusive solution could be a screen-less

residential gateway that hosts a web server with a login page to allow registered users to view energy-related statistics on the building, its different appliances, produce reports (e.g. PDF files) and / or configure the system. Another useful modality that could be supported would be allowing the BEMS to send alerts to the user in the form of emails, instant messaging messages or GSM SMS.

### 4.5 Building Entity Abstraction Layer

The “Building Abstraction Layer” (BAL) we propose is an analogue of Hardware Abstraction Layers used by operating systems: it hides the specifics of the building hardware beneath a set of generic models and interfaces, acting as a generic informational interface to the building as a physical system. The BAL may comprise a collection of modules that act as full-featured proxies or, more accurately, the equivalent of the drivers of an operating system, for each individual physical entity/subsystem of the building, using associated sensors and actuators to interact with the physical entity. These sensors can (and, in principle, they should) be shared among the entities; for example, an infrared camera and an ambient microphone could be used to monitor all appliances of a room and the room itself.

The problem addressed by this Building Abstraction Layer can be stated in the most general possible way as follows (Figure 41): the BAL acquires data the building and controls it in return through shared sensors and actuators distributed as monitoring and control points through the building.

A set of entities, subsystems of the whole building, are defined as the components that are relevant for controlled and monitored by the targeted application. These subsystems are distinct physical entities which are fully-fledged physical systems in their own right. The overall building is thus made up of the composition of these physical subsystems for what is relevant to the application at hand. The sensors and actuators are not target entities themselves they are used just as transparent intermediaries.

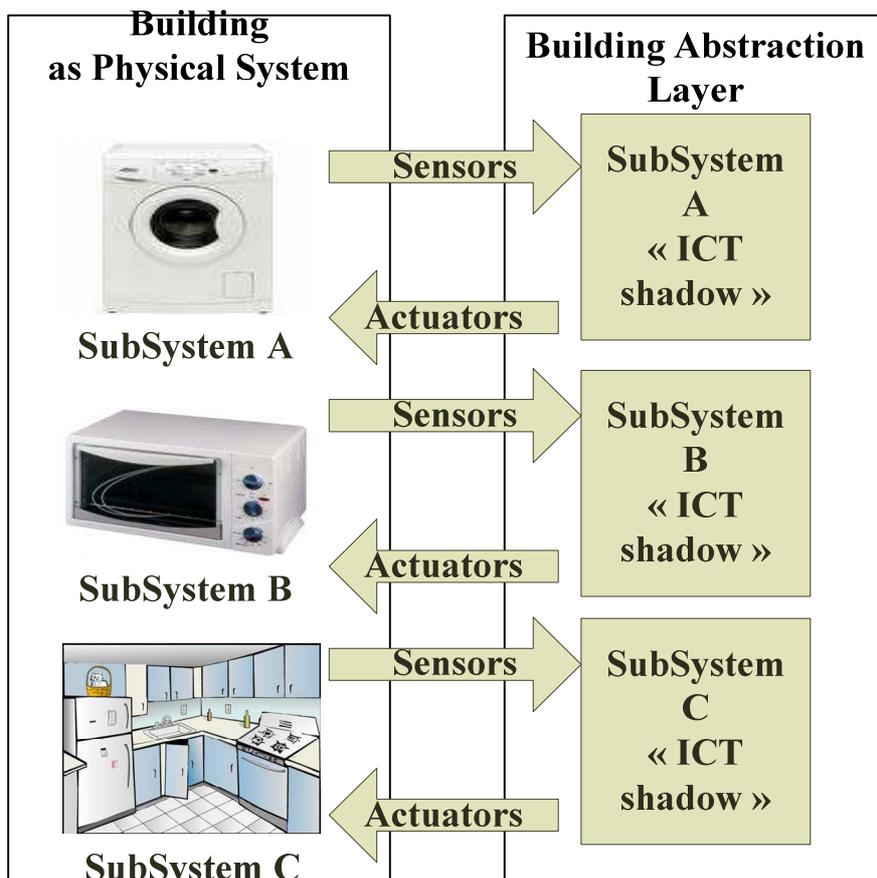


Figure 41: Conceptual view of physical entity “shadowing” in the building abstraction layer

The Building Abstraction Layer (BAL) will “shadow” or mirror each of these physical subsystems individually through matching self-contained software components that will offer their interface to services and applications. The BAL should have the capability to create and configure these components automatically, both for the initial configuration stage and when reconfiguration would be needed because of a change in the environment. This implies the need to associate automatically the subset of sensors and actuators that are used as intermediaries for the monitoring and control of a given subsystem, and to update the model of the subsystem accordingly.

#### 4.5.1 Models for target entities

The models that the Building Abstraction layer relies on are drawn from a generic ontology of building entities. This ontology and the corresponding models are described in the information models chapter of this document.

The discrete-event automata models used to represent target entities are a tradeoff between expressivity and ease of identification. The full description of a physical system such as the target appliance or room would normally require a continuous-state and continuous-time model, but the automatic identification of the parameters of such models would be impossible.

#### 4.5.2 Identifying the target subsystems

For the purpose of this document we define a target subsystem as a self-contained subset of the overall building that can and need be individually monitored and actuated, either indirectly through sensors and actuators or directly, if it is equipped with its own network connection. These two possibilities are not actually exclusive, and a subsystem with a network connection may still need to be monitored/controlled through complementary external sensors and actuators because the network connection does not provide access to the required data or functionality. Examples of such target physical subsystems are:

- Existing technical systems such as HVAC or security management systems
- Rooms, floors, or any relevant subsets of the building
- Appliances and devices including all types of pieces of home or office equipment, including components of more complex systems such as HVAC, as long as it makes sense to deal with them individually rather than to hide them within a larger subsystem
- Components of the building (including walls, roof, openings, etc.), as long as it makes sense to deal with them individually rather than to hide them within a larger subsystem.

These mostly non-digital subsystems have to be integrated in the BAL in a way similar to what is done with regular networked entities. This means they have to be identified and matched to an existing model that can be specific or generic, exact or approximate.

#### 4.5.3 Defining the relevant state of target subsystems

We consider the target subsystem as delineated before to be a full-fledged system in system-theoretic sense. Its state vector  $S(T)$ , a function of time  $T$ , can be defined in a system-theoretic sense as encapsulating the necessary and sufficient information to predict the future states and outputs of the system given its future inputs. This comprehensive definition has to be restricted for practical reasons to those dimensions of the state that are actually relevant for the target application. For an energy management application, the state of energy-relevant subsystems should comprehend their state as thermodynamic systems, which may comprise the amount of energy they are actually consuming, storing and generating. For a home automation application the mechanical state of these subsystems would be more relevant.

Being continuous as they normally are for a physical system, these states may for simplicity reasons be lumped into discrete “state categories” (e.g. a mode of operation for an appliance),

that could correspond to the states of a simpler discrete-state model. This finite state model can be made richer by complementing discrete state variables with continuous variables (attributes) capturing relevant properties related to the state (e.g., time spent in that mode, temperature, percentage filled, etc.). These digitized states together with the relevant attributes are then stored as the state of the shadow ICT subsystem.

Monitoring is concerned with the identification and tracking of these instantaneous states of the physical subsystem, on the basis of relevant sensor data.

Controlling refers to change the state of the physical subsystem by using actuators.

#### **4.5.4 Self-configuration/reconfiguration**

In the proposed concept, self-configuration is the procedure that makes it possible to spontaneously and automatically integrate a physical subsystem into the BAL. It concerns associating the sensors and actuators via WSAWs and creating dynamically an information model in the information system matching the physical subsystem.

The term re-configuration refers to the continually ongoing adjustment of the system to account for changes in the environment situation, such as removing or adding a new sensor/actuator.

#### **4.5.5 Interface to services/applications**

The upper interfaces that are exposed to application from BAL abstract away sensor and actuator data at a level corresponding to the states and associated attributes of the target subsystems, as defined above.

For monitoring purposes, an application/service can obtain the instantaneous state of an entity as a discrete state, together, if required with associated attributes. This state is estimated as a result of the fusion, aggregation and consolidation of data from multiple sensors available in the room. In the examples below, the state of a room could be whether it is occupied, the type of activity going on, the attributes could be its temperature, the number of persons present, etc.

For control purposes, an application can change the state of an entity to another state, if admissible, or change associated attributes. In the examples below, the state of a room could be changed to dark by sending coordinated commands to individual actuators, such as those controlling shades and light fixtures.

#### **4.5.6 REST interface**

An example implementation of Building Abstraction Layer interfaces with REST primitives would work as follows:

- Each target physical entity and each component of the state of these entities would be assigned an URI :
- A GET command at this URI would be used by a client application to retrieve the instantaneous value of these state components
- A PUT command would be used to set the value of these components to the parameter value transmitted in the command, if this value is permissible and the corresponding entity is actionable

### **4.6 Sensor & actuator Interface layer**

In the FINSENY D4.2 deliverable “Smart Buildings: coarse-grain architecture” a common interface layer for sensors and actuators present in the smart buildings has been proposed, assuming that all sensors and actuators available inside the building are shared between all building applications.

This assumption implies that the applications, services and BAL (Building Abstraction Layer) must ignore the type of physical connection such as serial, I2C port, Zigbee, PLC, KNX, BACNet, etc. that is used at this level. They should just be able to communicate with the physical devices through a unified interface that abstracts away the peculiarities of these specific communication technologies.

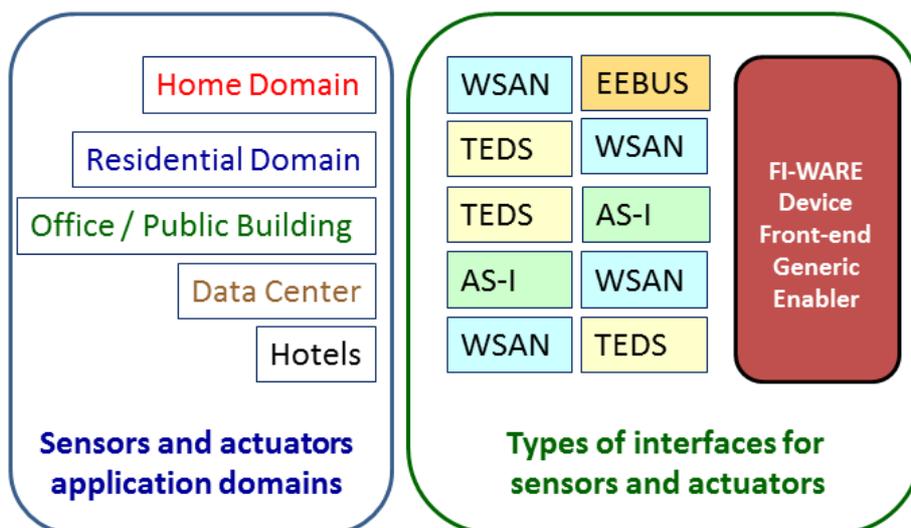


Figure 42: Sensor & actuator Interface and their application domains

That said, in this sub-chapter are treated, more in detail, the types of interfaces and their layers that best fit with the types of smart building domains previously analyzed in deliverable D4.1 and D4.2, as shown in **Figure 42**.

#### 4.6.1 TEDS interface layers

A Transducer Electronic Data Sheet (TEDS) is a standardized method of storing sensors and actuators identification, calibration, correction data, and manufacturer-related information.

TEDS formats are defined in the IEEE 1451 set of Smart sensors and actuators interface standards developed by the IEEE Instrumentation and Measurement Society’s Sensor Technology Technical Committee that describe a set of open, common, network-independent communication interfaces for connecting sensors and actuators to microprocessors, instrumentation systems, and control/field networks.

One of the key elements of the IEEE 1451 standards is the definition of TEDS for each sensors and actuators. The TEDS can be implemented as a memory device attached to the sensors or actuators and containing information needed by a measurement instrument or control system to interface with sensors and actuators.

TEDS can, however, be implemented in two ways. First, the TEDS can reside in embedded memory, typically an EEPROM, within the sensors/actuators itself which is connected to the measurement instrument or control system. Second, a virtual TEDS can exist as a data file accessible by the measurement instrument or control system. A virtual TEDS extends the standardized TEDS to legacy sensors and applications where embedded memory may not be available.

Using TEDS it is possible for a smart sensor to directly communicate measurements to a system. Networking of transducers (sensors or actuators) in a system and communicating transducer information via digital means versus analog cabling facilitates easy distributed measurements and control. TEDS sensors/actuators can provide flexibility, improve system performance, and ease system installation, upgrade, and maintenance. Thus, the trend in industry is moving toward distributed control with intelligent sensing architecture. These enabling technologies, in addition to the usual fields of application such as aerospace,

automotive, industrial automation, military and homeland defenses, they are now increasingly used in the smart buildings and homes domain.

For convenience, hereinafter, we will use the term "transducer" to indicate sensors or actuators.

**4.6.1.1 TEDS model**

The different modules of the smart transducer model can be grouped into functional layers as shown in **Figure 43**. The transducers and signal conditioning and conversion modules can be grouped into a building block called a Smart Transducer Interface Module (STIM). Likewise, the application algorithm and network communication modules can be combined into a single entity called a Network Capable Application Processor (NCAP). With this functional partitioning, transducer to network interoperability can be achieved in these manners:

- 1) STIMs from different sensor manufacturers can “plug and play” with NCAPs from a particular sensor network supplier,
- 2) STIMs from a sensor manufacturer can “plug and play” with NCAPs supplied by different sensor or field network vendors,
- 3) STIMs from different manufacturers can be interoperable with NCAPs from different field network suppliers.

Using this partitioning approach, a migration path is provided to those sensor manufacturers who want to build STIMs with their sensors, but do not intend to become field network providers. Similarly, it applies to those sensor network builders who do not want to become sensor manufacturers.

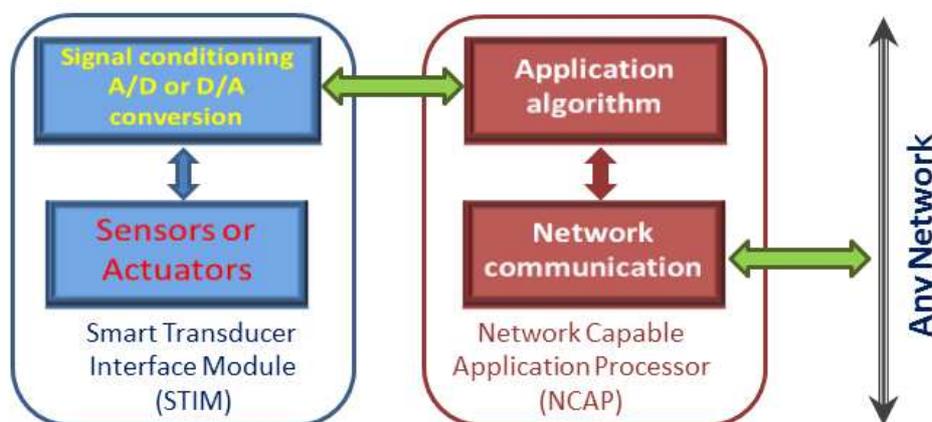


Figure 43: Functional layers in TEDS interface

**4.6.1.2 TEDS applications through networking**

With the emergence of computer networking technology, transducer manufacturers and users alike are finding ways to apply this networking technology to their transducers for monitoring, measurement, and control applications. Networking smart sensors provides the following features and benefits:

- enable peer-to-peer communication and distributed sensing and control,
- significantly lower the total system cost by simplified wiring,
- use prefabricated cables instead of custom laying of cables for ease of installation and maintenance,
- facilitate expansion and reconfiguration,
- allow time-stamping of sensor data,

- enable sharing of sensor measurement and control data,
- provide Internet connectivity, meaning *global* or *anywhere*, access of sensor information.

A distributed measurement and control system can be easily implemented based on the IEEE 1451 standards.

An application sample based on model of IEEE 1451 is shown in **Figure 44**. Three NCAP/STIMs are used to illustrate the distributed control, remote sensing or monitoring and remote actuating. In the first scenario, a sensor and actuator are connected to the STIM of NCAP 1, and an application software running in the NCAP can perform a locally distributed control function, such as maintaining a constant temperature for a bath.

The NCAP reports measurement data, process information, and control status to a remote monitoring station or host. It frees the host from the processor-intensive, closed-loop control operation.

In the second scenario, only sensors are connected to NCAP 2, which can perform remote process or condition monitoring functions. In the third scenario, based on the broadcast data received from NCAP 2, NCAP 3 activates an alarm when the predetermined level of any physical quantity exceeds a critical set point. As illustrated in these examples, IEEE 1451-based sensor network can easily facilitate peer-to-peer communications and distributed control functions.

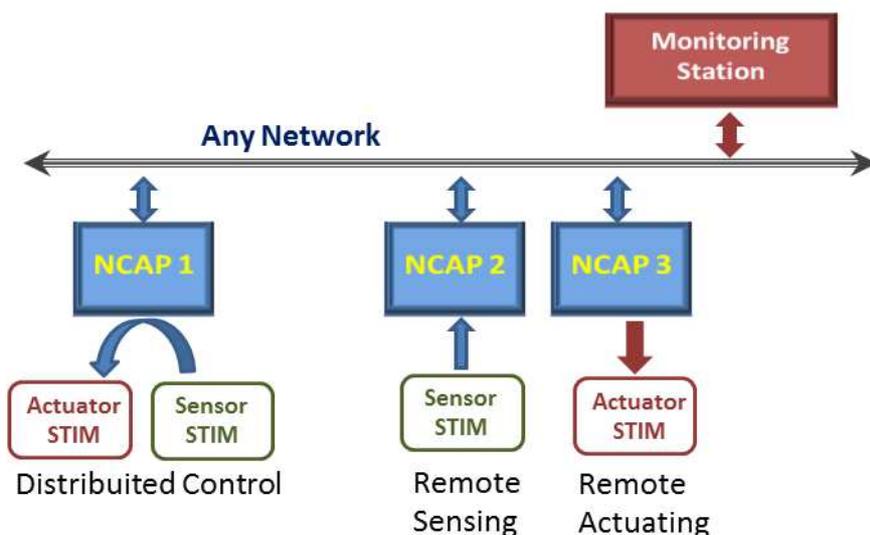


Figure 44: Application model of IEEE 1451 interface

#### 4.6.1.3 TEDS standard interfaces

The Technical Committee on Sensor/actuators Technology of the Institute of Electrical and Electronics Engineer (IEEE)'s Instrumentation and Measurement Society sponsored a series of projects for establishing a family of IEEE 1451 Standards. These standards specify a set of common interfaces for connecting transducers to instruments, microprocessors, or field networks.

The TEDS has many benefits:

- Enable self-identification of sensors or actuators - A sensor or actuator equipped with the IEEE 1451 TEDS can identify and describe itself to the host or network via the sending of the TEDS.
- Provide long-term self-documentation - the TEDS in the sensor can be updated and stored with information such as location of the sensor, recalibration date, repair record, and many maintenance-related data.
- Reduce human error - automatic transfer of TEDS data to the network or system eliminates the entering of sensor parameters by hands which could induce errors due to various conditions.
- Ease field installation, upgrade, and maintenance of sensors - this helps to reduce life cycle costs because only a less skilled person is needed to perform the task by simply using “plug and play”.

IEEE 1451, designated as Standard Transducer Interface for Sensors and Actuators, consists of the following document standards:

IEEE 1451.0: Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats.

IEEE 1451.1: Network Capable Application Processor (NCAP) Information Model for Smart Transducers.

IEEE 1451.2: Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats.

IEEE P1451.3: Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems.

IEEE P1451.4: Mixed-mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats.

IEEE P1451.5: Wireless Communication and Transducer Electronic Data Sheet (TEDS) Formats.

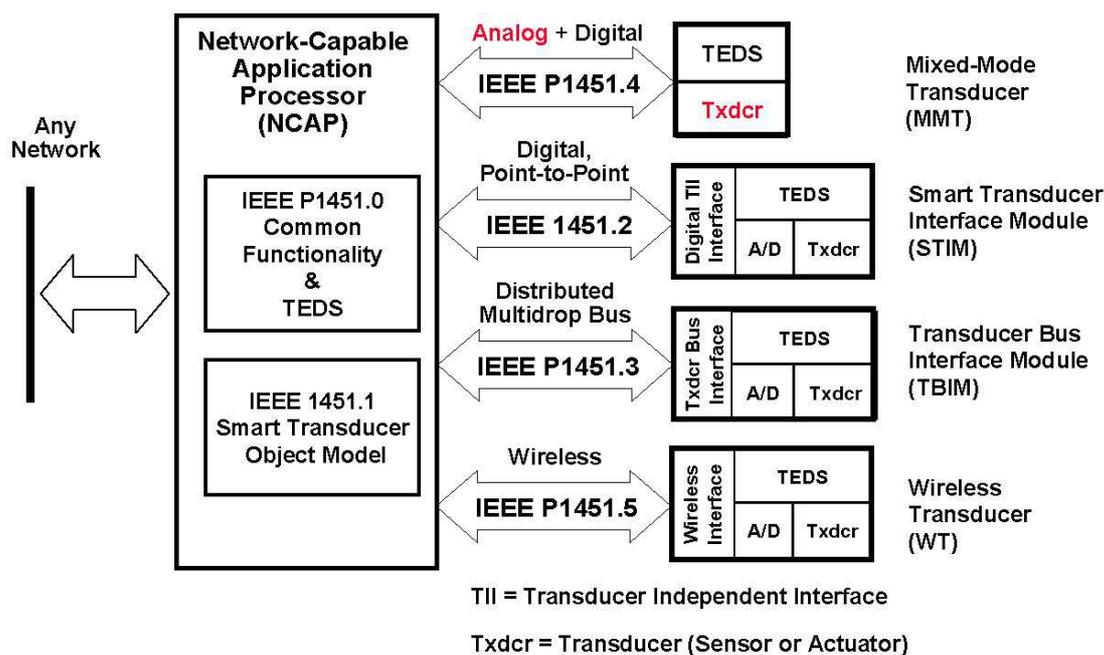


Figure 45: Family of IEEE P1451 Standards

In particular, the IEEE 1451.2 standard defines a TEDS, its data format, and the digital interface and communication protocols between the STIM and NCAP interface layer. A conceptual view of IEEE 1451.1 NCAP is shown in Figure 46, which uses the idea of a “backplane” or “card cage” to explain the functionality of the NCAP, while in Figure 47 the NCAP detail is shown.

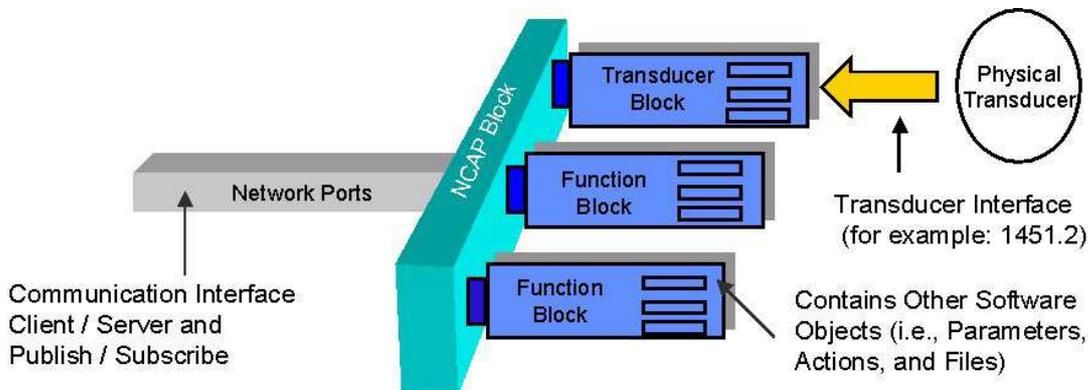


Figure 46: Conceptual view of IEEE 1451.1

The NCAP centralizes all system and communications facilities. Network communication can be viewed as port through the NCAP and communication interfaces support both client-server and publish-subscribe communication models. Client-server is a tightly coupled, point-to-point communication model, where a specific object, the client, communicates in a one-to-one fashion with a specific server object, the server.

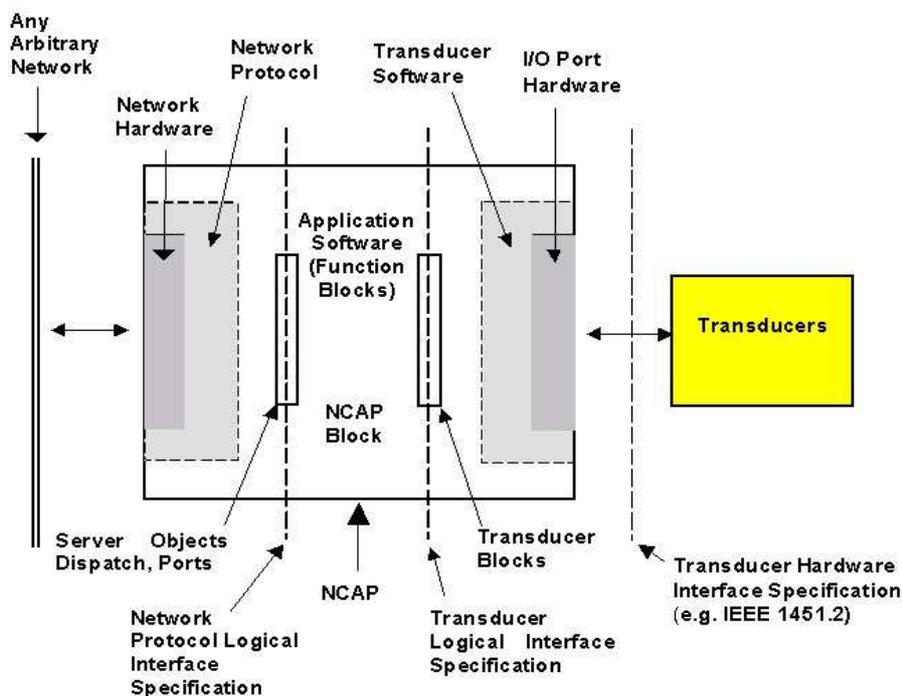


Figure 47: IEEE 1451 logical interface

On the other hand, the publish-subscribe communication model provides a loosely coupled mechanism for network communications between objects, where the sending object, the publisher object, does not need to be aware of the receiving objects, the subscriber objects. The loosely coupled, publish-subscribe model is used for one-to-many and many-to-many communications.

### 4.6.2 AS-Interface

The Actuator Sensor Interface, or AS-Interface or AS-i, was developed by a group of sensor/actuators manufacturers. It has become the standard for discrete sensors and actuators in process industries around the world. It is also a bus system, used for low-level field applications in data center to communicate with small binary sensors and actuators using the AS-Interface standard.

This modernizes automation systems effectively and eliminates wire bundles completely, with only one wire cable required for all devices, compared to one cable from each device needed for point-to-point wiring. Junction boxes are also eliminated, and the size of the control cabinet needed is significantly reduced. Plug-and-play wiring supports all typologies.

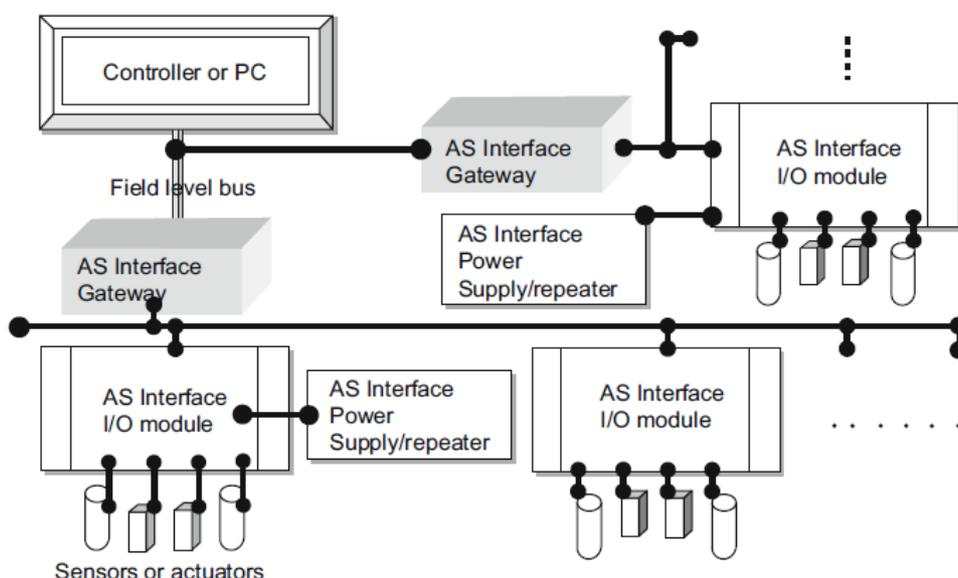


Figure 48: Actuator Sensor Interface architecture

In an AS-Interface architecture, a programmable controller such as a PLC, SCADA, or PC controls sensors and actuators via field-level buses, including Foundation Fieldbus, Profibus, etc.

As displayed in **Figure 48**, the AS-Interface has gateways directly connected to the field-level bus and the I/O module. The latter is the device which contacts the sensors and the actuators. A field-level bus may be able to support several AS-Interface gateways depending on the manufacturing specification and the system design, each of which fits a segment of a control system.

The AS-Interface is a single-master system. This means that there is only one master per AS-Interface network to control the operations of process. This polls all AS-Interface slaves one after the other and waits for a response.

As shown in **Figure 49**, the first interface is between the master CPU and the master communication processor; the second is between the master communication processor and AS-Interface cable. Process data and parameter assignment commands are transferred via the first interface, and user programs have suitable function calls and mechanisms available for reading and writing via this interface. On the other hand, information is exchanged with the AS-

Interface slaves via the second interface between the master communication processor and AS-Interface cable.

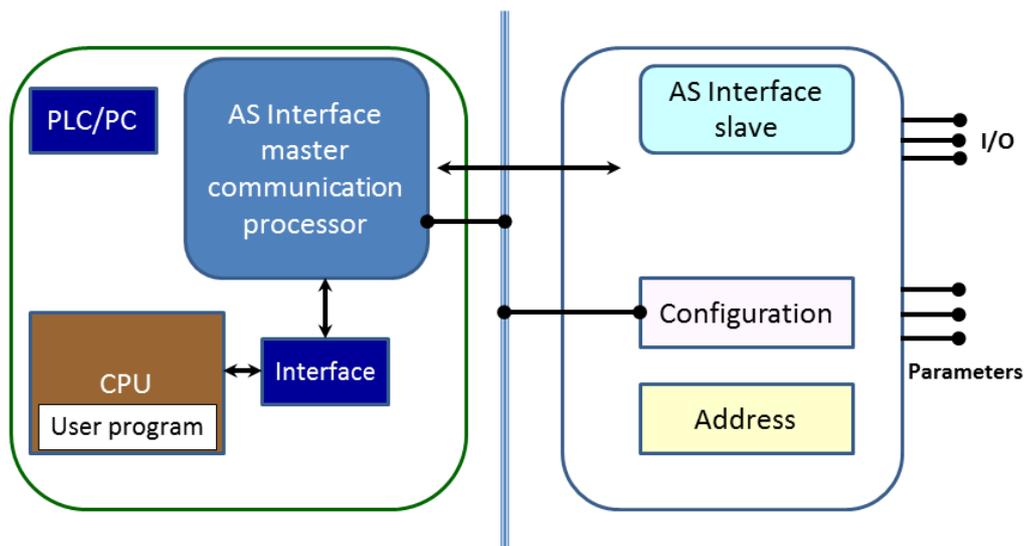


Figure 49: Actuator Sensor Interface operation

Various functions are available on the interface to control the master and slave interaction from the user program.

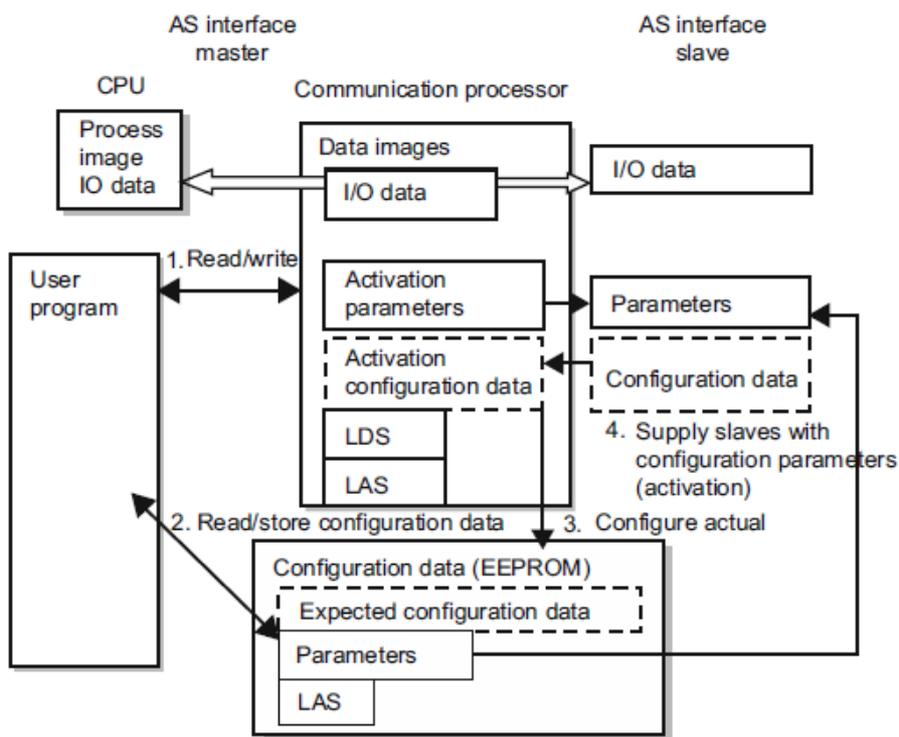


Figure 50: How an Actuator Sensor Interface performs an operation

The possible operations and the direction of data flow are illustrated in **Figure 50**, where the numbers 1, 2, 3 and 4 have the following meanings:

- (1) Read/write. When writing, parameters are transferred to the slave and to the parameter images on the communication processor; when reading, parameters are transferred from the slave or from the communication processor parameter image to the CPU.
- (2) Read and store (configured) configuration data. Configured parameters or data are read from the nonvolatile memory of the communication processor.
- (3) Configure actual. When reading, the parameters and configuration data are read from the slave and stored permanently on the communication processor; when writing, the parameters and configuration data are stored permanently on the communication processor.
- (4) Supply slaves with configured parameters. Configured parameters are transferred from the nonvolatile area of the communication processor to the slaves.

### 4.6.3 EEBUS interface layers

The EEBus forms the interface between in-house communication and the energy supplier. For this purpose, the EEBus supplies an application-neutral standardized interface. It connects the IP world of the Smart Grid and Smart Phone to the still-dominant non-IP-networks in the home automation area. The EEBus basic idea is:

- Unified technical solution for the connection of energy world apartment,
- IP address for each apartment
- Comprehensive device network for load management (devices are parts of Smart Grid)

Regarding household, it integrates the entire sensor and actor universe into a continuous communications system for efficient energy management and better comfort at home.

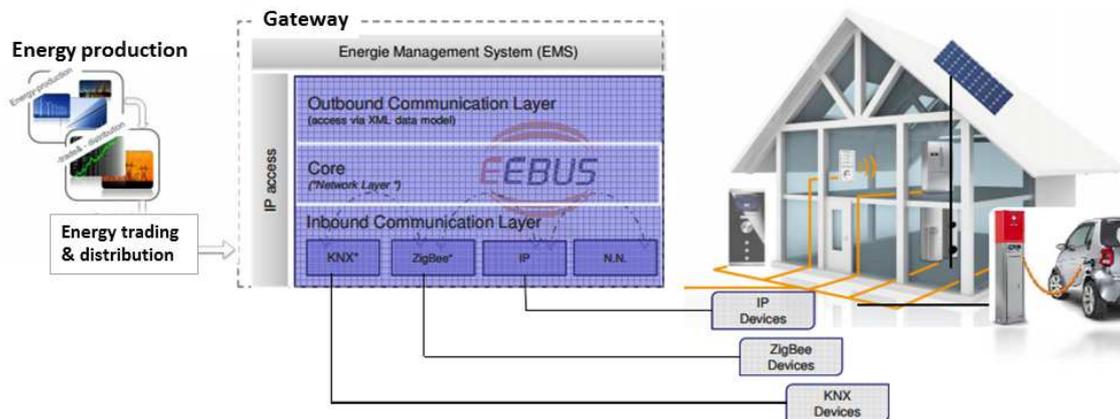


Figure 51: EEBus interface layers

With reference to **Figure 51**, the communication is divided into two: in one hand, the local data exchange within the house and on the other hand, the connection between energy suppliers and home access. Both sections are characterized by different requirements and conditions:

For the communication within the company already exist numerous standards, which currently account EEBus KNX, ZigBee and TCP / IP. KNX has different transmission media available, of which currently employs EEBus data transmission via power line (KNX power line or short KNX PL) and by radio in the 868 MHz band (short KNX RF).

ZigBee is a wireless solution in the 2.4 GHz band. Ethernet with TCP / IP does eventually come into play when an existing computer network can be shared (see "Inbound communication layer" in the graph). For each of these systems there are in EEBus both the right hardware

platform for the physical connection and its own software adapters for the integration of the respective protocol. This modular design allows for future additions.

The differences in the technical equipment of households are diversified. Utility companies can in his communication with them not to consider any option, but requires a uniform as possible, the individual device abstract view on the budget.

The Core or network layer has two functions: first, where there is the translation between the utility derived from the abstract XML data and the actual situation in the household, for example by electricity price signals are translated into the minutes of the household appliances. The second task is the in-house data exchange with which the devices communicate across borders log example above, in which weight they contribute to a reduction in power.

The EEBus uses the aforementioned standards and expand them where necessary. A new physical layer of KNX PL + EEBus uses the CENELEC B band and applies to the OFDM method which is an approximately 15-fold higher data rate than in the past. Backward compatibility with conventional PL devices is ensured.

To better explain the difference with the classic architectures, in **Figure 52** is shown the comparison between standardization architecture and EEBus.

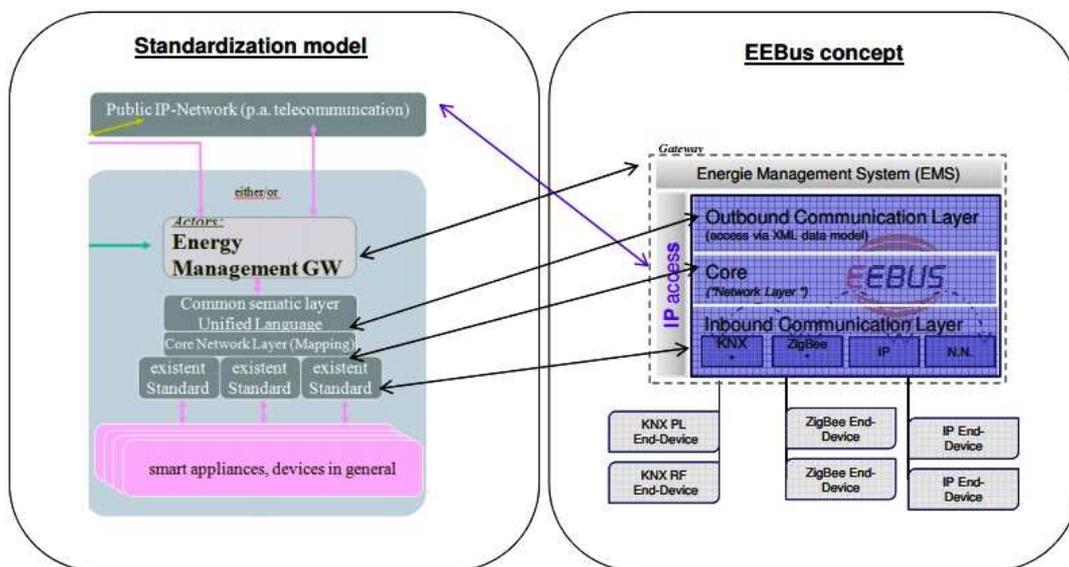


Figure 52: Comparison between standardization architecture and EEBus.

The use of power management features requires standardized descriptions etc. for price information, data or device classifications that specify the manner in which a terminal can participate in home energy management.

This information constitutes a common understanding of all participating devices and is the basis for cooperation. Missing it, no applications or services are exchanged. Such functional specifications for energy management include ZigBee partially present, in contrast, is not practical nor KNX. The EEBus adds these missing definitions.

EEBus used in its basic functions (e.g. for multithreading, XML parsing), makes use of Qt framework (*QT is a cross-platform application framework that is widely used for developing application software with a graphical user interface*). This is EEBus stack on multiple platforms can be used, such as Linux, Windows or MAC. The EEBus architecture and interface layers are shown in **Figure 53**.

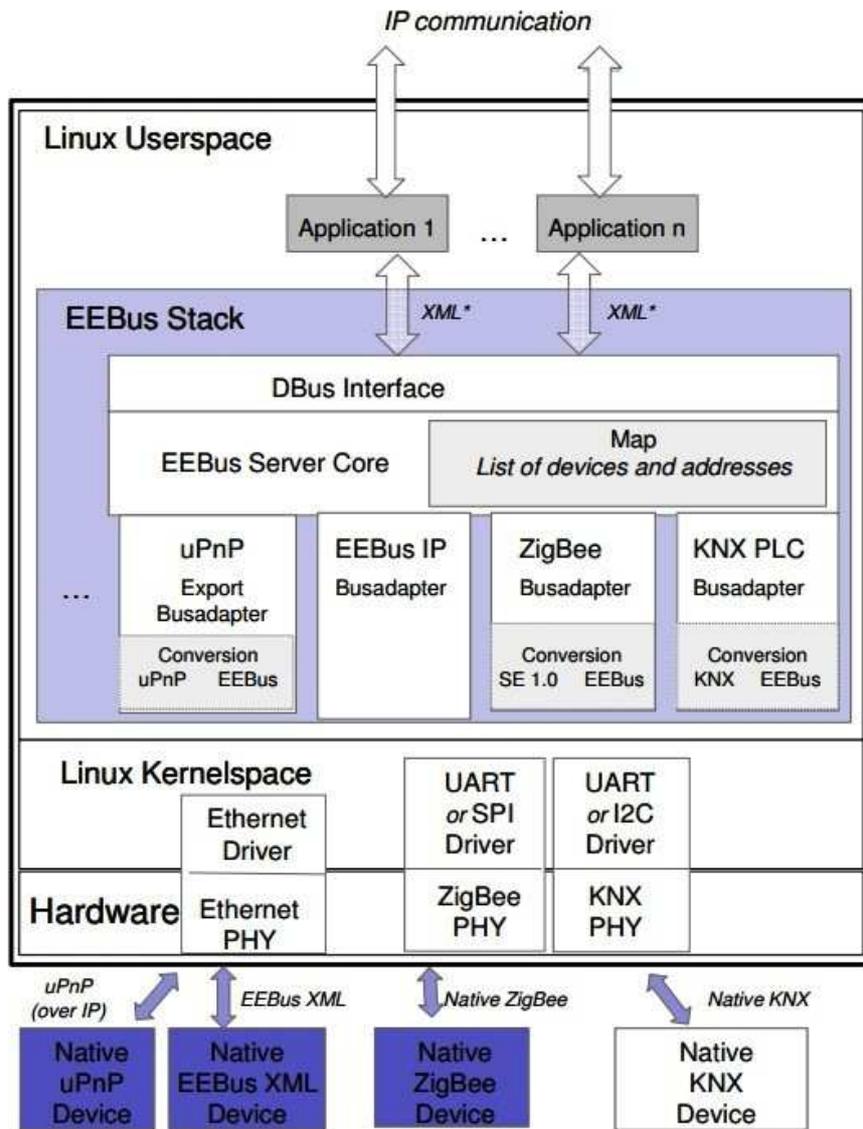


Figure 53: The EEBus architecture and interface layers

### 4.6.4 WSAN interface layers

Even if the needs and the management of the energy depend on the various specific building sub-domains (home, residential, office, public building and data center etc.) the generic components architecture as a whole is substantially the one illustrated in the Figure 54, where are shown the main actors involved in an control system infrastructure for the Smart Buildings.

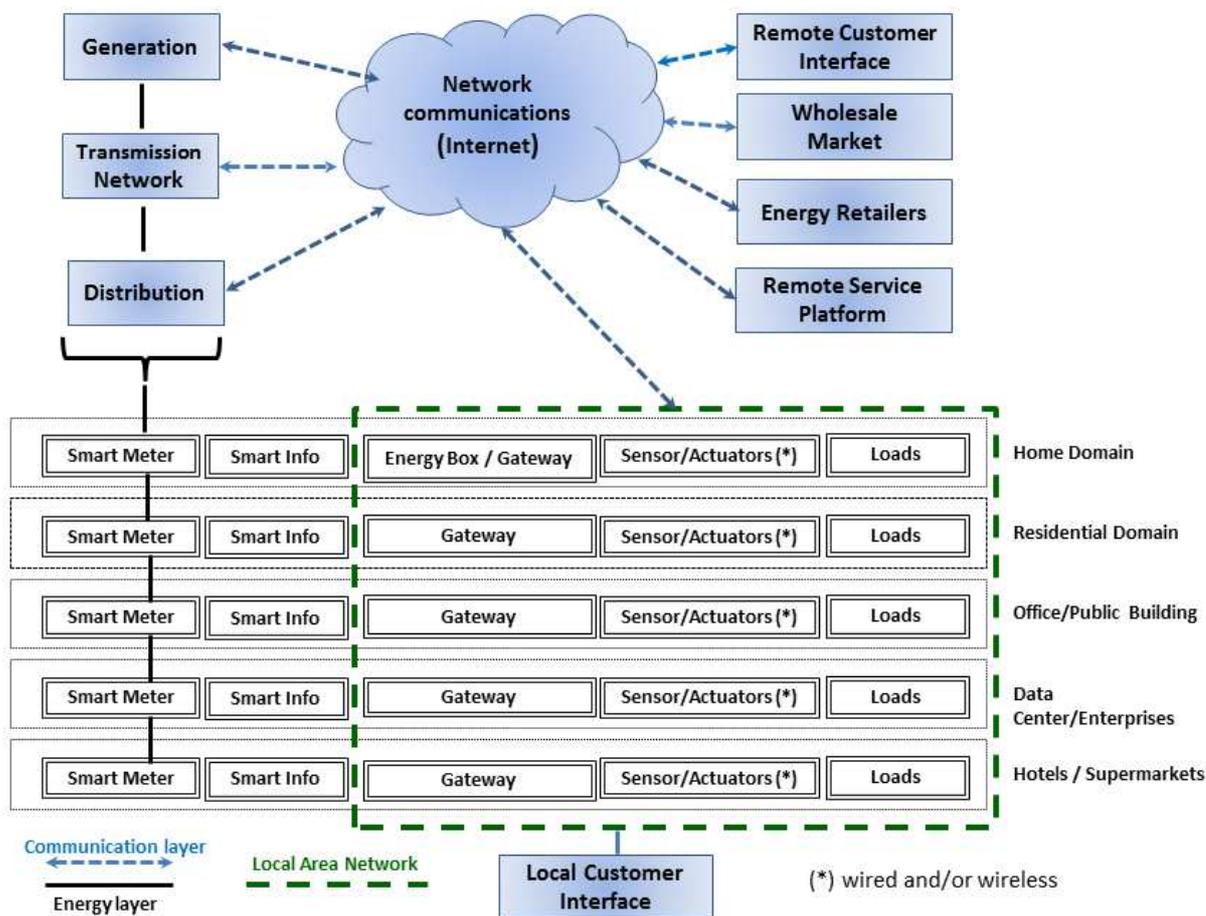


Figure 54: Component architecture for smart buildings

Compared to the wired sensors and actuators technologies, the WSAN (*Wireless Sensor/Actuator Network*) have provided new options that overcome challenging installation constraints and remove physical or financial constraints associated with hard wiring for “smart buildings” systems. Wireless environments are easily adapted to changing business needs or new facility requirements. Also, eliminating wiring and its related effort accelerates the installation process and simplifies retrofitting and system extension. Wireless- based systems offer building owners and facility managers more choices and fewer constraints, including: ease of deployment, cost benefits, scalability of the network and simpler and more flexible system design.

As can be seen from the **Figure 54**, the interface between the WSAN world and the rest of the management system of the optimization of power is represented by the “Gateway”.

In the “Smart Home” context, it also used the term "Energy Box", a WSAN interface device composed by two components:

- A Home Gateway (*ADSL*)
- OSGi (Open Service Gateway initiative) framework with HAN wireless communication capability. OSGi is a Java framework for developing and deploying modular software programmes and libraries. It has two parts:

- the first part is a specification for modular components called bundles, which are commonly referred to as plug-ins. The specification defines an infrastructure for a bundle's life cycle and determines how bundles will interact.
- the second part of OSGi is a Java Virtual Machine (JVM)-level service registry that bundles can use to publish, discover and bind to services in a service-oriented architecture (SOA).

The Energy Box can offer a WEB user interface and provide an Execution Environment (OSGi framework for Java) to host third-party application (e.g. a SW component implementing the algorithm to calculate the energy price at a given time, provided by the Energy Retailer).

For specific end customer services that require more substantial computing power, the computation of the algorithms is redirected to the remote service platform that operates in a cloud computing environment as shown in **Figure 55**.

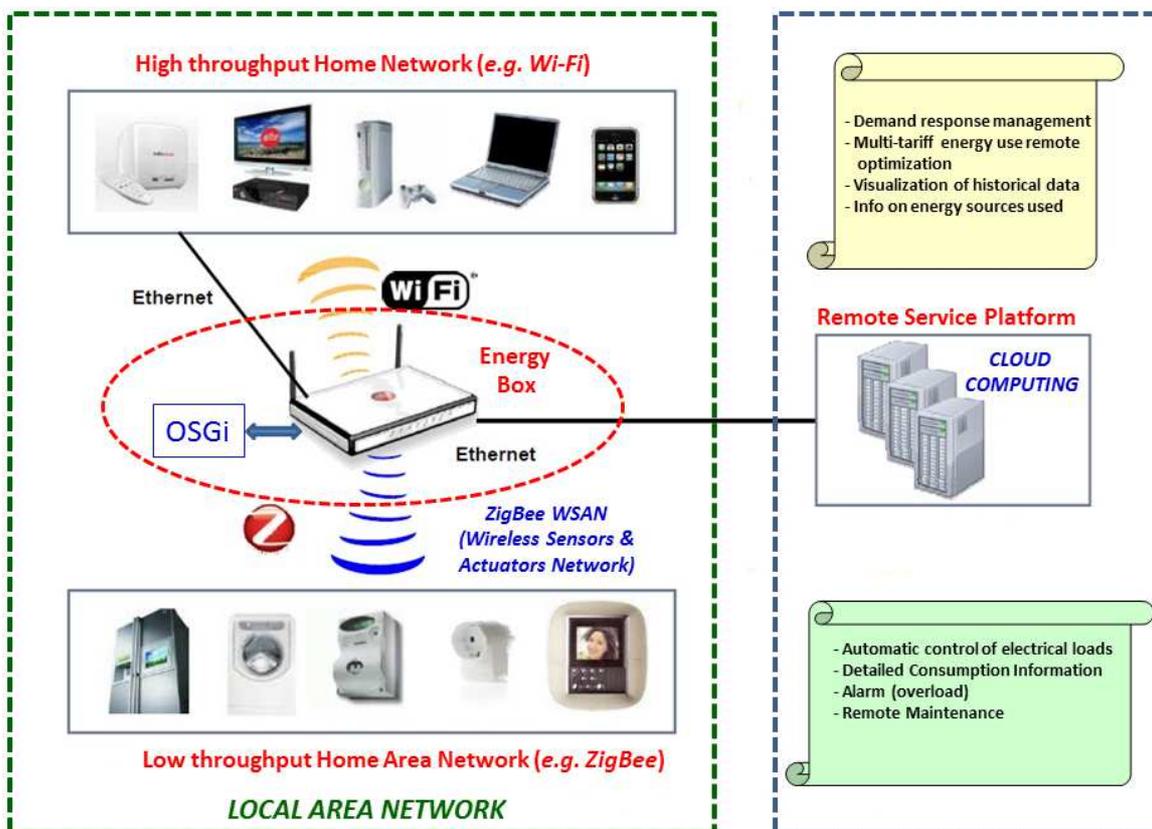


Figure 55: Energy Box as WSA interface.

The **Figure 56-a** shows the WSA topology where the sensors send the collected data to the Energy Box entity that execute certain control algorithms to produce control commands and send them to actuators. Finally, the actuators perform the actions. In this context, both the sensor data and control commands need to be transmitted wirelessly in a single-hop or multi-hop fashion. A high-level view of the applications flow of this infrastructure is depicted in **Figure 56-b**.

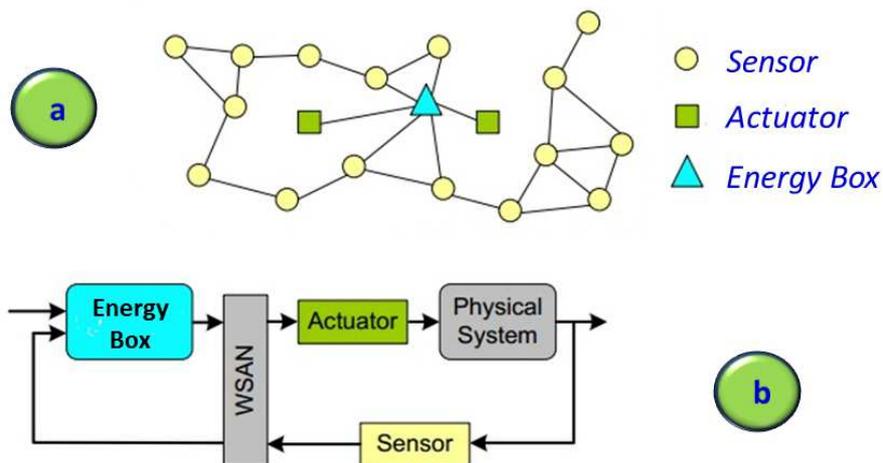


Figure 56: WSAN topology (a) and abstraction of control application (b)

From the radio-node point of view, there are many short-range wireless communications technologies available now in IT, automation and other mobile data-communication fields. The wireless technologies related to WSAN applications are mainly:

- ZigBee technologies;
- 802.11 compliant (Wi-Fi) technologies;
- Bluetooth technologies;
- other proprietary technologies.

The above “Energy Box” considerations, made for the “Home” environment, are applicable for any kind of smart buildings with the use of the “Smart Gateway”. More in detail, the gateway can be divided into three blocks: the WSAN connection mechanism, the Center Control Unit and the interconnection section, as shown in **Figure 57**.

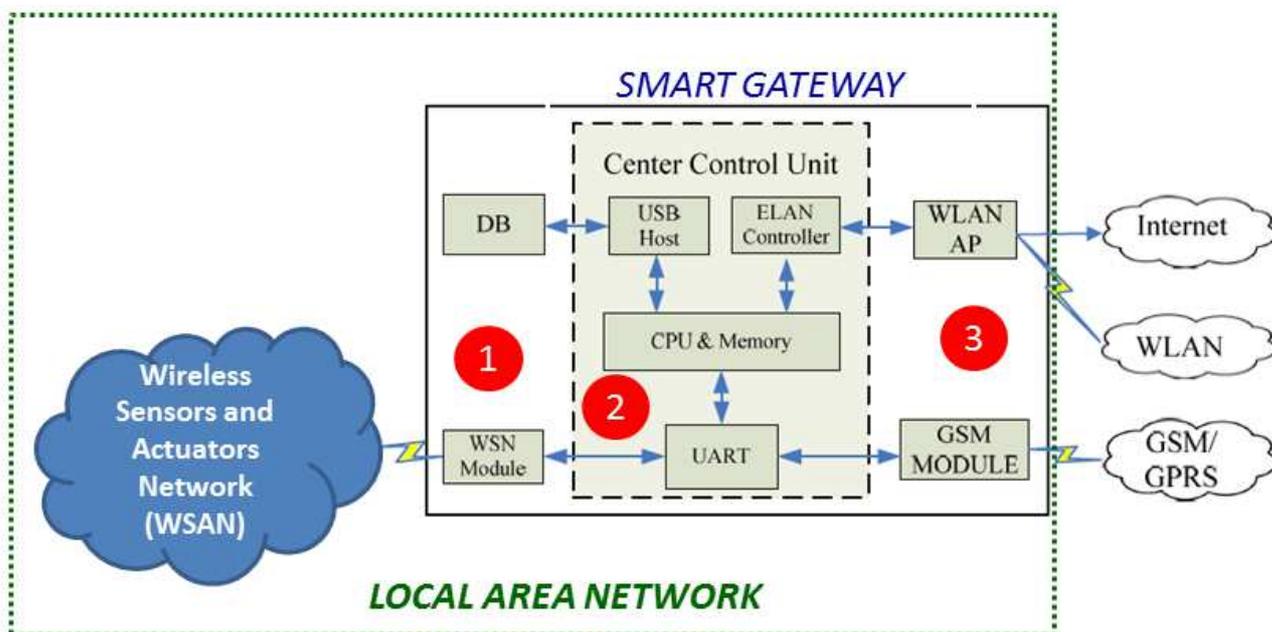


Figure 57: The three blocks of the Smart Gateway

The first block shown in **Figure 57** is the WSAN connection mechanism. The gateway has to receive data from sensor/actuator nodes and gives commands to them. To achieve this goal, the gateway has to provide mechanism to join into the WSAN. Through this part, the gateway provides connection to WSAN in both hardware and software ways. At hardware way, the

gateway has to handle interface compatible and signal translation and rate conversion. For software, it has to translate the protocol using in WSN and extract the valuable data which can be used by the program using in the gateway.

The second part is center control unit which requirements are as follow: enough storage space, fast enough processing speed and adequate external interfaces. Recently, the inclusion of wireless sensor networks in the Internet became feasible due the standardization of IPv6 over Low-power Personal Area Networks (6LoWPANs), which introduces an adaptation layer below IP, enabling the transmission of IPv6 datagrams over IEEE 802.15.4 wireless links.

Basically, the adaptation layer performs header compression of IPv6 and transport layer headers, creating a new header composed by a few bytes. Inside the LoWPAN, nodes do not need to decompress this header, since every node knows the compressed field’s contents. For instance, nodes use 16-bit addresses or link-layer 64-bit unique IDs instead of full IPv6 addresses because their prefixes are the same for all nodes. The adaptation layer also handles packet fragmentation and reassembly in order to support the IPv6 minimum MTU (maximum transmission unit). These modifications made IPv6 suitable for low-power devices, including sensor nodes.

**Figure 58** shows the structure of the gateway that establishes communication between end users and a WSN. This architecture is composed by four main elements: gateway, IPv6 hosts, IPv4 hosts, and the 6LoWPAN WSN.

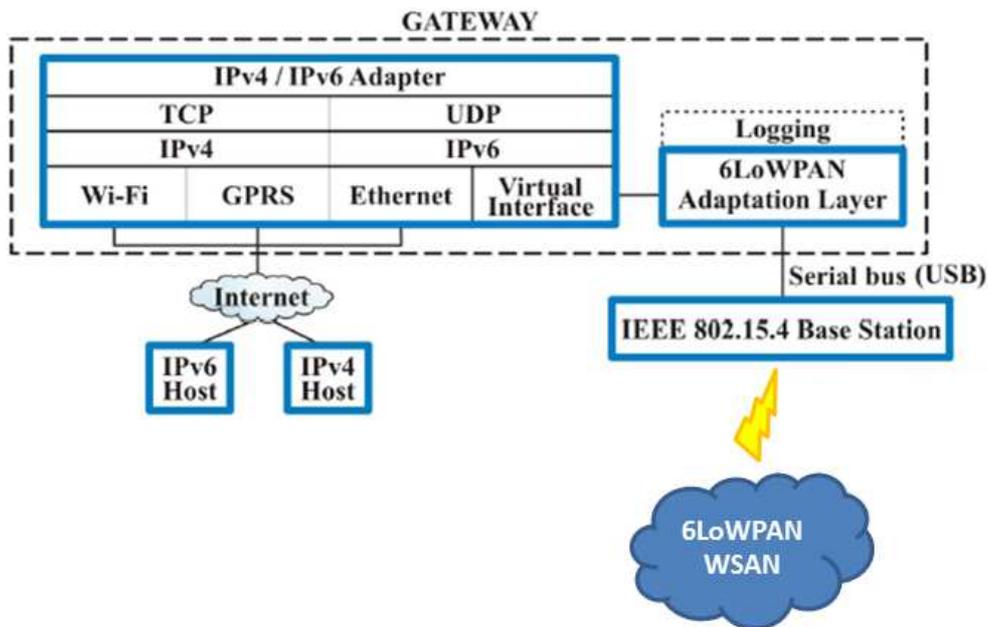


Figure 58: Interface layer of the 6LoWPAN gateway

In this case the gateway is an interface responsible for integrating the 6LoWPAN WSN in IPv6 networks. Its main tasks involve compression and decompression of IPv6/6LoWPAN headers, in order to forward 6LoWPAN messages from the WSN to external IPv6 networks and vice versa, and fragmentation/reassembly of large messages.

To do these tasks, the gateway must have a 6LoWPAN adaptation layer, and it needs to be set up to route IPv6 packets. WSN users can be represented as IPv6 hosts or IPv4 hosts. The former are capable of communicating with sensor/actuators nodes directly. Therefore, they can send commands for specific nodes or test whether a node is reachable or not. IPv4 hosts cannot connect to 6LoWPAN WSN directly, due the incompatibility between IPv4 and IPv6. In this case, the gateway acts like a proxy for them, becoming responsible for sending commands to a specific node, using an IPv6 packet, and returning the result back to the IPv4 host, using an IPv4 packet.

The 6LoWPAN WSN is a network capable of routing IPv6 packets through 6LoWPAN compression and fragmentation techniques. In order to do this, it is necessary that all nodes in the WSN support the 6LoWPAN stack. As the transport layer protocol, user datagram protocol (UDP) is used because it is lightweight and has low complexity when compared with transport control protocol (TCP). However, TCP can also be used.

**Figure 59** illustrates the exchange of messages between IPv6 clients and the 6LoWPAN WSN through the gateway. The first part of the figure, represented by the symbol “(A)”, shows how the IPv6 client sends a command to the sensor node, whereas the second part “(B)” shows the opposite. The gateway plays the role of an IPv6 router, with the additional functionality of compression/decompression/fragmentation/reassembly of the IPv6/6LoWPAN packets, and registration of the operations in the log file. In WSN based on multi-hop communication, sensor/actuators nodes forward the 6LoWPAN packet to the destination, which will process it. If the WSN is designed for only send information to an IPv6 client, the communication between them is represented just by the second part.

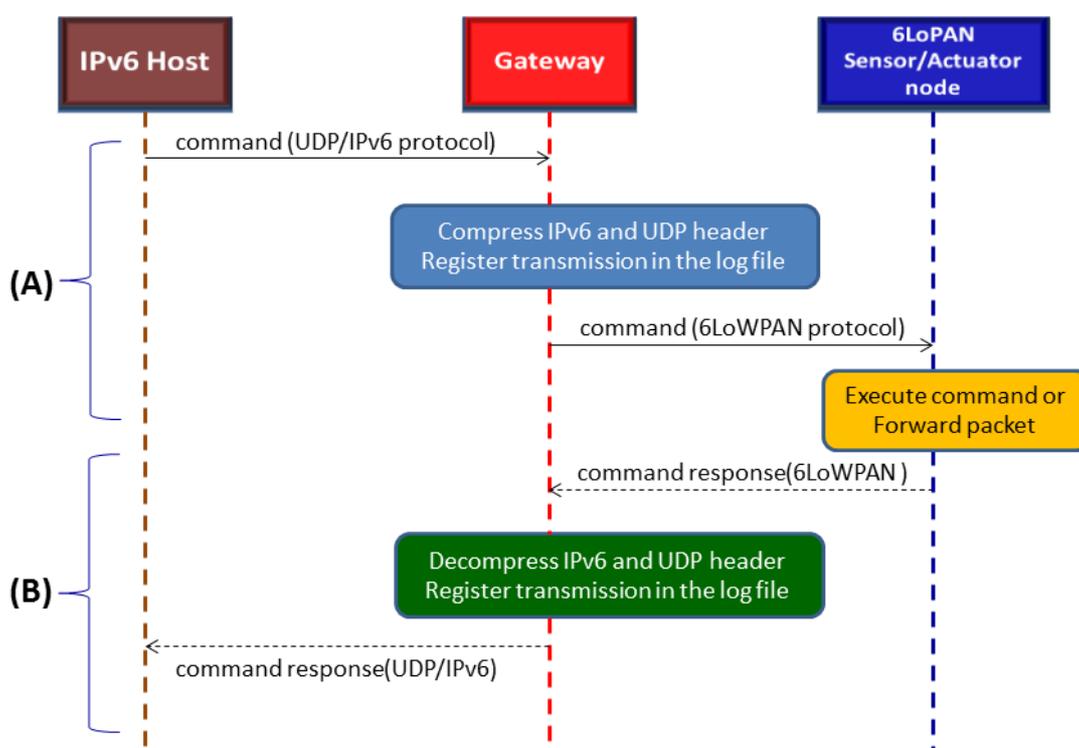


Figure 59: Message flows between IPv6 hosts, gateway and 6LoWPAN wireless sensor network.

- 
- The gateway enabled IPv6 hosts to communicate directly with sensor/actuators nodes, obtaining their readings or testing their connectivity in real time.

In the third block of **Figure 57** there is interconnection mechanism. In this part, the gateway provides physical connection to Internet, WLAN and GSM/GPRS network and finish protocol and signal translation and rate conversion. For Internet connection, an ELAN (*Emulated Local Area Network*) controller is needed. It can be provide either by an external unit or integrate in the center control unit.

On the basis of this controller, coupled with the WLAN Access Points, then the gateway is able to connect to a WLAN. This WLAN access point allows the gateway to connect to wireless

---

network using Wi-Fi (or other technologies) to the destination host within the wireless signal coverage of this AP.

There are two steps to connect to internet and WLAN. First step, the gateway uses the IP address provides by the server in the internet. Second, the gateway provides DHCP to build up WLAN with an extra wireless AP. These two steps can be implemented by center control unit or a powerful wireless router. When this task is completed by center control unit, system will have a higher integration and the wireless AP can be simpler and smaller size.

## 5 Mapping onto FI-WARE Generic Enablers

This chapter reports how the FI-WARE Generic Enablers (GE) maps with the Smart Building Functional Architecture High-level Building Blocks defined and described in chapter 4 wherever this is applicable.

The GEs defined by FI-WARE are described in the FI-WARE Product Vision [4]. It provides a high-level description of the FI-WARE Platform aiming at providing a framework for development of smart applications in the Future Internet.

Next sections give a short description of the FI-WARE GEs and the mapping itself between the different components of the defined functional architecture in the Smart Buildings scenario work package and the GEs.

### 5.1 FI-WARE Framework

Over the past few years, a number of technology innovations took place and started paving the way for alternative directions in the ICT landscape and new business opportunities. The era of the Future Internet in Europe is already ongoing and triggered by the following advancements:

- Industrialization and process-standardisation of IT. Cloud computing and open-service delivery platforms are changing the provisioning framework of ICT products. The new model considers all ICT products as services, which can be provisioned on a pay-per-use basis.
- Increased network capacity and bandwidth. New wireless networking technologies, such as LTE (4G), and the deployment of Fibre to The Home (FTTH) are offering enough capacity to new applications with enriched content, irrespective of the underlying physical medium.
- Internet of Things and ubiquitous computing. The vision of ubiquitously connecting intelligent devices and sensors and offering new possibilities for contextual and environmental information sensing, processing and analysis is taking up. To this respect, the door is opened to new automation and control applications and services in various sectors.
- Internet of Services. Internet has accelerated the creation of complex value networks of service providers, consumers and intermediaries, bringing to businesses and customers innovative applications, better tailored to their needs. These networks increasingly span various different players that historically have worked largely separated from each other, thereby leading to more agile and dynamic business relationships and environments never seen before.

This Future Internet will address some key demands and expectations from various market stakeholders. More particularly:

- End customers are seeking to gain access and easily consume services that can effectively assist them in their daily lifetime. Some of the underlying problems involved are the management of the ever-growing information and the seamless and uninterrupted access from anywhere, at anytime and from any device. Furthermore, customers are asking for improved means of communication and collaboration within their social networks, families and neighbourhoods, in real-time and while being mobile, meeting all the related security and privacy requirements.
- Enterprises and organizations are seeking to offer their customers a more compelling user experience and better services. For that reason, they are targeting service offerings with more personalized interaction experience, exploiting contextual user data. In order to develop and operate their services, new methods, technologies and tools are needed to speed up the time to market, to establish value added services, which may be better configured in partnership with others and to simplify access to relevant resources and capabilities.

- Application/service developers and providers are challenged to build smart applications that cover the above mentioned needs from consumers and enterprises. From a development perspective, such applications and services should be built on top of powerful, but easy to use APIs, be based on standards and offer flexible deployment, provisioning and management frameworks. Additionally, they should exploit economies of scale and protect investments in the long run. Finally, the ability to combine applications from different sources necessitates innovative revenue sharing models across partners and potentially also customers.

In the framework of the Future Internet, FI-WARE project is building a Core Platform, the “FI-WARE Platform” or simply “FI-WARE”, which targets to increase the global competitiveness of the European ICT economy, by introducing an innovative infrastructure for cost-effective creation and delivery of versatile digital services, providing high QoS and security guarantees. More particularly, FI-WARE is an open platform, based on elements, called Generic Enablers (GE), which offer reusable and commonly shared functions, serving a multiplicity of Usage Areas across various sectors. The ability to serve a multiplicity of Usage Areas distinguishes GEs from what would be labelled as Domain-specific Common Enablers (or “Specific Enablers” for short), which are enablers that are common to multiple applications, but all of them specific to a very limited set of Usage Areas.

The key objectives of FI-WARE are the identification and specification of GEs, together with the development and demonstration of respective reference implementations. Any implementation of a GE comprises a set of components and offers capabilities and functionalities that can be flexibly customized, used and combined for many different Usage Areas, enabling the development of advanced and innovative Internet applications and services.

### 5.2 Description of FI-WARE Generic Enablers

The high level goal of the FI-WARE project is to build the Core Platform of the Future Internet. Generic Enablers (GE) are the building blocks of FI-WARE platform. Any implementation of a Generic Enabler (GE) is made up of a set of components which together support a concrete set of Functions and provide a concrete set of APIs and interoperable interfaces that are in compliance with open specifications published for that GE.

The Core Platform to be provided by the FI-WARE project is based on GEs linked to the following FI-WARE Technical Chapters each of them having several GEs. More detailed information can be found in the FI-WARE product vision wiki page [4].

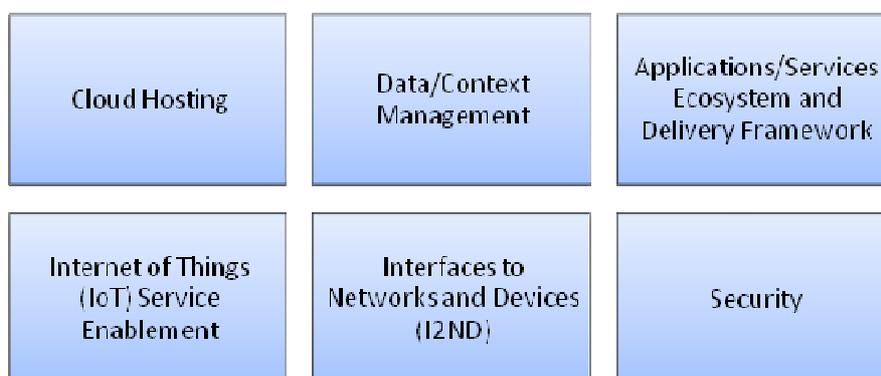


Figure 60: FI-WARE Chapters

- **Cloud Hosting** – the fundamental layer which provides the computation, storage and network resources, upon which services are provisioned and managed. The list of GE belonging to this chapter are:
  - **IaaS DataCenter Resource Management:** provides the basic Virtual Machine (VM) hosting capabilities, as well as management of the corresponding resources within the DataCenter that hosts a particular FI-WARE Cloud Instance.
  - **IaaS service Management:** introduces a layer on top of IaaS Resource Manager GEs in order to provide a higher-level of abstraction to Application/Service providers.
  - **PaaS Management:** provides to the users the facility to manage their applications without worrying about the underlying infrastructure of virtual resources (VMs, virtual networks and virtual storage) required for the execution of the application components.
  - **Object Storage:** comprises a storage service that operates at a more abstract level than that of block-based storage.
  - **IaaS Cloud-edge Resource Management:** “cloud proxy”, located in the home system and equipped with storage and cloud hosting capabilities
  - **Resource Monitoring:** for monitoring the different GEs (via exposed metrics) and their interaction
- **Data/Context Management** – the facilities for effective accessing, processing, and analyzing massive volume of data, transforming them into valuable knowledge available to applications. The list of GE belonging to this chapter are:
  - **Publish/Subscribe Broker:** enables publication of events by entities so they become available to other entities which are interested in processing the published events.
  - **Complex Event Processing:** analysis of event data in real-time to generate immediate insight and enable instant response to changing conditions
  - **Big Data Analysis:** processes huge amounts of previously stored data in order to get relevant insights in scenarios where latency is not a highly relevant parameter.
  - **Compressed Domain Video Analysis:** extracts meaningful information from video content, stored or data streams.
  - **Unstructured data analysis:** analyses high volumes of unstructured data coming from the Internet in almost real time for a later process and analysis.
  - **Meta-data Pre-processing:** acts as a mediator between different metadata formats that need to inter-work
  - **Location Platform:** addresses issues related to Location of mobile devices
  - **Media-enhanced Query Broker:** provides an intelligent, abstracting interface for retrieval of data from the FI-WARE data management layer in addition to the publish/subscribe interface as another modality for accessing data.
  - **Semantic Annotation:** helpful for editors to categorize content in a meaningful manner.
  - **Semantic Application Support:** provides an infrastructure for metadata publishing, retrieving and subscription, and a set of tools for infrastructure and data management.
  - **Social Network Analysis:** analyses the social interactions of users to unveil their social relationships and social communities.
  - **Mobility Analysis:** transforms geo-located user activity information into a mobility profile of the user.
  - **Real-time recommendations:** analyses the behavior of a user through a service in order to make a recommendation of an item that such a user will most likely be interested in.
  - **Web behavior analysis for profiling:** is responsible for the derivation of profiles extracted from the activity of each individual user.

- **Opinion mining:** provides the analysis of textual sources to derive information about user's opinions.
- **Applications/Services Ecosystem and Delivery Framework** – the infrastructure to create, publish, manage and consume FI services across their life cycle, addressing all technical and business aspects. The list of GE belonging to this chapter are:
  - **USDL service descriptions:** the FI-Ware approach will allow comprehensive service descriptions by employing the Unified Service Description Language (USDL) in its registry and repository.
  - **Repository:** provides an API for reading, filtering, and aggregating service information from the repository as well as maintaining service descriptions.
  - **Registry:** acts as a universal directory of information used for the maintenance, administration, deployment and retrieval of services.
  - **Marketplace:** provides the needed functionality for people to offer and deal with services like goods and finally combine them to value added services.
  - **Business Models & Elements provisioning System:** provides flexible way to define the manner in which services and applications can be sold and delivered to the final customers.
  - **Revenue Settlement & Sharing System:** manages in a common way how to distribute the revenues produced by a user's charges for the application and services consumed.
  - **SLA Management:** provides the needed functionality for the management of Service Level Agreements.
  - **Composition editors:** helps the service provider to create application mashups and composed services.
  - **Composition execution engines:** for exposing and executing the composed services.
  - **Mediation:** provides interoperability solutions by means of mediation component acting as a *broker* between service consumers and providers.
  - **Multi-channel / Multi-device Access System:** manages user's contextual information in order to support service composition adaptation corresponding to user's preferences, profile and device.
- **Internet of Things (IoT) Services Enablement** – the bridge whereby FI services interface and leverage the ubiquity of heterogeneous, resource-constrained devices in the Internet of Things. The list of GE belonging to this chapter are:
  - **IoT Communications:** provides common and generic access to every kind of things regardless of any technological constraint on communications.
  - **IoT Resources Management:** proposes unified service and operational support management functions enabling the different IoT applications and end users to discover, utilize and activate small or large groups of IoT resources and manage their properties.
  - **IoT Data Handling:** collects large amount of IoT-related data, produced by a huge number of IoT resources almost in real-time
  - **IoT Process Automation:** proposes to Application/Services Providers generic capabilities enabling to use subscription and rules **templates that will ease** programming of automatic processes involving IoT resources.
- **Interface to Networks and Devices (I2ND)** – open interfaces to networks and devices, providing the connectivity needs of services delivered across the platform. The list of GE belonging to this chapter are:
  - **Connected Devices Interfacing (CDI):** provides to FI-WARE chapters and Use Case Projects applications and services with the means to detect and to optimally exploit capabilities and aspects about the status of devices.
  - **Cloud Edge:** allows federating connected devices and provides an API layer to allow the cloud-based applications to interface with devices through the implementation of interfaces and application program interfaces (APIs) towards device features.

- **Network Information and Control (NetIC):** provides to FI-WARE chapters as well as usage area applications and services with the means to optimally exploit the network capabilities, by means of the implementation of interfaces and APIs towards networking elements.
- **Service, Capability, Connectivity and Control (S3C):** is the manifestation of an adaption layer between the targeted network control layer for Fixed-Mobile-Convergence: Evolved Packet Core (EPC) and all possible applications and services.
- **Security** – the mechanisms which ensure that the delivery and usage of services is trustworthy and meets security and privacy requirements. The list of GE belonging to this chapter are:
  - **Identity Management:** provides authentication/access control and identity/attribute assertions as a service to relying parties.
  - **Privacy:** provides a set of functionality similar in scope to the Identity management generic enabler but enhanced using special privacy enhancing technologies.
  - **Data Handling:** provides a mechanism for controlling the usage of attributes and data based on the concept of ‘sticking’ a data usage policy to the data to which it applies.
  - **Context-based security and compliance:** supports additional security requirements requested by a specific subset of applications as a result of the application of very specific regulatory constraints.
  - **Optional Security Service Enabler:** used to customize the security service description within USDL-SEC when the security functionality is not covered by the specification.

### 5.3 Mapping Smart Building High-level Building Blocks to FI-WARE GEs

As explained and detailed in section 4.2.2, the high level functional architecture for smart buildings can be shown in different layers, each of them having different high-level building blocks. The functionality provided by these blocks can be partially covered by FI-WARE generic enablers as shown in the following table.

<b>Smart Building Architecture Layer</b>	<b>Smart Building Functional Architecture High-level Building Blocks</b>	<b>FI-WARE Chapter</b>	<b>GEs</b>	
Application layer (section 4.3)	Building energy management controller	Applications/Services Ecosystem and Delivery Framework	Multi-channel / Multi-device Access System	
		Cloud hosting	PaaS Management	
		Security	Identity Manager	
Service layer (section 4.4)	Entity virtualization services	Cloud hosting	PaaS Management	
Entity Abstraction layer (section 4.5)	Space proxies	Internet of Things (IoT) Services Enablement	IoT Resources Management	
	Legacy equipment proxies	Internet of Things (IoT) Services Enablement	Communications	
			Process Automation	
	Legacy Systems proxies	Internet of Things (IoT) Services Enablement	Interface to network devices (I2ND)	Connected Devices Interfacing (CDI)
			IoT Resources Management	
	Interface to network devices (I2ND)	Connected Devices Interfacing (CDI)		
Device layer (section 4.6)	Building connected devices	Internet of Things (IoT) Services	Communications	

		Enablement	Resources Management
		Interface to network devices (I2ND)	Connected Devices Interfacing (CDI)
	Building Sensors / Actuators	Internet of Things (IoT) Services Enablement	IoT Communications
		Interface to network devices (I2ND)	Connected Devices Interfacing (CDI)

Table 4: Mapping of FI-WARE GEs to functional building blocks of the FINSENY architecture

In a more graphical way, following the structure shown in section 4.2.2, the next figure shows this mapping.

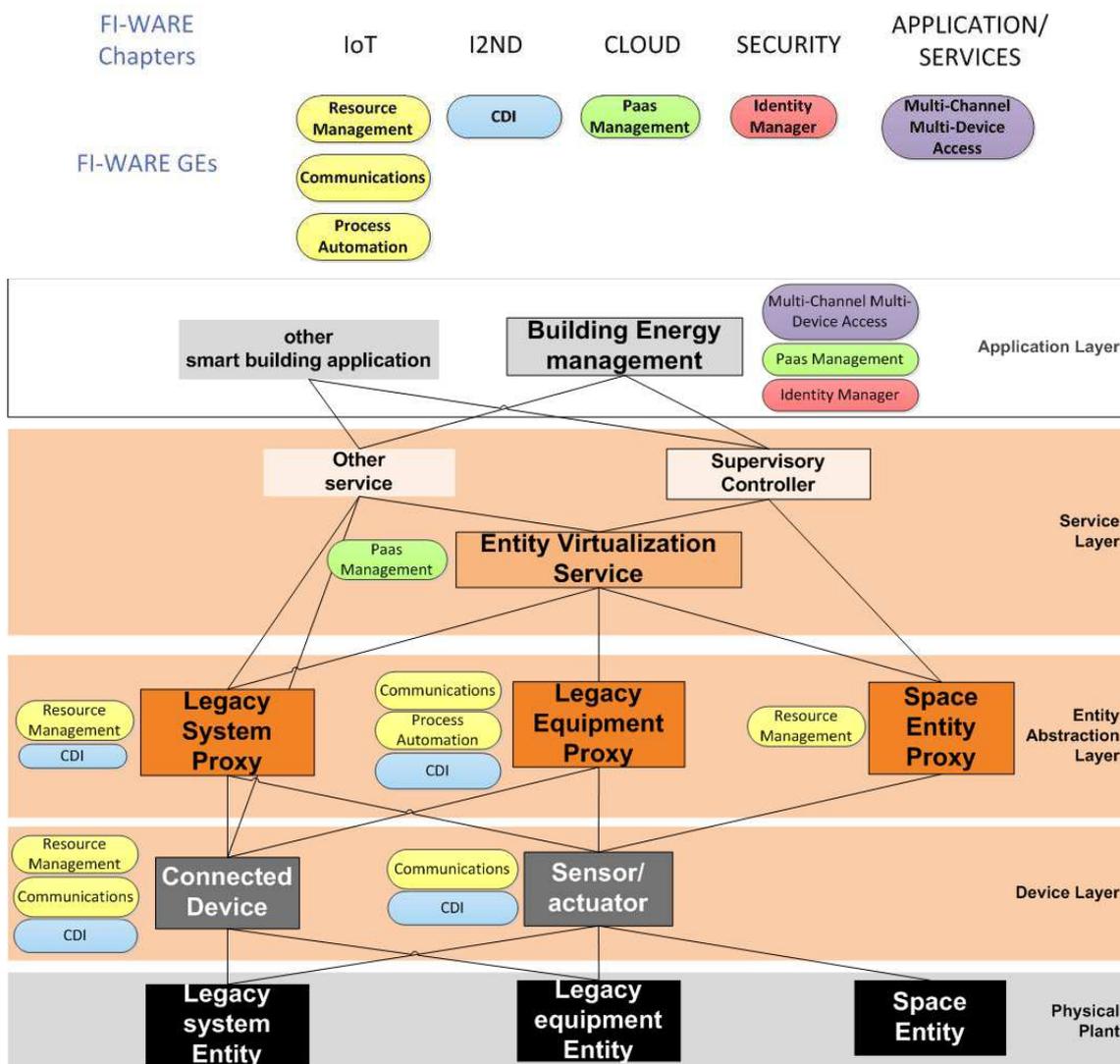


Figure 61: Mapping of FI-WARE GEs to functional building blocks of the FINSENY architecture

An overview justification of the mapping is the following:

- Application layer:
  - Building energy management high-level block:
    - Multi-channel / multi-device access: for the connection between the BEMS and the controlled appliances belonging to a Home Area Network (HAN)
    - PaaS management: to facilitate the management of applications, most of which will be hosted remotely, without worrying about the underlying infrastructure of virtual resources required for the execution of the application components.
    - Identity management: providing the mechanisms for authenticating the entities and other accessing the applications
- Service layer:
  - Entity Virtualization Service:
    - PaaS management: it provides facilities to manage applications and software modules (updates, installation, start/stop, versioning and maintenance), facilitate the deployment of entities without worrying about the underlying infrastructure. These services may be hosted locally or remotely
- Entity abstraction layer:
  - Legacy System Proxy:
    - IoT Resource management: it provides management of functions for discovering, utilize and activate energy related building legacy systems. So, BEMS are able to provide functions for dynamic registering and unregistering of appliances.
    - Connected Devices Interfacing (CDI): it provides the means to detect and to optimally exploit capabilities and aspects about the status of devices, so it could be used to discover new power-related legacy systems.
  - Legacy Equipment Proxy:
    - Communications: provides common and generic access to every kind of things regardless of any technological constraint on communications between building devices.
    - Process automation: proposes to Services Providers generic capabilities enabling to use subscription and rules templates that will ease programming of automatic processes involving building devices.
    - Connected Devices Interfacing (CDI): it provides the means to detect and to optimally exploit capabilities and aspects about the status of devices, so it could be used to discover new power-related appliances.
  - Space Entity Proxy:
    - IoT Resource management: it provides management of functions to discover, utilize and activate space entities such as rooms. So, BEMS are able to use functions for dynamic update of entities.
- Device layer:
  - Connected device:
    - IoT Resource management: it provides management of functions for discovering, utilize and activate connected devices. So, BEMS are able to provide functions for dynamic registering and unregistering these devices.
    - Communications: provides common and generic access to every kind of things regardless of any technological constraint on communications between connected devices.
    - Connected Devices Interfacing (CDI): it provides the means to detect and to optimally exploit capabilities and aspects about the status of devices, so it could be used to discover new power-related connected devices.
  - Sensor and actuator:
    - Communications: provides common and generic access to every kind of sensor and actuators regardless of any technological constraint on communications between them.

- Connected Devices Interfacing (CDI): it provides the means to detect and to optimally exploit capabilities and aspects about the status of sensors and actuators, so it could be used to discover new ones.

### 5.3.1 Covering Networking Requirements

In terms of Networking a BEMS has the following requirements:

- Networking within the Home Access Network (HAN)
- Networking with Service Cloud
- Networking with Overlay Grid Entities (MicroGrid/DSO)

For covering networking/communications requirements, FI-WARE’s “Interface to Networks and Devices (I2ND)” chapter specifies a Connected Devices Interfacing (CDI) GE as described in [4].

Overlaying Smart Grid entities such as Microgrid and DSO will make use Real Time Data Buses that will rely on a “Middleware for efficient and QoS/Security-aware invocation of services and exchange of messages” specified in FI-WARE’s 1<sup>st</sup> Open Call. BEMS will connect to a bus as such to exchange time critical information.

#### 5.3.1.1 IoT Service Stack

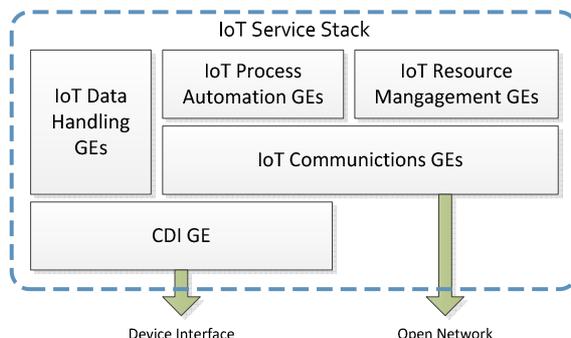


Figure 62: IoT Node Service Stack

The IoT Service Stack implemented in a Smart or a Gateway device may integrate GEs from the following technical subsections:

- **IoT Services Enablement / IoT Communications**

IoT communications provides common and generic access to devices (meters, sensors, DERs, appliances) regardless of any technological constraint on communications, typically integrating several protocols and manage discontinuity of connectivity for nomadic devices. It will allow the Smart Building framework to gain homogeneous access to dedicated things and devices and to be able to manage real time issues such as QoS etc.

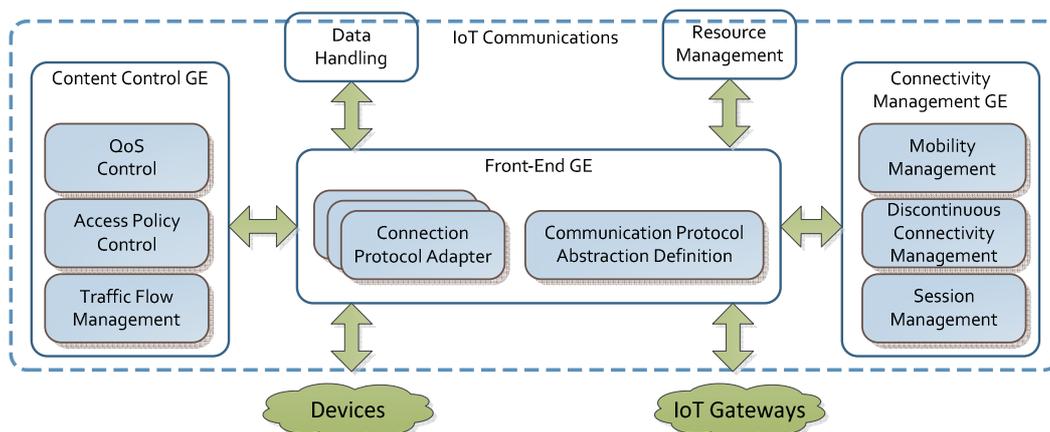


Figure 63: IoT Communications GEs

• **IoT Services Enablement / IoT Resource Management**

IoT Resources Management proposes unified service and operational support management functions enabling the different IoT applications and end users to discover, utilize and activate small or large groups of IoT resources and manage their properties. In doing so, the IoT Service Enablement will focus on global identification and information model schemes for IoT resources, providing a resolution infrastructure to link them with relevant things and developing a common remote management tool for configuring, operating and maintaining the IoT resources on a large scale and with minimum human intervention.

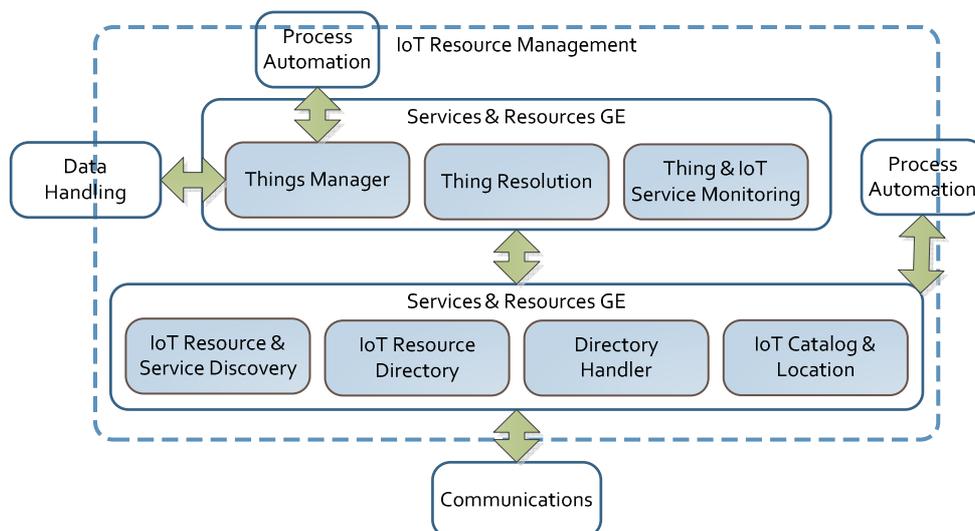


Figure 64: IoT Resource Management GEs

• **IoT Services Enablement / IoT Process Automation**

Process automation is dealing with low level processes describing the interactions of the IoT resources hosted by devices and IoT gateways (micro business rules), providing modeling constructs to describe these interactions, templates to capture the events occurring at this level of interaction and knowledge accumulation from the observation of interaction patterns occurring between the devices and IoT gateways. More specifically, the following features were identified:

- IoT knowledge management: increasing the intelligence of IoT Services capabilities over a long period of time, including
  - handler of application domain ontologies,

- knowledge base collector
- a design reasoner to execute classification of assertions and other semantic queries, target a collaborative framework between micro business rules processing engines
- Support to IoT-aware Business Process Management: enabling the programming of Business Processes where part of the process is able to run near or adjacent to IoT resources and gateways. This may imply defining means supporting important IoT characteristics directly into business processes notations.

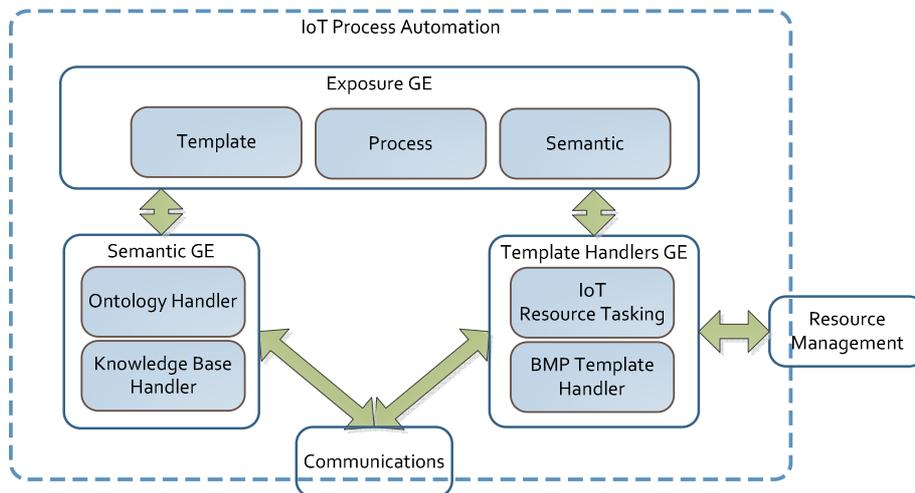


Figure 65: IoT Process Automation GEs

• **IoT Services Enablement / IoT Data Handling**

IoT Data Handling will allow the Smart Building Application and Control function blocks Providers to collect large amount of Device-related data, produced Smart Building devices in Real-Time.

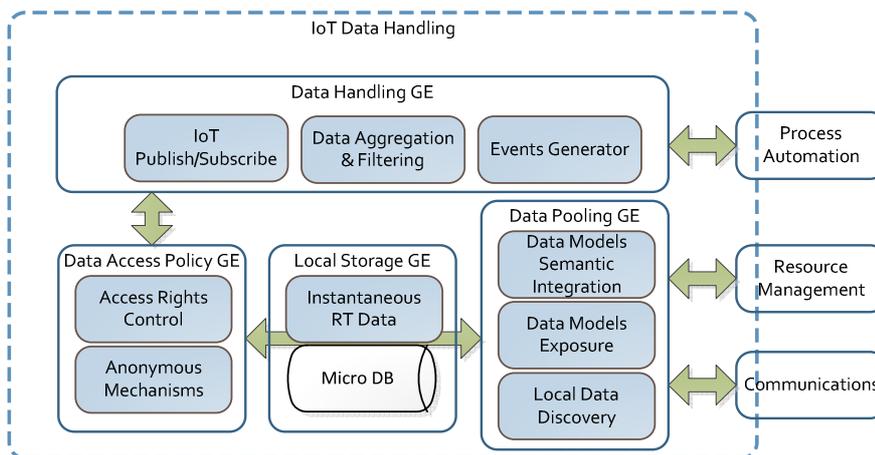


Figure 66: IoT Data Handling GEs

• **Data/Context Management / Connected Devices Interfacing**

The CDI GE is in charge of addressing a broad set of connected devices, not only the mobile ones, each adopting specific technology solutions in terms of hardware, software, middleware and runtime platforms, in particular concerning the development of applications.

A first challenge to be faced by the CDI GE is to provide homogeneous interfaces for application development. It is recognized that the extreme fragmentation of platforms adopted for connected devices, including a variety of different OSes and programming languages, is introducing several troubles to develop once for all the application and make it run on all such devices. As introduced in Section 3.1, different programming paradigms can be considered, which are exemplified in Figure 68. One step towards solution to fragmentation is represented by the adoption of middleware technologies (Java is one of the most relevant in this context), although not equally well supported by the majority of connected device platforms. Other emerging solutions rely on web based technologies available on most terminals. This trend seems to favour the development of applications which can run on web browsers or runtime engines.

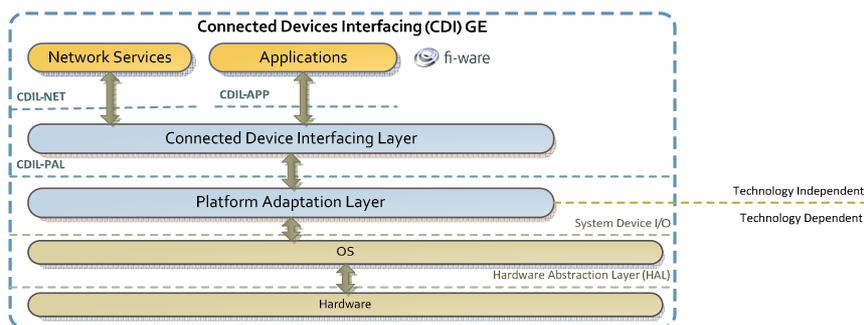


Figure 67: CDI GE

**5.3.1.2 FI-WARE mapped design for Device Layer components**

In Smart Building Architecture device layer components implement function aggregators implementing purpose built or network local control functions which expose to the overlay controlling entity aggregated function or notification digests.

Using FI-WARE design patterns these aggregators will build on IoT Gateway modules which will control clusters of connected devices, sensors and actuators and legacy equipment, identified in FI-WARE terms as IoT Resources

A device will have either to accommodate a FI-WARE IoT-Resource as specified by Connected Device Interfacing (CDI) GE and IoT Service Stack and a or connect to an IoT Gateway device which instantiates a device specific IoT Resource which incorporates a CDI GE to the interface connecting the device (e.g. RS-232, USB, etc.)

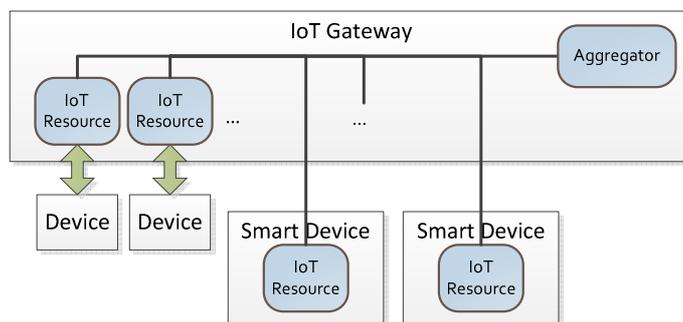


Figure 68: IoT Gateway aggregating devices functionality from devices with/and without computing capabilities

IoT Gateways can build on Device Front-End GE which is presented in the next paragraph.

Full device layer IoT Management and Supervisory functionality specified in the Entity Abstraction Layer is integrated in the IoT Backend which implements overlaid IoT Resource

Management and Security functions, while interfaces with purpose built, Smart Building control and supervisory services further discussed in §5.3.4

### 5.3.1.3 FI-WARE - Device Front-end GE

FI-WARE project delivers a novel service infrastructure, building upon elements, called GE (Generic Enablers) which offer reusable and commonly shared functions making it easier to develop Future Internet Applications in multiple sectors. This infrastructure will bring significant and quantifiable improvements in the performance, reliability and production costs linked to Internet Applications – building a true foundation for the Future Internet. In the Internet of Things (IoT) the power of combining ubiquitous networking with embedded systems, RFID, sensors and actuators makes the physical world itself a relevant part of any information system.

In the FI-WARE project, the interface with different sensors & actuators is handled by the Device Front-end GE. This GE deals with the incoming/outgoing traffic from/to Devices and IoT Gateways. It comprises a number of Connection Protocol Adapters and a component dealing with the Communication Protocol Abstraction Definition. For communication with different devices, the idea of this approach is to use “templates” to translate between protocols.

Each of the Connection Protocol Adapters is capable of handling one of the protocols that are accepted in FI-WARE, translating it into a unique internal language, which normalizes the different communication protocols within the platform.

Inside the Front-end GE, the definition of this “internal language” is contained in the “*Communication Protocol Abstraction Definition*” that provides a number of “templates” that can be used to translate the protocol (e.g. this translation can be achieved by creating an object, i.e. a “token”, specific for each incoming message type, containing all the data contained in the incoming message, that can be “injected” into the platform; likewise an outgoing token can be translated into a protocol message that will be sent by the platform to the specific Device or IoT Gateway).

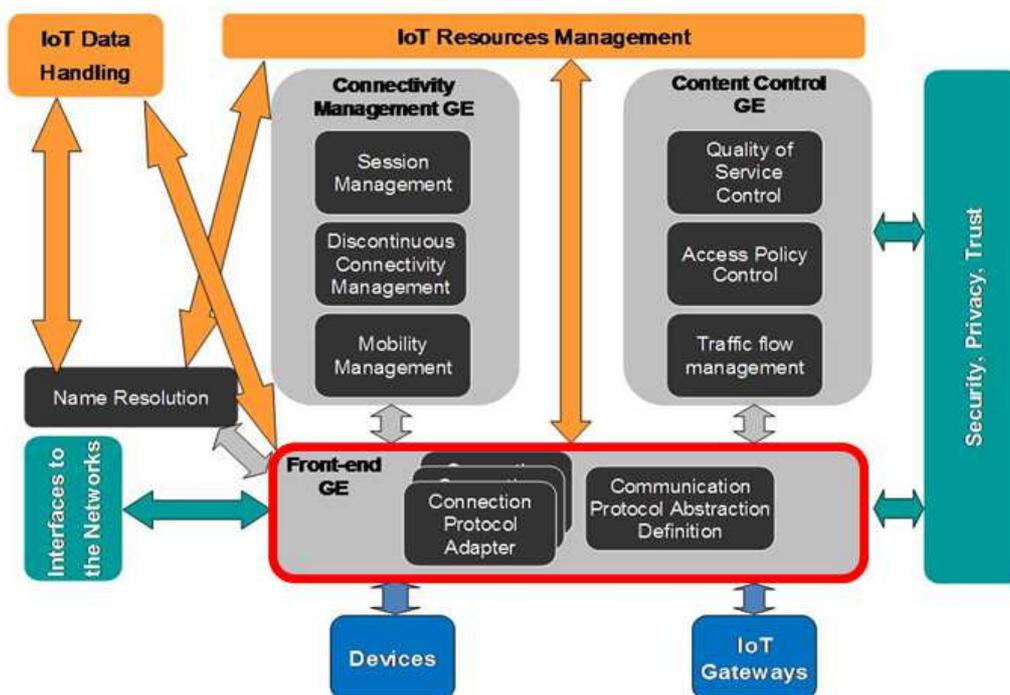


Figure 69: Architecture of IoT Communications (framed in red the Front-end GE)

The second component shown in Figure 69, inside the Front-end GE, is the “*Connection Protocol Adapter (CPA)*”. The CPA is capable of handling one of the protocols that are accepted in FI-WARE, translating it into a unique internal language, which normalizes the different communication protocols within the platform.

The Front-end GE relies on Security, Privacy & Trust Generic Enablers in order to perform the necessary management of security aspects. As an example, the Front-end GE relies on Security Generic Enablers to decrypt and encrypt the traffic incoming into and outgoing from Devices and IoT Gateways (e.g. managing of the asymmetric cryptography based on public and private keys). It also relies on Security Generic Enablers coping with management of access rights as it will be described later. Additionally, it relies on the IoT Data handling for storing and retrieving the templates from the Communication Protocol Abstraction Definition.

Moreover the Front-end GE exploits also the functions of the Name Resolution GE. This is a GE that can be shared between IoT Communication and IoT Resources Management. For the IoT Communication it translates the physical addresses received by the network into the addresses used internally within the platform and gives support to IoT Resources Management also for other address related functions.

About the sensors and actuators devices, in FI-WARE project they are included in Interface to Networks and Devices (I2ND) Architecture. The I2ND architecture covers four Generic Enablers (GEs): CDI (Connected Device Interface), CE (Cloud Edge), NETIC (NETwork Information and Control) and S3C (Service Capability, Connectivity and Control).

All these four GEs have interfaces and APIs according to their underlying technologies.

In particular, the CDI (Connected Device Interface) have features such as remote access from a control environment, exposure of own functionality (device status, sensors, actuator, etc).

### 5.3.2 Event & Data Processing

#### *Complex Event Processing*

**Complex Event Processing GE (CEP)** (see Figure 70) is specified in FI-WARE’s “Data/Context Management” chapter. It analyses event data in real-time, generates immediate insight and enables instant response to changing conditions. Some functional requirements this technology addresses include event-based routing, observation, monitoring and event correlation. The technology and implementations of CEP provide means to expressively and flexibly define and maintain the event processing logic of the application, and in runtime it is designed to meet all the functional and nonfunctional requirements without taking a toll on the application performance.

Entities connected to CEP (application entities or some other GEs like the Publish/Subscribe Broker GE) can play two different roles: the role of Event Producer or the role of Event Consumers. Note that nothing precludes that a given entity plays both roles. Event Producers are the source of events for event processing. Following are some examples of event producers:

- External applications reporting events on user activities such as "user placed of new order" and on operation activities such as "delivery has been shipped".
- Sensors reporting on a new measurement. Such a sensor generated event can be consumed directly by the CEP GE. Another alternative is that the sensor event is gathered and processed through the IoT GEs, which publish context events to the Publish/Subscribe GE, having the CEP be a context consumer of the Publish/Subscribe GE.

They can provide events in two modes:

- "Push" mode: The Event Producers push events into CEP by means of invoking a standard operation CEP exports.
- "Pull" mode: The Event Producer exports a standard operation that CEP can invoke to retrieve events.

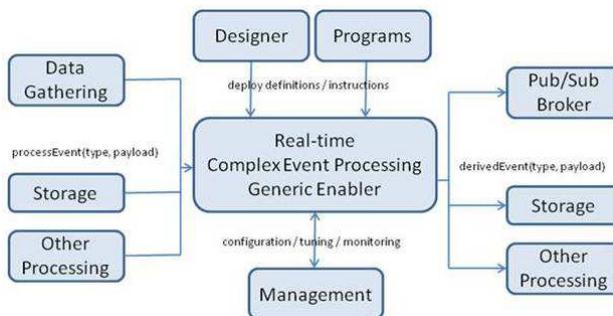


Figure 70: Complex Event Processing (CEP) GE

CEP implements event processing functions based on the design and execution of Event Processing Networks (EPN), as shown in Figure 71. It can accommodate multiple Input (Event Producer) and Output (Event Consumer) interfaces and multiple event processing entities (Agents).

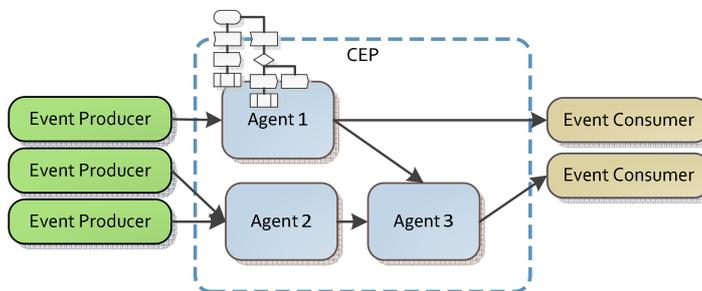


Figure 71: EPN Definition Example

An EPN definition will be created in design-time, and a compiled or interpreted instance will be uploaded in the CEP Run-Time engine for Run-Time execution on an EPN execution engine. Agent modules will handle the interfacing of the CEP engine with other software components of the architecture, implementing event and data translation, event queuing etc.

For EPN design and event specification there have been proposed several specifications such as DOLCE (Description Language for Complex Events).

**Big Data Analysis**

*Big Data Analysis GE* (see Figure 72) is a GE designed to process huge amounts of data either stored or continuous. In the first case where the data are stored and latency is not a highly relevant parameter, analysis is carried out in batch mode, whereas in the case of a continuous stream of data, analysis is carried out on-the-fly.

The technology behind builds on Google’s MapReduce framework for processing parallelizable problems across huge datasets using a large number of computers (nodes), collectively referred to as a cluster (if all nodes are on the same local network and use similar hardware) or a grid (if the nodes are shared across geographically and administratively distributed systems, and use more heterogeneous hardware). In this scenario there are two steps: First a "Map" step where a master node takes the input, divides it into smaller sub-problems, and distributes them to worker nodes. A worker node may do this again in turn, leading to a multi-level tree structure. The worker node processes the smaller problem, and passes the answer back to its master node. Then a "Reduce" step where the master node collects the answers to all the sub-problems and combines them in some way to form the output – the answer to the problem it was originally trying to solve.

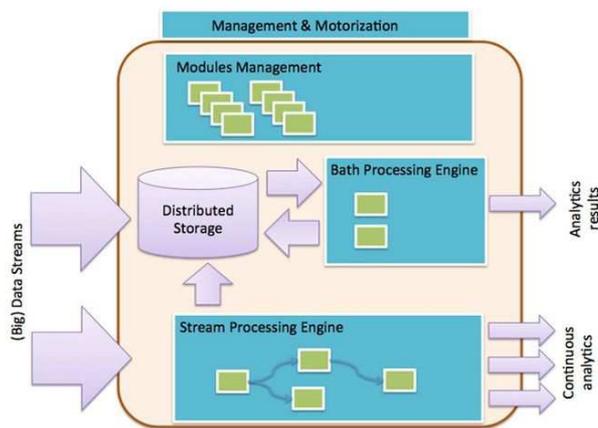


Figure 72: Big Data Analysis GE

Some of the key properties of the GE are the following:

- Requires a data/processing-intensive execution platform
- Best suited for data where a short/medium term history processing is needed for the analysis

### 5.3.3 Interface with Marketplace and Grid

The interface with marketplace and grid function handles the information flows with the marketplace and the aggregator / Microgrid or DSO. These information flows convey signals (price but also other parameters) that can be used to implement demand side management and peak shaving measures.

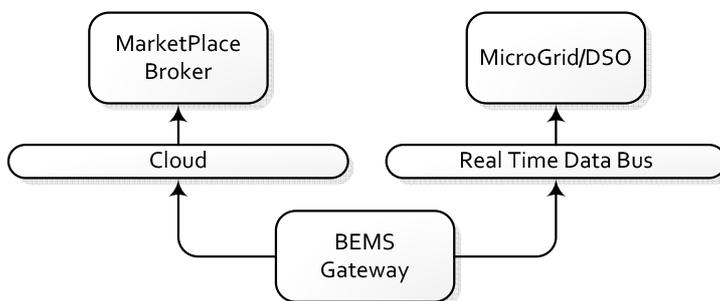


Figure 73: Interface with Marketplace and Grid

A Complex Event Processing GE can be the container of business intelligence agents that can orchestrate the processing of real time events and data acquisition and decide upon actions and output events.

A Complex Event Processing GE comprises of Agents and Adaptors.

Agents are State Machine Instances, specified and configured at an installation phase that can process input events. They implement the core Business Intelligence processing, using input/output events either from other from other agents or adaptors.

Adaptors are mediation modules that implement data conversions and transaction adaptation and negotiation functions that ensure the delivery of information messages in the appropriate format.

In the case of the Marketplace & Grid interface we would need such a CEP entity to perform complex event orchestration and communicate with internal components, overlay entities and HAN appliances/devices in order to implement User and Grid specific policies which correlate on one hand dynamic tariff info, consumption restrictions (e.g. MicroGrid Islanding mode), user

profile rules and on the other hand status, operations and tasks related to BEMS appliances and local DERs.

The following image describes a potential deployment scenario of a Marketplace & Grid specific CEP.

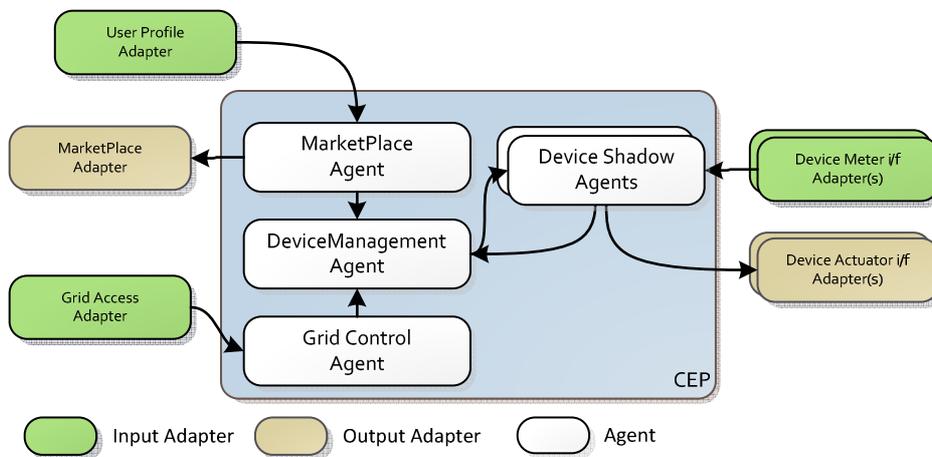


Figure 74: Example of a Marketplace & Grid specific CEP deployment scenario.

In this scenario the Marketplace adapters can collect information regarding tariffs from a Marketplace Broker specified in FI-WARE’s “Applications/Services Ecosystem and Delivery Framework” chapter as “Marketplace GE”, implementing client side interfaces.

The process maybe controlled via a CEP’s Marketplace Agent that can digest configuration from a user profile then send events to a Device Management agent to reschedule automated tasks and proceed to device specific actuation operations.

Identity Management negotiation may be performed by FI-WARE’s “Security” chapter “Identity Manager GE”.

Device Specific Agents and Adaptors can be provided in a Domain Specific Repository and downloaded dynamically in a deployment specific dynamic configuration upon system installation and as part of the device discovery procedure.

**5.3.4 Building Abstraction**

The “Building Abstraction Layer” (BAL) as proposed is an analogue of Hardware Abstraction Layers used by operating systems: it hides the specifics of the building hardware beneath a set of generic models and interfaces, acting as a generic informational interface to the building as a physical system. Therefore it will rely on IoT/Device aggregation and complex event processing functions as described in 5.3.1.2 & 5.3.2

**5.3.4.1 Composite abstraction definitions for data and control**

Applications can use a mediator between different metadata formats in the power-related readings to be composed and inter-work. So, it is possible to get composed reading data from a set of sensors or translating a single higher order function into a set of independent directives to individual actuators.

Device models and the corresponding CEP agents and adapters can be provided from a centralized model repository and BEMS specific HAN configuration can be created dynamically upon system configuration using wizards and discovery tools in a domain specific user interface.

Building Block	Smart Building Functional Architecture High-level Building Blocks	Domain-specific or FI-WARE Chapter	FI-WARE Generic Enabler	Comment/ Mapping Justification
Sensor & Actuator I/F (§4.6)	Collect, accumulate, display and propagate readings	IoT	IoT Communications	It provides access to every power-related readings regardless of any technological constraint on communications
			IoT Resources Management	It provides management of the functions used to collect, accumulate, display and propagate power-related readings.
			IoT Data Handling	Collects power-related readings from the different sensors.
		Cloud Hosting	Object Storage	For the storage of power-related reading and related information for the function
		Interface to Networks and Devices (I2ND)	Connected Devices Interfacing (CDI)	Connects devices from the HAN to the backend system
			Network Information and Control (NetIC)	Provides APIs for exploiting network capabilities in order to facilitate the collection of power-related reading
		Data/Context management	Complex Event Processing	It provides accumulation and real-time analysis of the power-related readings over defined time intervals.
		Security	Identity Manager	Providing the mechanisms for authenticating the user access to power-related values displayed

Building Block	Smart Building Functional Architecture High-level Building Blocks	Domain-specific or FI-WARE Chapter	FI-WARE Generic Enabler	Comment/ Mapping Justification
			Privacy	Provides privacy functionality to the power-related values collected from the sensors.
Building Abstraction (§4.4.6)	Define composite abstractions for data and control	IoT	IoT Data Handling	Applications can use a mediator between different metadata formats in the power-related readings to be composed and inter-work. So, it is possible to get composed reading data from a set of sensors or translating a single higher order function into a set of independent directives to individual actuators.
		Domain specific		Virtualization of building entities: One-to-many and many to one mappings
		Domain specific		Repository of models for physical entities
		Domain specific		Security at the entity Level.
	Discover appliances	IoT	IoT Resources Management	It provides management of functions for discovering, utilize and activate energy related building entities. So, BEMS are able to provide functions for dynamic registering and unregistering of appliances.
		Interface to Networks and Devices (I2ND)	Connected Devices Interfacing (CDI)	It provides the means to detect and to optimally exploit capabilities and aspects about the status of devices, so it could be used to discover new power-related appliances.

Building Block	Smart Building Functional Architecture High-level Building Blocks	Domain-specific or FI-WARE Chapter	FI-WARE Generic Enabler	Comment/ Mapping Justification
		Security	Identity Manager	Providing the mechanisms for authentication
Shared Services (§4.4)	Access historical information	Data/Context management	Big Data Analysis	Processes huge amounts of power-related stored data to be displayed as historical information to final users.
		Cloud Hosting	Object Storage	For the storage of power-related reading and related information for historical reports
	Remote updates, installation and maintenance	Cloud Hosting	PaaS Management	It provides facilities to manage applications and software modules (updates, installation, start/stop, versioning and maintenance), facilitate the deployment without worrying about the underlying infrastructure.
		Security	Privacy	Provides cryptographic signatures to validate the integrity and to authenticate the entity that implements a software module.
Application (§4.3)	Monitor and control assets	Interface to Networks and Devices (I2ND)	Connected Devices Interfacing (CDI)	For the connection between the BEMS and the appliances belonging to a HAN
		Security	Identity Manager	Providing the mechanisms for authenticating the BEMS accessing the appliances to control
		Domain specific		Non-application-specific supervisory control. Functions on the BEMS for controlling appliances

Building Block	Smart Building Functional Architecture High-level Building Blocks	Domain-specific or FI-WARE Chapter	FI-WARE Generic Enabler	Comment/ Mapping Justification
	Interface with marketplace and grid	<u>Applications/Services Ecosystem and Delivery Framework</u>	Marketplace	It provides the needed functionality for people to offer and deal with services like goods and finally combine them to value added services. This function handles the information flows with the marketplace and the aggregator / Microgrid or DSO.
		Security	Privacy	Provides cryptographic signatures to validate the integrity and to authenticate the entity interfacing the marketplace
			Identity Manager	Providing the mechanisms for authenticating the entities accessing the marketplace
		Domain specific		A specific enabler is required to interface applications with the specific smart energy marketplace
	Optimize, schedule and react	Interface to Networks and Devices (I2ND)	Connected Devices Interfacing (CDI)	For the connection between the BEMS and the controlled appliances belonging to a HAN
		Security	Identity Manager	Providing the mechanisms for authenticating the BEMS accessing the appliances to control
		Domain specific		Optimization engine: it complements the "interface with marketplace and grid" by allowing the BEMS to react based on the relevant economic information it receives from the marketplace or the grid administrator
Monitor and control non-	Interface to Networks	Connected Devices	For the connection between the BEMS and the	

<b>Building Block</b>	<b>Smart Building Functional Architecture High-level Building Blocks</b>	<b>Domain-specific or FI-WARE Chapter</b>	<b>FI-WARE Generic Enabler</b>	<b>Comment/ Mapping Justification</b>
	electrical automations	and Devices (I2ND)	Interfacing (CDI)	appliances belonging to a HAN
		Security	Identity Manager	Providing the mechanisms for authenticating the BEMS accessing the appliances to control
		Domain specific		Non-application-specific supervisory control
	Interface with user	Domain specific		Non-application-specific UI toolkit

## 6 Information models

### 6.1 Ontology of building entities

The data being handled in present-day IoT/M2M services lack both the semantics and the level of abstraction required to treat them as a pool of common data available in a given environment such as buildings and to share them between different applications, without these applications needing to know beforehand the specifics of these data. The usual way to provide this higher level information models is by attaching semantics to the data itself (units or basic metadata making it possible to interpret the data unambiguously) or to the context (physical low-level context information such as location, time, which has been extensively covered in context management frameworks).

Beyond these very generic solutions, a sound and relevant foundation for high-level information models should acknowledge and understand that M2M applications, and in our case those M2M applications deployed in the building energy management and building automation domains, are not interested in sensors and actuators themselves, but in what is being sensed by the sensors, or acted upon by actuators. The relevant level of abstraction for M2M data pooling should thus not be confined to individual sensors and actuators readings, even if they are enriched with proper metadata. It should rise to the level of physical entities that are being sensed by sensors and acted upon by actuators. In the building environment, these entities may be appliances, people, rooms of the building, etc. These entities are generic, intrinsic to the building environment and not tied to a specific IoT application such as energy management. They can be legacy appliances or completely passive “things” or spaces and should not need to be directly connected through a direct network interface, or even be identified through a universal identification scheme such as RFID/EPC global, provided they can be sensed by sensors that are supposed to be deployed in this environment.

The proposal we made above for a building abstraction layer amounts to define and solidify into the chosen architecture framework such a “thing” information model corresponding to an intermediate layer of data management matched to the physical entities of the building environment. Relevant models of these entities are drawn from an ontology defining the primitive entities of this environment.

Examples drawn from such ontology are pictured in Figure 75 below, with two branches corresponding to rooms and appliances.

This ontology captures generic knowledge about the devices, appliances and rooms of the building domain through a multi-criteria hierarchical categorization and the definition of generic hybrid finite state models for each of the target subsystems. Initiating and configuring the Building Abstraction Layer proxy/driver for a given entity actually amounts to identifying, loading and iteratively adapting the most appropriate subsystem model from this model repository.

This ontology subsumes several relevant categorizations of appliances or rooms. As illustrated in Figure 75 it can be structured as a directed acyclic graph that makes it possible to follow a path from the most generic parent models (closer to the root of the graph) to the more specific models (closer to the leaves). At the root is the main class *building physical entity*, which in the scope of this document specializes into two descendant classes: appliance and room. Appliances and rooms can in turn be specialized and classified according to several different criteria that are either intrinsic to their main usage, or relative to the application (in our case energy management and energy efficiency). Examples of these criteria are illustrated below, each of them corresponding to intermediate nodes in the graph

Models are associated not only with the terminal nodes of the graph corresponding to the most specific categories, but also to the intermediate nodes. This full hierarchy of models provides a mechanism to identify a subsystem in an incremental way on the basis of observation data, starting from the most generic model if observed sensor data is inconclusive, and refining this first match to more specific models down the graph when further observation data becomes available.

As explained in the “building abstraction layer section, these entities can be self-configured into the system in a way similar to zeroconf discovery mechanisms used for configuration of network entities. Once configured, they are contextually identified and represented by an informational proxy that implements an executable version of their informational model as a self-contained subsystem and serves as an intermediary towards applications, providing them an interface for control and monitoring of this entity with the primitives defined in the corresponding model

Both devices and rooms can be described with hybrid finite-state discrete-time models, where state information is possibly complemented with continuous-valued attributes. These states and the relevant attributes are then stored as the state of the shadow ICT subsystem. These hybrid models represent a tradeoff between expressivity and ease of identification. The full description of a physical system such as the target rooms and appliances would normally require a continuous state & continuous-time model, but automatic identification of the parameters of such models would be nearly impossible. Examples of such models are given below for a category of appliance (a generic washing machine) and a category of room (a living room modeled by its different states).

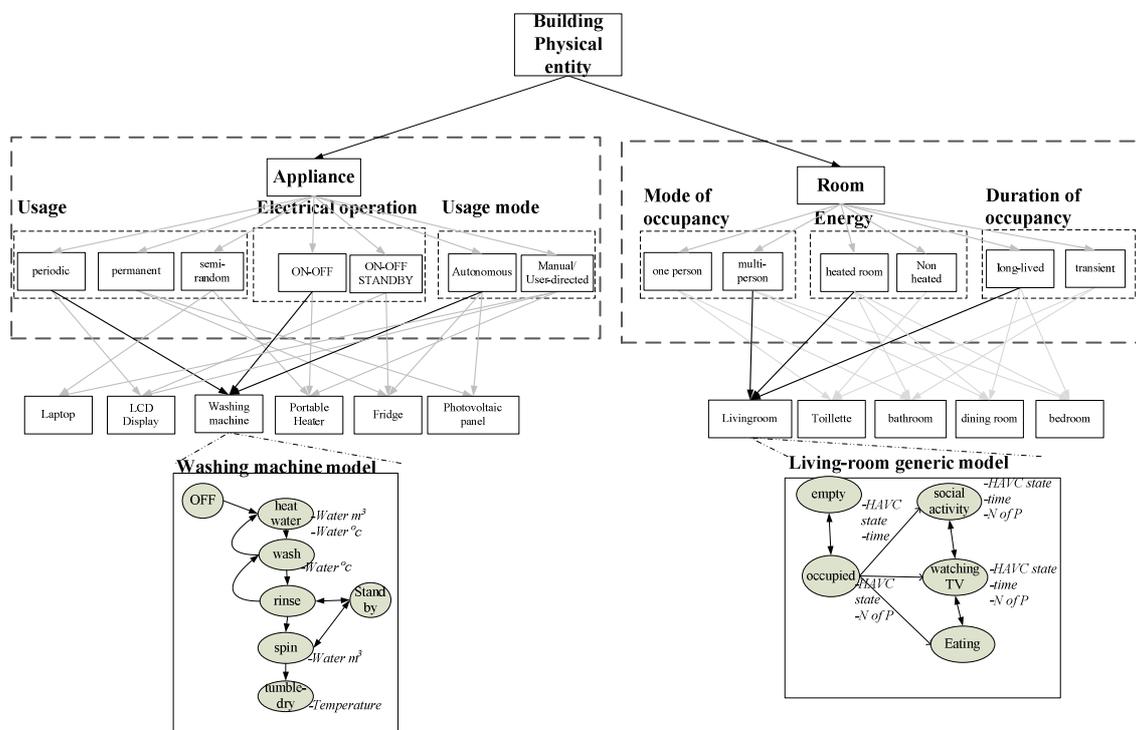


Figure 75 Example ontology of building entities to be represented in building abstraction layer

This “entity-centric” information model is fully aligned with options taken by the IoT enablement chapter of the FI-ware project, which does define such a layer above the “device” layer and associates generic enablers with this model.

### 6.2 Discrete-event models of individual building entities

We consider the target building entities to be full-fledged systems in a system-theoretic sense. Their state vector  $S(T)$ , a function of time  $T$ , can be defined in a system-theoretic sense as encapsulating the necessary and sufficient information to predict the future states and outputs of the system given its future inputs. This comprehensive definition has to be restricted for practical reasons to those dimensions of the state that are actually relevant for the target application. For an energy management application, the state of energy-relevant subsystems should comprehend their state as thermodynamic systems, which may comprise the amount of

energy they are actually consuming, storing and generating. For a smart environment automation application the mechanical state of these entities would be more relevant.

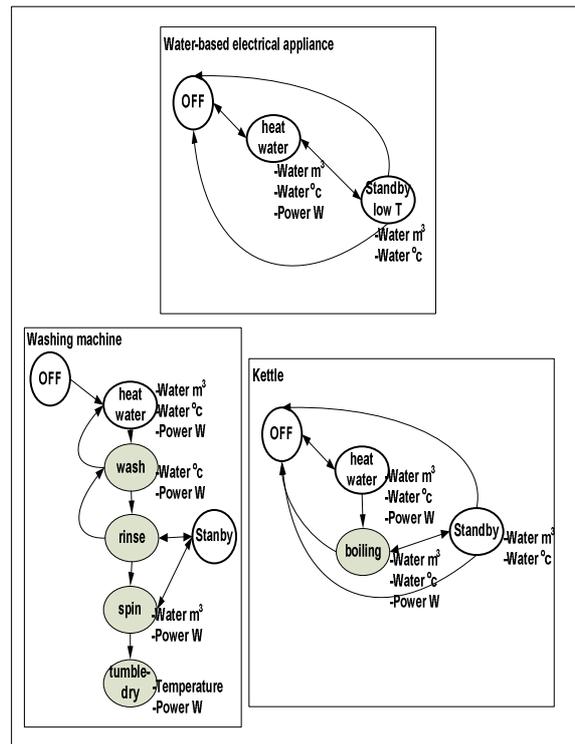


Figure 76 : Hybrid Discrete State model examples for home appliances

Being continuous as they normally are for a physical systems, these states may for simplicity reasons be lumped into discrete “state categories” (e.g. a mode of operation for an appliance), that could correspond to the states of a simpler discrete-state model. This finite state model can be made richer by complementing discrete state variables with continuous variables (attributes) capturing relevant properties related to the state (e.g., time spent in that mode, temperature, percentage filled, etc.). These digitized states together with the relevant attributes are then stored as the state of the shadow ICT subsystem.

Monitoring is concerned with the identification and tracking of these instantaneous states of the physical entity, on the basis of relevant sensor data. Controlling refers to change the state of the physical entity by using actuators.

For a generic washing machine and kettle, their discrete states as illustrated in Figure 76, the transition between two states is define the possibility of changing state, for each state, an/plural continuous variables can be detected by the associated sensing devices. With matching this model, an EIMC washing machine entity and kettle entity can be instanced.

Then for the upper layer, an entity group representation model can be found, as shown above the two models in Figure 76. This model is defined the discrete state of the intermediate node, mentioned above. It is for grouping the washing machine EIMC and kettle EIMC which have the “OFF”, “heat water” and “Standby” state. By matching this model, an EGC can be instanced in ICT system so that the system is capable of controlling both appliances as a water-based electrical appliance.

## 7 Communication layer

In the FINSENY D4.2 deliverable [2], SGAM methodology has been adopted to perform a top-down drill-down of a selected number of highly informative and relevant use cases from each domain.

The drill-down was not complete in that it only cursorily explores the lower Communication and Component SGAM layers; the aim was not to arrive at a complete specification of the architecture, but rather to propose a coarse grain view. Now, in this chapter the communication layers will be analyzed in more detail, proposing those communications that stand to adopt standardized solutions and for this reason are applicable to the Smart Grids pan-European environment.

As will be shown in the next sub-chapters, depending on whether the structure is a home, office, hotel, data center etc., different communications technologies are used.

Anyway, these communications fall into two main areas: wired and wireless communications, and, in some instances, wireless communications have some advantages over wired technologies, such as low-cost infrastructure and ease of connection to difficult or unreachable areas. However, the nature of the transmission path may cause the signal to attenuate.

On the other hand, wired solutions do not have interference problems and their functions are not dependent on batteries, as wireless solutions often do.

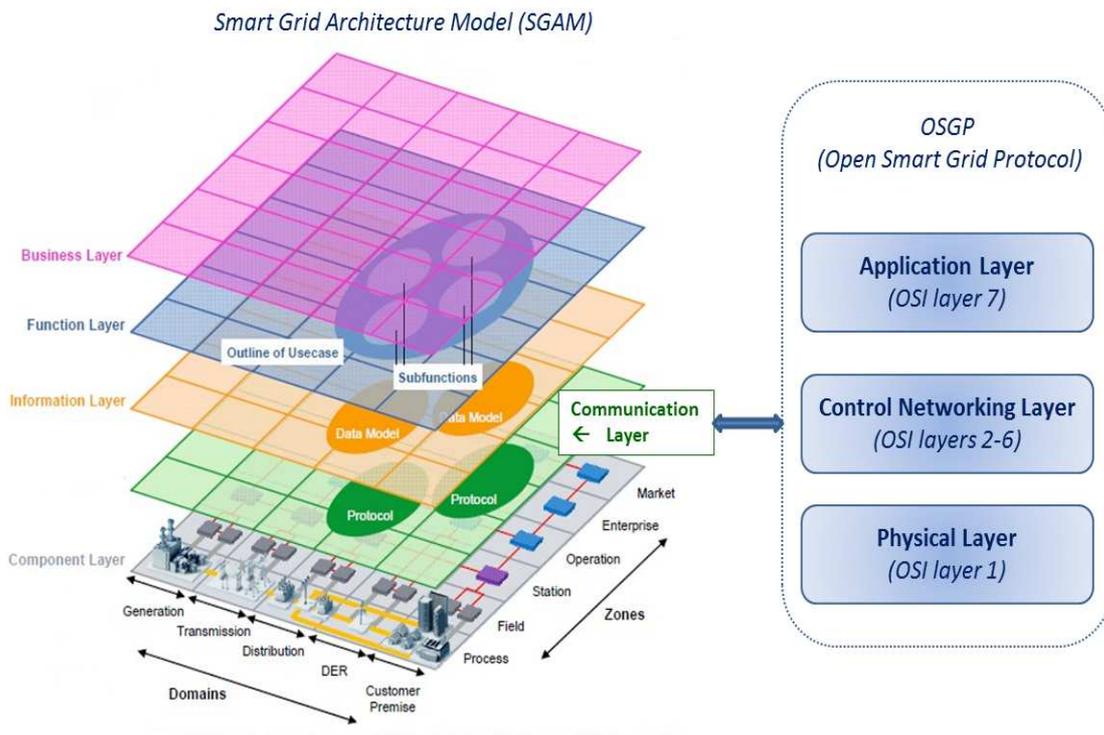


Figure 77: SGAM communication layer and OSGP/OSI protocols

The SGAM communication layers, that represent protocols and mechanism for the exchange of information between the Smart Grid components, has to be compatible with the worldwide standard stack models, such as ISO/OSI (Open Systems Interconnection) and OSGP (Open Smart Grid Protocol), as shown in **Figure 77**.

This is not so easy because, at worldwide level, there are a lot of standard communication involved in the smart building applications; in **Table 5** are summarized the most used. Some of

these communication protocols will be shown in the 7.3 sub-chapter, where the communications used inside the different kind of smart buildings are described.

<b>Communication protocols</b>	<b>OSI Layers</b>	<b>Maturity</b>	<b>Smart building applications</b>	<b>Data rate</b>
3GPP / UMTS / LTE	PHY, MAC, Network	High maturity and high penetration	Messaging; Voice/ Video streaming	9,6 Kbit/s – 100 Mbit/s
PLC	PHY, MAC	High; Several standards; numerous deployments	Smart metering Home automation	1 bit/s – 200 Mbit/s
IEEE 802.15.4	PHY, MAC	High; IEEE standardized; numerous manufacturers	Building automation Inventory/warehouse area; Location detection	20 Kbit/s – 1 Mbit/s
IPv6/6LowPAN	PHY, MAC	Medium	Personal area networks applications	256Kbit/s
ZigBee	PHY, MAC, Application	High; public specif.; Numerous OEMs;	Building automation; Low data rate telco applications; Smart energy	256Kbit/s
Z-Wave	PHY, MAC, Application	High; over 200 manufacturers	Building automation	40 bit/s
BACnet	All OSI layers	High; numerous installers worldwide;	Building automation	1 Mbps over twisted pair wiring (RS-485).
LonWorks	All OSI layers	High; ANSI standard; numerous devices	Building automation Smart metering	3,6 Kbit/s – 1,25 Mbit/s
Wireless M-Bus	PHY, MAC, Application	Low; Few manufacturers	Smart metering	16 – 66 Kbit/s
DigitalSTROM	PHY, MAC, Application	Low; limited to Germany and Switzerland	Building automation; Home energy consumption monitoring	50 – 100 bit/s
NFC / RFID	PHY, MAC	High; Several standards; numerous deployments;	Access control, electronic article surveillance, positioning	< 1 Kbit/s – 100 Kbit/s

Table 5: Communication protocols in smart building applications

In the next sub-chapter will be briefly exposed the OSI layers involved in OSGP stack, in order to make it easier to understand their roles in the communication layers along the rest of the chapter.

## 7.1 OSGP and ISO/OSI communication in the smart buildings

Smart buildings as a whole completely depend on communication infrastructure. Benefits of smart grid can be achieved by exchanging information on the bases of syntactic and semantic. A protected, consistent and economic power transmission is strongly linked with fast, well-organized and reliable communications infrastructure.

To achieve this milestone syntactic and semantic interoperability is the challenging task. In a communication environment every participant behave like a consumer or supplier of information, sometimes both; these participating parties may consist of subsystems having their internal communication. There are some protocols available specially designed for smart grid.

OSGP (Open Smart Grid Protocol) is a communication protocol used to exchange information with smart meters and smart grid devices. Before its deployment as an international European standard it was used to transform information very reliability and securely will millions of meters in de-facto open standard for smart metering.

OSGP is the most widely used communication standard for smart metering and smart grid applications and it is supported by multiple suppliers and software providers: it follows a modern, structured approach based on the OSI (Open Systems Interconnection) protocol model to meet the evolving challenges of the smart grid.

Here following are briefly summarized the OSI layers that OSGP includes in its three main shown in **Figure 77**.

Physical level (OSI 1): it coincides with the one OSI layer 1. Here the data is translated into transmittable signals and put on the network media (in wired or wireless mode) to travel across the network. It is therefore dealing with signalling issues including: analogue versus digital signalling, baseband versus broadband technology, asynchronous versus synchronous transmission and multiplexing.

Above the physical layer of the OSGP architecture is the Control Networking Layer that includes all the OSI 2-6 layers, and precisely: *Data Link Layer*, *Network Layer*, *Transport Layer*, *Session Layer* and *Presentation layer*. Here following a brief their description in order to highlight their functionality for the communication in various types of intelligent structure described in the next sub-chapters-

Data link layer (OSI 2): defines the format of data on the network: a network data frame, or packet, includes checksum, source and destination address, and data. The largest packet that can be sent through a data link layer defines the Maximum Transmission Unit (MTU). The data link layer handles the physical and logical connections to the packet's destination, using a network interface. A host connected to an Ethernet would have an Ethernet interface to handle connections to the outside world, and a loopback interface to send packets to it. The Data Link layer prepares data blocks usually referred to as frames for transmission, takes care of the synchronization of data transmission at both Source and Destination, and resolves issues related to accessing the medium shared by multiple users.

Network layer (OSI 3): is responsible for the delivery of data packets from the Source to Destination across the communication network. One of the main functions of the Network layer is to use packet switching techniques to switch data at each node. The Internet Protocol (IP) is the best example of a network layer implementation. IP is the only network protocol supported by all operating systems deployed in industry and open systems. Given the existing base of development and integration as well as the wave of devices that are IP-enabled, IP is the only viable choice of a network layer protocol.

Transport layer (OSI 4): it accepts data from the Session layer and segments the data for transport across the network. Generally, the Transport layer is responsible for making sure that the data is delivered error-free and in the proper sequence. Flow control, which is the process of managing the rate of data transmission between two nodes to prevent a fast sender from outrunning a slow receiver, generally occurs at the Transport layer. The Transport layer also provides the acknowledgement of the successful data transmission. The Transmission Control

Protocol (TCP), one of the major transport protocols, is typically used with the best-known network layer protocol, IP, and is referred to as TCP/IP.

Session layer (OSI 5): it establishes, manages, and terminates communication sessions. Communication sessions consist of service requests and service responses that occur between Applications located in different network devices. Communication sessions consist of requests and responses that occur between applications. Session-layer services are commonly used in application environments that make use of remote procedure calls (RPCs).

Presentation layer (OSI 6): it provides a variety of coding and conversion functions that are applied to Application layer data. These functions ensure that information sent from the Application layer of one system would be readable by the Application layer of another system. Some examples of Presentation layer coding and conversion schemes include common data representation formats, conversion of character representation formats, common data compression schemes and common data encryption schemes. Common data representation formats, or the use of standard image, sound and video formats, enable the interchange of Application data between different types of computer systems.

Application layer (OSI 7): it is at the top of the OSGP stack and coincides with the one OSI layer 7. The Application layer is the OSI layer closest to the end user, which means that both the OSI Application layer and the user interact directly with the Application. This layer interacts with Applications that implement a communication component. Such Applications fall outside the scope of the OSI model.

Application layer functions typically include identifying communication partners, determining resource availability, and synchronizing communication. When identifying communication partners, the Application layer determines the identity and availability of communication partners for an Application with data to be transmitted.

When determining resource availability, the Application layer must decide whether sufficient network resources for the requested communication are available. In synchronized communication, all communication between Applications requires cooperation that is also managed by the Application layer. There are dozens of different application layer protocols that enable various functions at this layer. Some of the most popular ones include HTTP, FTP, SMTP, DHCP, NFS, Telnet, SNMP, POP3, NNTP and IRC.

With reference to the OSGP layers above described, in the following sub-chapters the communication layers for different kind of structures are given.

## **7.2 Topological Segmentation of the communication protocols for Smart Buildings**

To ensure stability on the electricity grid, electricity supply and demand must remain in balance in real time. Traditionally utilities have called upon peaking power plants to increase power generation to meet rising demand. Demand-side management (DSM), also known as demand response (DR), works from the other side of the equation – instead of adding more generation to the system, it pays energy users to reduce consumption. Utilities pay for demand-side management capacity because it is typically cheaper and easier to procure than traditional generation.

From the “Smart Buildings” side, the a Building Management System (BMS) is a computer-based system that automatically monitors and controls a range of building services, including air conditioning, ventilation, heating, lighting and other consumers of energy within the building or sometimes groups of buildings.

It is evident that the communication between DSM and BMS is continue, in a close exchange of information; in the future, said exchange will be more and more M2M (*machine-to-machine*), i.e. managed completely by machines in automatic way.

So, before to detail the communication layers inside the different kind of smart buildings (*homes, residential buildings, offices/public buildings, data-center, hotels, etc.*), it is

fundamental to show the continuous interaction between the indoor and the outdoor environment, such as controls from remote service platforms and end-user customer interface.

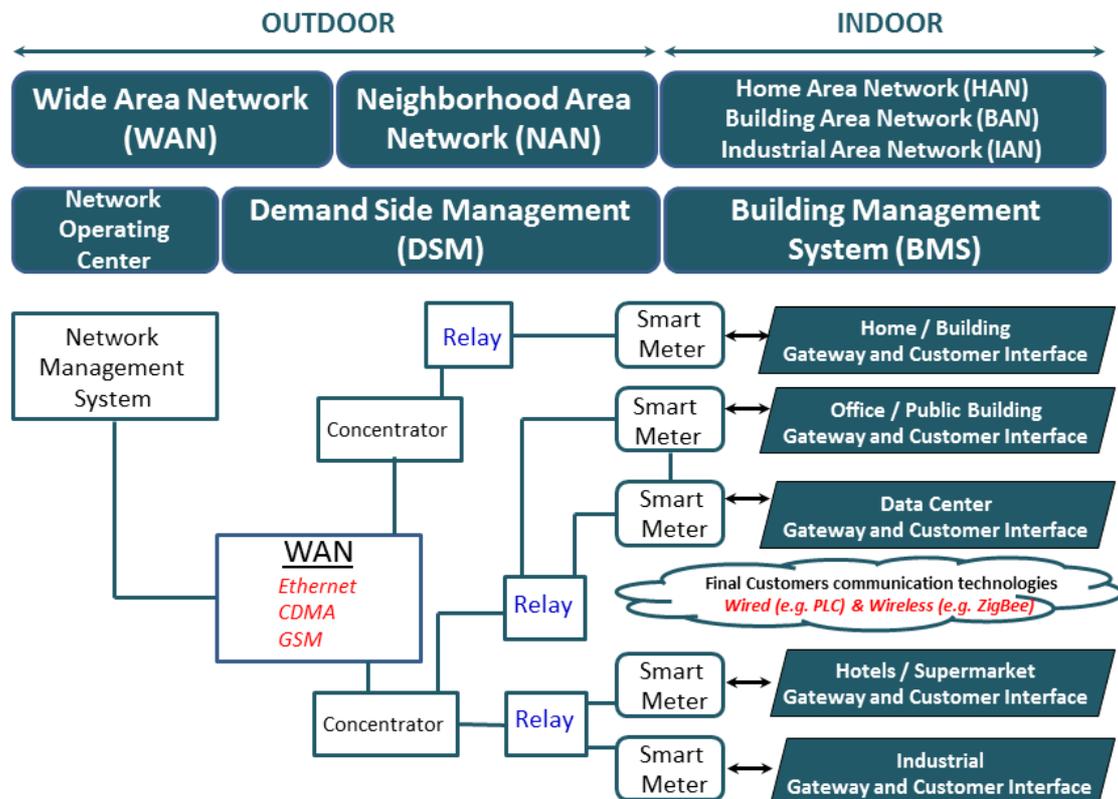


Figure 78: Topological segmentation in WAN, NAN and HAN

For this reason, the communications in Smart Buildings, with their relative technologies and protocols, is typically broken into three segments as shown in **Figure 78**:

- Wide Area Network (WAN): covers long-haul distances from the command center to local neighborhoods downstream.
- Neighborhood Area Network (NAN) manages all information between the WAN and the home area network.
- Home Area Network (HAN) extends communication to endpoints within the end-user home or business.

Each of these segments is interconnected through a node or gateway: a concentrator between the WAN and NAN and a smart-meter between the NAN and HAN. Each of these nodes communicates through the network with adjacent nodes.

The concentrator aggregates the data from the meters and sends that information to the grid operator. The smart-meter collects the power-usage data of the home or business by communicating with the home network gateway or functioning as the gateway itself.

Each segment can utilize different communications technologies and protocols depending on the transmission environments and amount of data being transmitted. In addition to the architecture choice between wireless and Power-Line Communications (PLC), there are a variety of wireless and PLC protocols to choose among.

As often happens, at worldwide level there a lot of standard communication between these actors, as summarized in Table 6.

REGION	WAN	NAN	HAN
--------	-----	-----	-----

North America	Cellular, WiMAX	G3-PLC, HomePlug®, IEEE 802.15.4g, IEEE P1901, ITU-T G.hnem, proprietary wireless, Wi-Fi	G3-PLC, HomePlug, ITU-T G.hn, Wi-Fi, ZigBee, Z-Wave
<b>Europe</b>	<b>Cellular</b>	<b>G3-PLC, IEEE P1901, ITU-T G.hnem, PRIME, Wi-Fi</b>	<b>G3-PLC, HomePlug, ITU-T G.hn, Wi-Fi, Wireless M-Bus, ZigBee, , Z-Wave</b>
China	Cellular, band-translated WiMAX	G3-PLC, RS-485, wireless to be determined	G3-PLC, RS-485, Wi-Fi, to be determined
Rest of the World	Cellular, WiMAX	G3-PLC, HomePlug, IEEE 802.15.4g, IEEE P1901, ITU-T G.hnem, PRIME, RS-485, Wi-Fi	G3-PLC, HomePlug, ITU-T G.hn, RS-485, Wi-Fi, Wireless M-Bus, ZigBee, Z-Wave

Table 6: Communication protocols in the world

In the following paragraphs will be given an overview of the communication layers and protocols adopted for the WAN, NAN and HAN.

**7.2.1 WAN communication protocols**

The **WAN** is the communications path between the grid operator and the concentrator.

A multi-tier network integrates communications throughout the distribution grid and uses a wide area network (WAN). To be fully effective, the utility’s WAN will need to span its entire distribution footprint, including all substations, and interface with both distributed power generation and storage facilities as well as with other distribution assets such as capacitor banks, transformers, and reclosers.

The utility’s WAN will also provide the two-way network needed for substation communication, distribution automation (DA), and power quality monitoring while also supporting aggregation and backhaul for the advanced metering infrastructure (AMI) and any demand response and demand-side management applications.

And many utilities will want to take full advantage of the investment in this WAN infrastructure to run other enterprise networking applications, including wireless communications for work crews in the field, site security with video surveillance, Voice over IP (VoIP), asset management, and more.

The WAN can be implemented over fibre or wireless media using Ethernet or cellular protocols, respectively. Cellular or WiMAX is most commonly used between the grid operator and the concentrator.

Also for the WAN, at worldwide level there a lot of standard communication between these actors, as summarized in **Figure 79**.

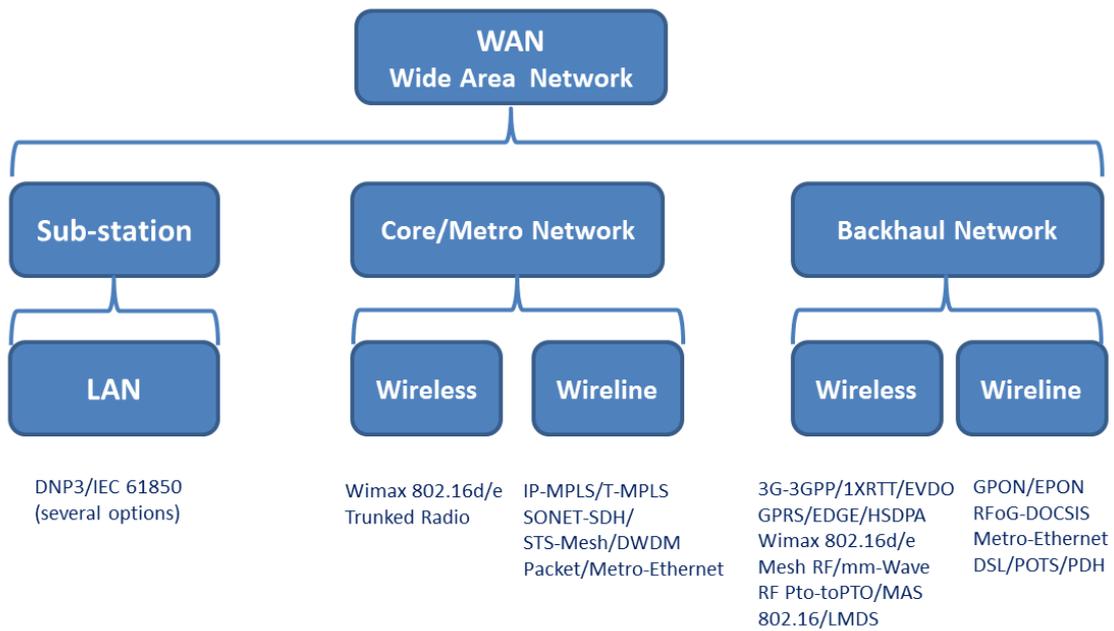


Figure 79: WAN mapping: communication protocols and network technology

### 7.2.2 NAN communication protocols

The NAN is the path between the concentrator and the meter. It uses either wireless or PLC. Typically, the concentrator communicates with anywhere from a few to hundreds of meters, depending on the grid topology and the communications protocol used.

The DSM is obviously placed here, because the energy retailers have to be in communication with the customer (HAN) and with the energy producer & transport (WAN).

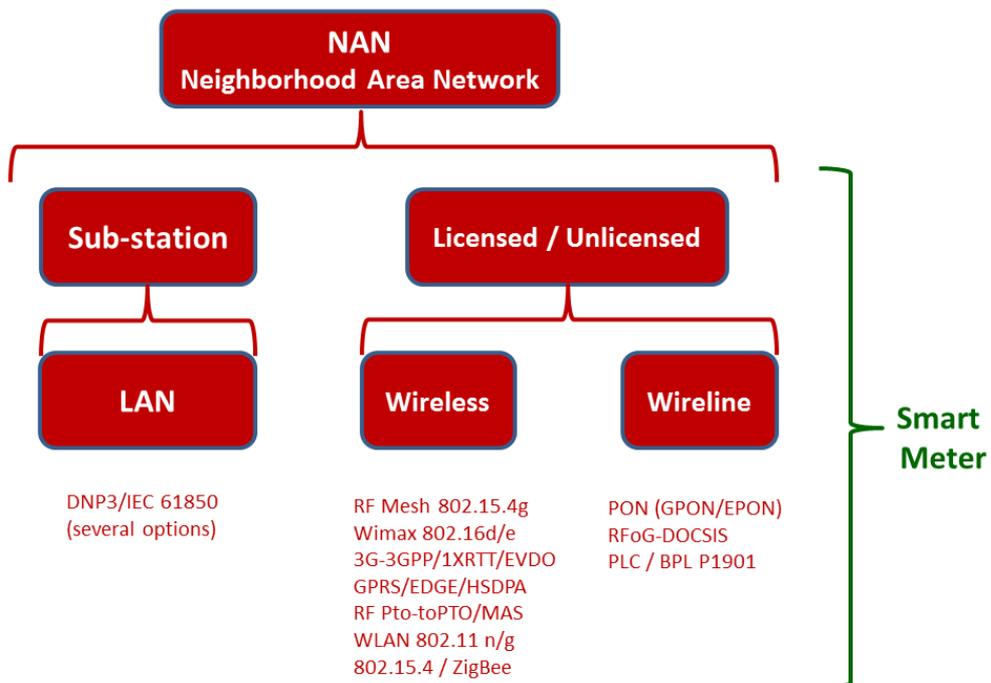


Figure 80: NAN mapping: communication protocols and network technology

Today, In the NAN portion, licensed and unlicensed protocols coexist, as shown in **Figure 80**. For this reason, several standards bodies are currently working with utilities and technology providers to define standards for wireless and PLC protocols.

### 7.2.3 HAN/BAN/IAN communication protocols

The **HAN/BAN/IAN** is used by utilities to extend the reach of their communication path to devices inside the building/home. This network can support functions such as cycling air conditioners off during peak load conditions, sharing consumption data with in-home displays, or enabling a card-activated prepayment scheme.

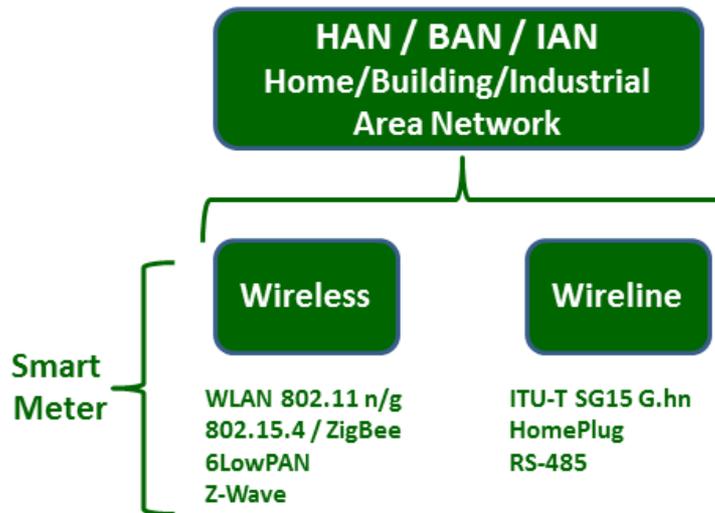


Figure 81: HAN/BAN mapping: communication protocols and network technology

In addition to the communication protocols shown in **Figure 81**, a HAN might also include: peer-to-peer (P2P) communications between devices inside the home; communications with handheld remote-control devices, lighting controls, and gas or water meters; as well as broadband traffic.

Protocols such as RS-485, ZigBee, Z-Wave, and HomePlug are used for this network. If there is a separate home gateway, it is possible that additional protocols could be used to communicate with appliances, thermostats, and other devices. Communications alternatives in the HAN can often coexist, but utility support will probably be limited to technologies needed to support the utility's primary objectives.

The next sub-chapters will detail the communication layers inside the different kind of smart buildings (*homes, residential buildings, offices/public buildings, data-center, hotels, etc.*),

## 7.3 Communication layers inside the different kind of smart buildings

As explained in sub-chapter 7.1, OSGP is a communication protocol based on the OSI (Open Systems Interconnection) protocol model and it is used in conjunction with the ISO/IEC 14908 control networking standard for smart grid applications.

More in detail, the intermediate and lower layers of the OSGP stack leverage the ISO/IEC 14908. The official ISO standard numbers for building automation worldwide are: ISO/IEC 14908-1, ISO/IEC 14908-2, ISO/IEC 14908-3, and ISO/IEC 14908-4. They are widely used in smart grid, smart city, and smart building applications with a myriad of devices deployed worldwide. ISO/IEC 14908 is highly optimized for efficient, reliable, and scalable control

networking applications. The low overhead of ISO/IEC 14908 enables it to deliver high performance without requiring high bandwidth.

At the application layer, OSGP includes the IEEE 1377 standard that provides common structures for encoding data in communication between End Devices (meters, home appliances, IEEE 1703 Nodes) and Utility enterprise collection and control systems using binary codes and XML content.

The IEEE 1377 standard exposes sets of tables that are grouped together into sections that pertain to a particular feature-set and related function such as Time-of-use, Load Profile, Security, Power Quality and more. Each standard Table Set (Data Model) can be expanded or restricted by the Manufacturer of the IEEE 1377 Device or home appliance using XML/TDL descriptive registered syntax (XML-based Table Definition Language) and enterprise data-value management using EDL (Exchange Data Language) in a manner that is machine readable.

IEEE 1377 standard provides tables in support of Gas, Water and Electric sensors and related appliances. It also provides tables for network configuration and management by referencing its companion standard IEEE 1703. IEEE 1377 is co-published as ANSI C12.19 and MC12.19.

In OSGP it is designed to be very bandwidth efficient, enabling it to offer high performance and low cost using bandwidth constrained media such as the power line.

For example, just as SQL provide an efficient and flexible database query language for enterprise applications, OSGP provides an efficient and flexible query language for smart grid devices. As with SQL, OSGP support reading and writing of single attributes, multiple elements, or even entire tables.

As another example, OSGP includes capabilities for an adaptive, directed meshing system that enables any OSGP device to serve as a message repeater, further optimizing bandwidth use by repeating only those packets that need to be repeated.

In a nutshell, OSGP is designed to support the communication requirements between a large scale deployment of such smart-grid devices and utility supplier or suppliers for the purposes of data collection, primarily for billing purposes by the utility or utilities involved, but including the provision of usage information to the consumer and the control of the consumer's use of utility services in the event of shortage of supply on the part of the utility or transport providers or insufficient payment by the consumer for utility services already supplied.

In the next paragraphs, the communication layers for each different kind of smart buildings are described.

### 7.3.1 Smart Home communication layer

With reference to the **Figure 83**, the information coming from the smart meter is distributed to the smart appliances that will adjust their cycles according to the available power and the energy tariff in order to optimize the consumption and to reduce the energy bill to the customer.

The smart info gathers the data sent via powerline (PLC) from the electronic meter and distribute them wirelessly inside the house, typically using the ZigBee communication protocol. ZigBee is a specification for a suite of high level communication protocols using small, low-power digital radios based on an IEEE 802.15.4 standard for personal area networks.

ZigBee devices are often used in mesh network form to transmit data over longer distances, passing data through intermediate devices to reach more distant ones. As shown in **Figure 82**, ZigBee builds upon the physical layer and medium access control defined in IEEE standard 802.15.4 for low-rate WPANs (Wireless Pico Area Network). The specification goes on to complete the standard by adding two main components: network layer and application layer.

In the case of the smart home, several sensor and actuators use the ZigBee radio protocol, so the relative mesh network is called WSN (Wireless Sensors and Actuators Networks). The WSN

get the sensors data and send the control commands to the actuators, in order to manage efficiently the electric loads.

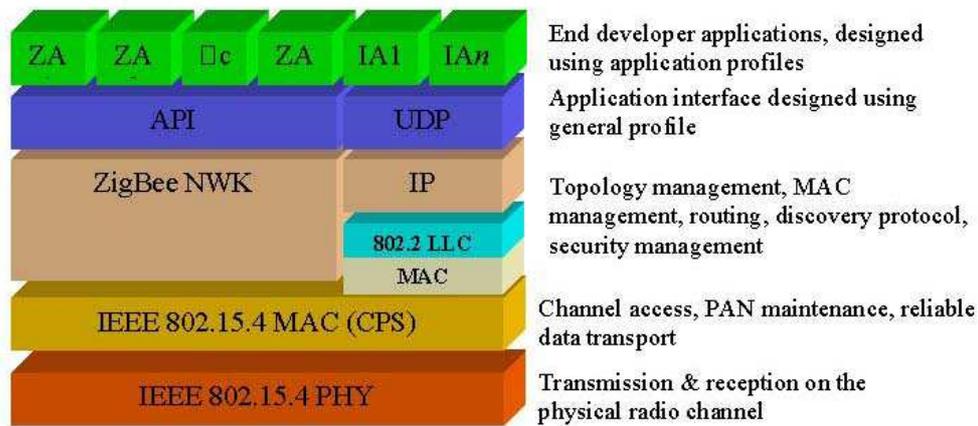


Figure 82: ZigBee Protocol Stack

The smart appliance receive the data via ZigBee from the smart info and manage their processes according the power availability and in agreement with the user preferences.

The Energy Box, which is also the HAN controller, is an ADSL gateway with OSGi (Open Service Gateway initiative) framework and HAN ZigBee wireless communication capability. Thanks to a Wi-Fi communication, the final customer can also read consumption reports and send control commands when he is at home, e.g. using a smart TV.

The Energy Box collects all the data sent from the domestic wireless network and forwards them outside thanks a broadband always-on connection (ADSL / Wi-Fi) giving the possibility to display the info about energy on the WEB portal or a smart-phone from a remote location.

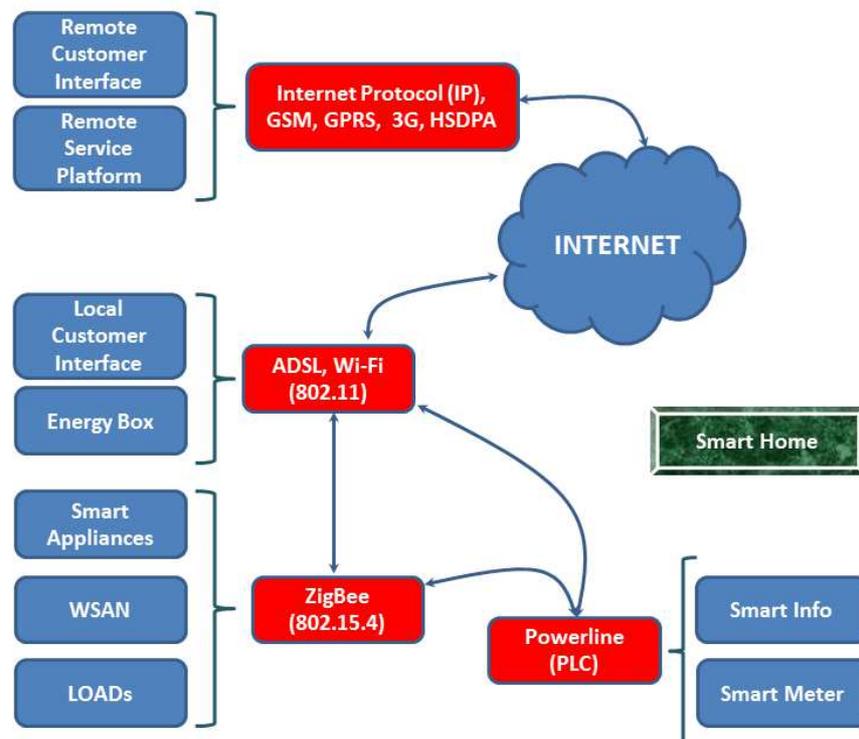


Figure 83: Smart Home communication flow

The remote Service Platform manages, together with any the Home Gateway, the HAN devices and provides service oriented interfaces for the development of third-party applications. It monitors and controls a plurality of individual entities and it represent de-facto the “Supervisory control” DSE (a FINSENY DSE).

### 7.3.2 Residential building communication layers

In a residential building there are common areas where energy wastage is greater because, at difference from the above smart home, the cost of the energy is not perceived individually (e.g. elevators, lighting of the stairs, heating of common space etc.).

For this reason, a residential building is a complex communication infrastructure where is necessary that a machine-2-machine communication will manage the whole. Actually are used manufacturers’ proprietary communications protocols or Internet protocols and open standards such as: BACnet, LonWorks, Modbus, XML, SOAP, DeviceNet etc.

The most popular of them is BACnet (*Building and Automation Control Network*), a building automation and control networking protocol. It was designed specifically to meet the communication needs of building automation and control systems. Typical applications include: heating, ventilating, and air-conditioning control, lighting control, access control, and fire detection systems.

BACnet was designed for wired communication infrastructure, but it is constantly evolving and now it embraces new technologies, such as wireless networking for the smart grid applications.

So, it is becoming common use the term "wireless BACnet" and recently, the ZigBee Alliance (*low-power wireless nodes, IEEE 802.15.4 standard*) announced the establishment of the first BACnet-approved standard for wireless mesh sensors in commercial buildings.

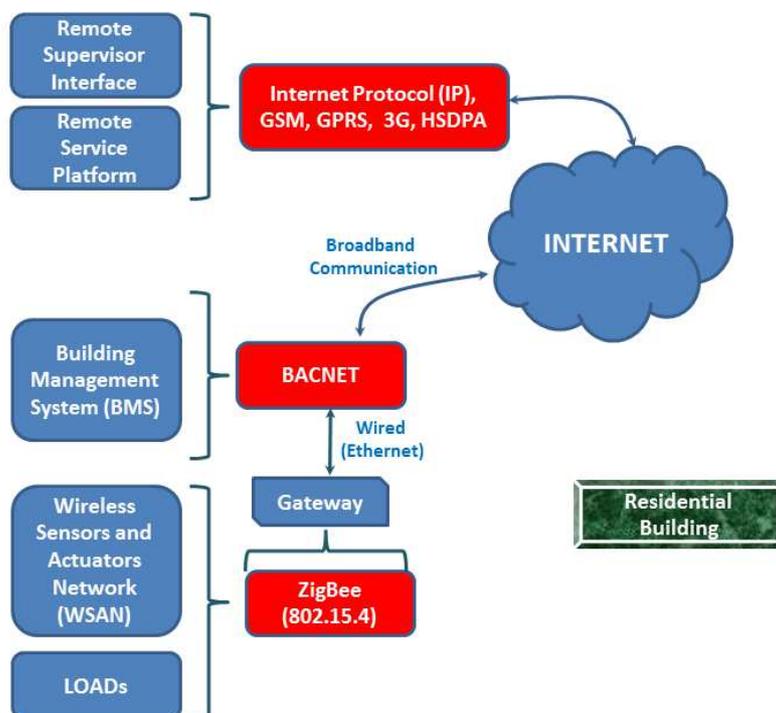


Figure 84: Residential building communication layers

Looking at **Figure 84**, a Building Management System (BMS), through a BACnet based on a wired network (e.g. Ethernet) communicates with the WSAN gateway based on ZigBee radio

communication. In this way, the WSA sends sensors data and receives control command for the actuators which manage the electrical loads (*Air conditioners, free cooling, valve, lighting, fire sensors, etc.*).

The BACnet is also connected to Internet through a broadband communication, typically on optical fiber, using the Internet Protocol; so it is possible to communicate with a remote supervisor interface in those cases where more than one residential building has to be remotely monitored (*also with a wireless communication through a GSM or GPRS or HSDPA protocols*). Moreover a remote service platform, through the IP protocol, can execute sophisticated algorithms in order to optimize the whole power consumption in very complex residential buildings communicating via the Internet with the BMS.

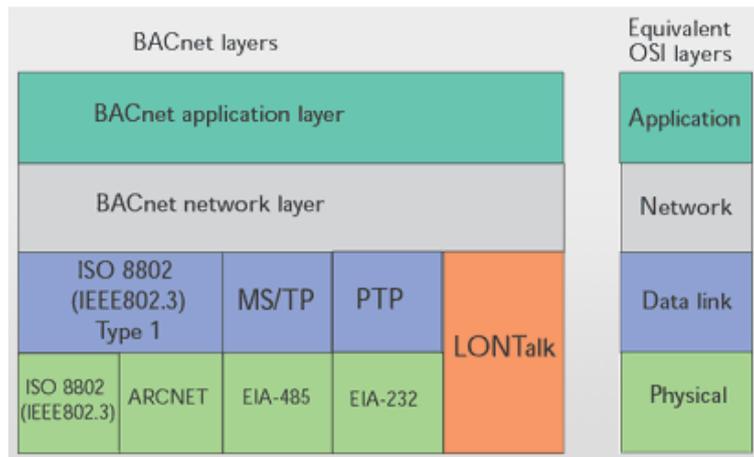


Figure 85: BACnet Protocol Stack

More in detail, BACnet defines a set of hardware and software rules that dictate how data and control information passes across the network. BACnet allows different types of transport protocol and LAN technology (**Figure 85**) allowing the best cost/performance/technology permutation for each situation.

### 7.3.3 Offices/public buildings communication layers

As for the residential buildings, the management of energy efficiency in offices and public buildings is a problem with no easy solution. Offices and the public buildings consume more electricity than any other facility category. Basically, this is due to two reasons: the occupants are not directly involved about the economic aspects (energy bill) and it is required a lot of energy to provide safe and comfortable places to work, shop, eat, and recreate.

The communication layers are similar to the residential buildings ones (*BACnet and ZigBee*), but also the LonWorks Protocol is adopted. It was developed for networking devices over media such as twisted pair, powerlines, fiber optics, and RF. LonWorks is popular for the automation of various functions in industrial control, home automation, transportation, and buildings systems such as lighting and HVAC.

As shown in **Figure 86**, the LonWorks® protocol provides services at each layer of the OSI seven layer reference model. The protocol is open for anyone to implement, and, since from its invention, the protocol has become an ANSI standard, an IEC standard, a Chinese national standard, and recently has achieved ISO standardization.

At physical layer, multiple physical links are supported such as RS-485, power line (PLC) transceivers and third-party wired, wireless, and fiber transceivers. The LonWorks standard provides for nodes to just work together, without any prior arrangement, even if they come from different companies and developers. It provides a way for devices to exchange information and cooperate, to become, in effect, *a federation of peers*.

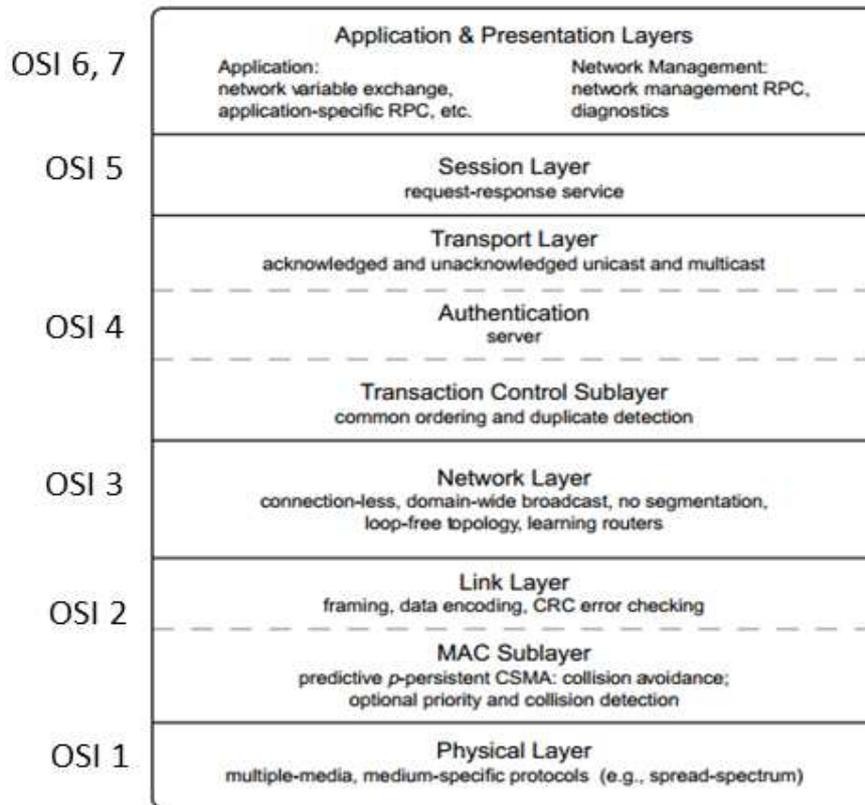


Figure 86: LonWorks Protocol Stack

In **Figure 87** are highlighted the Office/Public building communication layers. The elements of a BMS need to be linked together to transfer data through a communication network. The most common method is data cabling using shielded or unshielded twisted pair; in this case, the typical communication protocol is BACnet. Other options include fiber optics, which can provide increased security and/or radio links (e.g. ZigBee protocol).

Existing communication network cabling may be already available in the offices and in the public buildings, e.g. redundant voice systems, but will need to be checked and proved suitable before inclusion into the project scheme. Dedicated voice networks using ADSL modems, can provide a basic communication infrastructure, particularly on extensive sites where buildings are spread over a large area.

Alternatively, in the office in particular, is increasingly likely to have already a Structured Cabling System (SCS) providing an infrastructure on each floor, with a grid of outlet connections allowing connection of IT equipment such as PC's and printers. BMS may be able to use this existing grid to link major smart grid components directly together (e.g. sensors, actuators and the Control and Service Platform).

ZigBee Radio communications applied to BMS through a gateway are expanding rapidly. Wireless Sensor and Actuator Network (WSAN) are more and more used because their installation does not use cabling and, moreover, their protocols allow adding on-fly new sensors in the mode "Plug & Play".

As for the residential buildings, the BACnet or the LonWorks is also connected to Internet through a broadband communication using the Internet Protocol. This allows the BMS communication with a remote supervisor interface (e.g. through a GSM or GPRS or HSDPA wireless communication) and with a remote service platform, through the IP protocol.

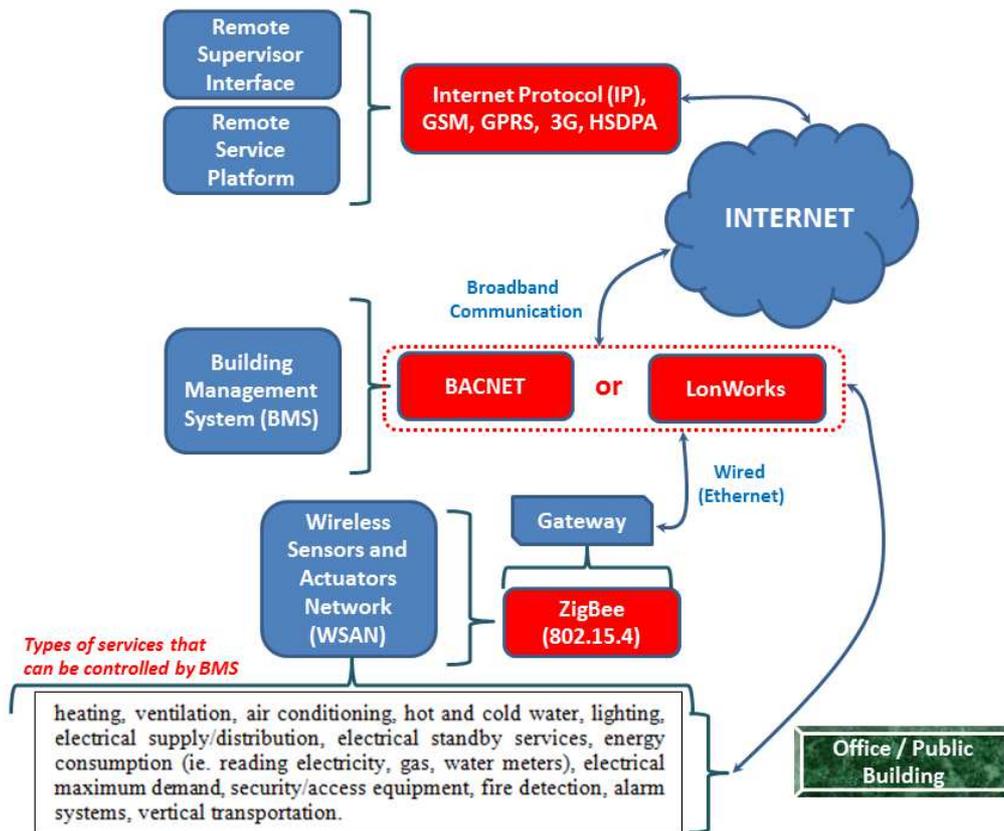


Figure 87: Office/Public building communication layers

### 7.3.4 Data Center communication layers

In a Data Center is mandatory the capability to monitor and cap power in real-time at server, rack, zone, and data center levels. This means the ability to monitor and manage aggregated power consumption within a rack, zone, or data center based on available power and cooling resources. Moreover, in the data center is fundamental to maintain a high quality of service (QoS) even when the power consumption is reduced.

With reference to **Figure 88**, the WSAN system manages conditioners, free-cooling, intrusion etc., and the info/control data are sent wirelessly using technologies such as ZigBee. All the ZigBee radio nodes data are collected from a gateway, which is able to communicate, through a Wi-Fi connection, with the local monitor/control interface.

At the same time, the WSAN data collected by the gateway can be sent, through a high speed Internet connection (*e.g. IP over fiber*), with the remote Supervisor Interface, for sophisticated controls in order to guarantee, for instance, the best compromise between the local conditioning and the relative power consumption.

The power consumption due to the server racks is detected from the smart meter; this information is sent to the workload optimizer typically through a wired communication (*e.g. Ethernet*).

The workload optimizer, typically uses a wireless communications (*e.g. Wi-Fi*) to reach, the data center gateway and, from here, it reaches the remote services platform through a high speed Internet communication, in order to obtain management data command for ensure the best compromise between the quality of service and the power consumption.

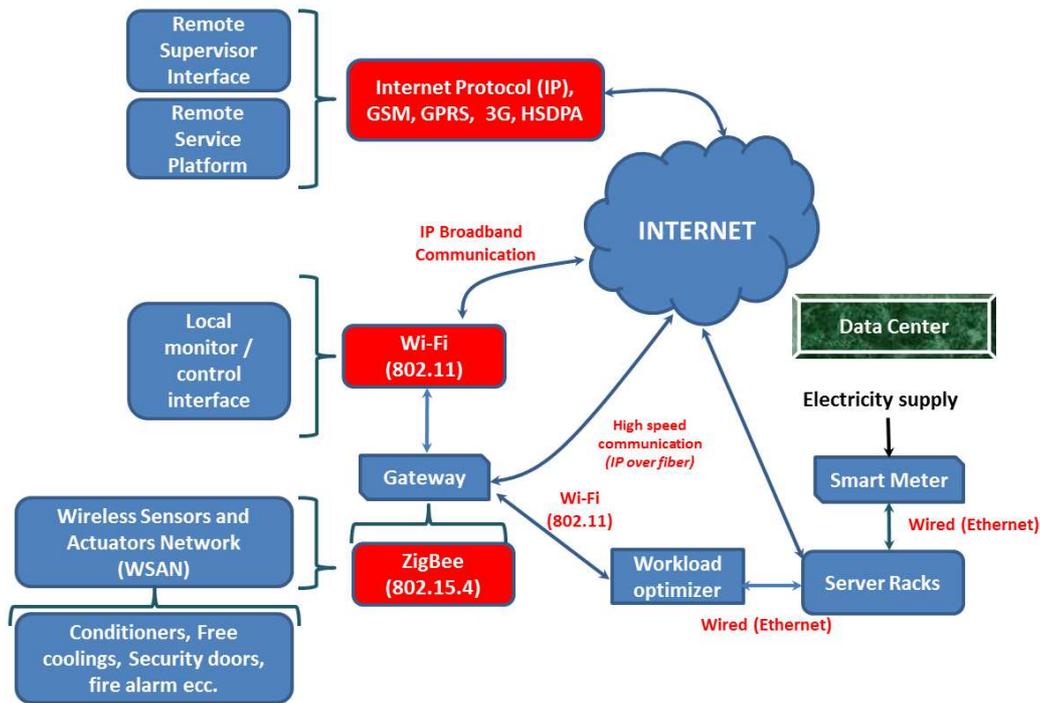


Figure 88: Data center communication layers

### 7.3.5 Hotels communication layers

In this domain, the communications have to take in account that the major difference between residential buildings and hotels is that some energy efficiency actions, such as load shedding, must not impose comfort constraints to the hotel guest because he has paid to have a comfortable housing.

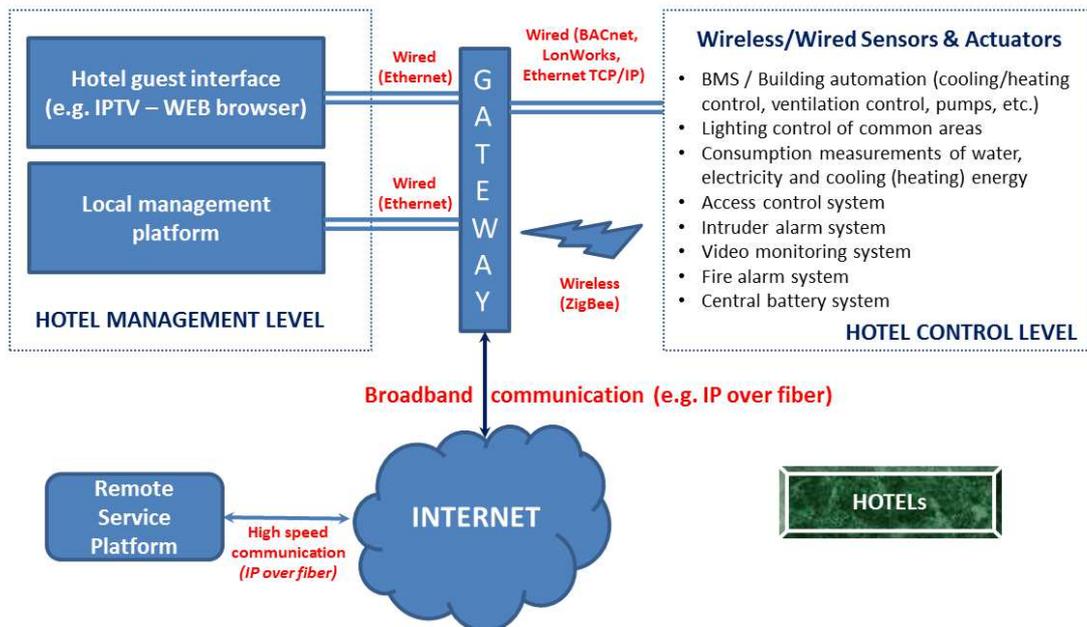


Figure 89: Hotel communication layers

This means that the communication can't handle only a simple system of sensors and actuators, but must take into account many parameters related to the functional comfort of the housing. In addition, the system of a large hotel should handle simultaneously hundreds of rooms, so this complexity is best handled on a platform of services operating in a cloud computing through a broadband communication system.

In the hotel control level box of **Figure 89**, sensors of each type collect the data and send them in wired (e.g. Ethernet) or wireless (e.g. ZigBee radio node) mode to a gateway.

The gateway send all these data through an Ethernet connection (*at medium or large throughput, depends by the number of the rooms*) to the hotel local management platform, where all the systems (*controls of cooling, ventilation and lighting, consumption measurements, access controls, intruder alarms, fire alarms etc.*) shall be monitored and managed.

To enable efficient functional system communication and to provide maximum flexibility and to respond to changes in the hotel layout, the physical communication system employed in the hotels must support the use of several communication protocols, such as the already described BACnet, LonWorks, Ethernet TCP/IP and Internet communication technologies.

For big hotels, the control systems are also connected through a high speed Internet communication (*e.g. IP over fiber*) with a Remote Service Platform, for operate energy trading, centralized remote monitoring, alarm and fault detection of connected building management and security systems.

**7.4 Characteristics of the communication protocols in Smart Buildings**

In the previous sub-chapters it has been shown the communication architectures and the segmentation of the communication protocols. Here are described (separately for WAN, NAN and HAN) the characteristics of the communication protocol keeping in account advantages, disadvantages and recommendation.

With reference to **Table 7**, WAN is a wide area network in which separate areas of coverage or cells are connected, wirelessly or in wired way, to provide service to a large geographic area.

In the eMarket4E future vision, the final user or prosumer will use wireless and portable devices for the Customer Control Interface (CCI) operations. So, in the next Smart Grids scenarios it will be normal to speak of WWAN (Wireless Wide Area Network) services that are typically delivered to smart phones and other handheld devices sold by cellular service providers and their retail partners but other mobile devices can use them as well.

Today some tablet and netbooks and have WWAN cards installed; it is also possible to purchase WWAN cards to install by themselves. Unlike Wi-Fi cards, which can be used in just about any hotspot, WWAN devices must be provisioned specifically for access at own service provider's network. The service provider will take care of billing for roaming access that involves other provider networks.

<b>NETWORK</b>	<b>Protocol</b>	<b>Advantages</b>	<b>Disadvantages</b>	<b>Recommendation</b>
WAN	Wireless (2G / 3G / LTE cellular, GPRS).	Extensive cellular infrastructure is readily available; large amount of aggregated data can be exchanged over a long haul.	Utility must rent the infrastructure from a cellular carrier for a monthly access fee; utility does not own infrastructure.	Wireless usually works best.

Table 7: WAN protocol characteristics

The **Table 8** represents the Neighborhood Area Networks (NAN) protocols, that supports a variety of applications including not only electricity usage measurement and management, but also advanced applications such as Demand Response (DR), which gives users the

opportunity to optimize their energy usage based on real-time electricity pricing information, Distribution Automation (DA), which allows electric grid state monitoring and control, and automatic fault detection, isolation and serves as a foundation for future Virtual Power Plants, which comprise distributed power generation, residential energy storage (e.g., in combination with Electric Vehicle (EV) charging), and small scale trading communities.

<b>NETWORK</b>	<b>Protocol</b>	<b>Advantages</b>	<b>Disadvantages</b>	<b>Recommendation</b>
NAN	Wireless ISM	Long range; leaps transformers	Currently proprietary; dead spots complicate installation and maintenance	Useful in some topologies, such as in the U.S.
	IEEE 802.15.4g	Long range; leaps transformers	Not yet an accepted standard	Useful in some topologies
	ZigBee	Low cost; low power consumption allows battery operation; well-known standard	Low data rate; very short range; does not penetrate structures well	Unlikely to be used in NANs
	First-generation PLC (FSK, Yitran, Echelon)	Low cost	Unreliable; low bandwidth	Bandwidth and reliability inadequate for the smart grid
	Early-generation Narrow-band OFDM	Better range, bandwidth, and reliability than FSK	Does not cross transformers; does not coexist with first-generation PLC	Not recommended for new designs due to cost and compatibility concerns
	Broadband PLC	High data rate	Does not cross transformers	making it too costly for most large-scale deployments
	G3-PLC	Highly reliable long-range transmission; crosses transformers, reducing Infrastructure costs.	Not yet an accepted standard	Excellent for NAN worldwide

Table 8: NAN protocol characteristics

The **Table 9** represents the Home Area Network (HAN) protocols, a dedicated network connecting devices in the home such as displays, load control devices and ultimately “smart appliances” seamlessly into the overall smart metering system.

<b>NETWORK</b>	<b>Protocol</b>	<b>Advantages</b>	<b>Disadvantages</b>	<b>Recommendation</b>
HAN	ZigBee	Well-known standard that offers low cost and low power	Very short range; does not penetrate structures well	Well suited for communication between water and gas meters
	Wi-Fi	Popular technology with high data rates	Medium range; does not penetrate cement buildings or basements	Good for user applications, but no provisions for meeting utility objectives
	First-generation PLC (FSK, Yitran, Echelon)	Low cost	Not reliable in home environments	Unlikely to be used in homes due to high levels of interference
	Early-generation narrowband OFDM	Better range, bandwidth, and reliability than FSK	Does not cross transformers; does not coexist with first-generation PLC	Not recommended for new designs due to cost and compatibility concerns
	Broadband PLC	High bandwidth	Short range is not sufficient for communicate with NAN	Good for user applications, but no provisions for meeting utility objectives
	G3-PLC	Highly reliable; sufficient data rate; IPv6 enables networking with many devices	Not yet an accepted standard	Excellent for HAN worldwide

Table 9: HAN protocol characteristics

### 7.5 Communications protocol constraints

Each of the communication scenario described above entails different requirements for the underlying communications technologies. These requirements can be examined across five primary criteria:

- bandwidth,
- latency,
- reliability,
- security,
- cost.

Note that the requirements described in the **Table 10** are general assessments, since absolute measures vary with specific instantiations. The intent is to highlight the issues to be considered when developing detailed requirements.

Attribute	Notes	Parameter Definitions	Smart Meter – AMI Constraints	Substation Automation Constraints	Distribution Automation Constraints
Bandwidth	[1]	<p><u>Low</u>: &lt;250Kbps</p> <p><u>Medium</u>: 250Kbps to 1Mbps</p> <p><u>High</u>: &gt; 1Mbps</p>	Low-Medium	Low-High	Low-Medium
Latency	[2]	<p><u>Loose</u>: tolerate both high latency and high variability in latency</p> <p><u>Medium</u>: some relative limits to absolute latency and/or variability</p> <p><u>Tight</u>: strict requirements for absolute latency and/or variability.</p>	Loose	Medium-Tight	Low-Medium
Reliability	[3]	<p><u>Low</u>: no operational harm if connectivity lost for significant time (minutes/few hours).</p> <p><u>Medium</u>: operations impacted, but unlikely loss of service if connectivity lost for significant time.</p> <p><u>High</u>: significant harm may occur if connectivity lost for significant time.</p>	Low-Medium	High	Medium-High
Security	[4]	<p><u>Low</u>: no major operational harm if link were intentionally compromised</p> <p><u>Medium</u>: significant but limited harm if link were intentionally compromised</p> <p><u>High</u>: highly visible and widespread harm if link were intentionally compromised.</p>	Medium-High	High	High
Cost	[5]	<p><u>Low</u>: relatively low infrastructure and operating costs</p> <p><u>Medium</u>: relatively moderate infrastructure and operating costs</p> <p><u>High</u>: relatively high infrastructure and operating costs.</p>	Low	Medium-High	Medium-High

Table 10: Communications protocol constrains

[1]: is the difference between the upper and lower frequencies in a continuous set of frequencies.

[2]: is the measure of time delay experienced in the communication system.

[3]: is the ability of the communication system to perform its required functions under stated conditions for a specified period of time

[4]: is the degree of protection to safeguard the Smart Grid systems against danger, damage, loss, and crime.

[5]: are the infrastructure and operating costs.

## 8 Components & Communication Infrastructure

### 8.1 Components

This WP addresses all types of buildings in a comprehensive way, as self-contained systems that encompass all the fixed, movable and mobile physical components of the building.

Buildings are of different types. This starts from smaller residential (home) to large commercial buildings. The latter cover a wide range in variety, may it be offices, schools or hotels on the one hand and industrial or transportation buildings on the other hand. All of these specific buildings have certain constraints and demands. This WP addresses various kinds of buildings. With respect to the FINSENY objectives the buildings only cover the Customer and DER domains in the SGAM plane. The DER domain is used to consider the units for generation and storage of energy in the customer premises. That is why the baseplates of the scope of this Smart Buildings WP have been stretched in this deliverable.

In the hierarchy of the power system automation, the buildings are at the lowest levels. In consequence they can be considered mapped in the process and field level. Nevertheless they interact with above layers, notably for aggregation and commercial use cases. In addition the smart buildings can contain own generation equipment and intelligent components to play a key role. In a very broad view, the target systems comprises all parts of the buildings and all pieces of building equipment that have a direct or indirect impact on the energy input and output of the building. This includes all appliances/apparatuses that consume, generate or store energy, the components of the building such as walls and windows that regulate the exchange of energy between the inside and the outside, but also, in a more indirect way, subsets of the building such as floors or rooms and that make sense as separate units for managing energy in the building.

We can map the components referenced in the deliverable D4.1 [1] on the component layer of the SGAM methodology and put them into the table below. On the component level, it is not intended to show Software or Management function, which reside on a higher level. Only components or equipment are available. Due to the fact, that Smart buildings have a broad coverage, these can be seen as examples and are not complete. E.g. you can find more power generation equipment beside PV or wind in and around buildings. As we focus on electrical energy, there are even more aspects to other energy formats and we need to consider interfaces on ICT level and on the energy flows to exchange. Examples are given in the table below, which gives an overview of components, together with their definition.

Table 11: Component overview

Component	Role
<b>ICT Infrastructure containing functionality to measure, communicate and control the energy related information</b>	
Home Energy Monitor	Interfacing equipment with the smart buildings (can be a complex system involving electrical, electronics and other components).
Grid Interfacing equipment	Interfacing equipment with the Grid (can be a complex system involving electrical, electronics and software components). Used to obtain tariff information and to support the selling of the locally produced energy to the grid.
Smart Building interfacing equipment	Interfacing equipment with the smart building (can be a complex system involving electrical, electronics and software components but on a lower wattage compared to the Grid interfacing equipment).

Component	Role
Building Gateway	Equipment connected to the appliances, sensors and actuators of the building, including the meter. It is also connected to a router to gain access to the web but also to the other IP devices of the building, such as PC, TV, smartphones. It may act as a communication gateway, for example to send to the cloud metering data, either from the meter, from smart plugs or from the smart appliances.
Appliances	All equipment held by the customer in his home and that consume energy, and, so, that are connected to the electrical installation of the home. Appliances are for example: the hot water tank, electric heaters, air conditioner, washing machine, dishwasher, dryer, fridge, oven, lights and other appliances. Smart appliances are appliances that have communication abilities and that can meter and send their energy consumption, and react to information such as tariff periods, direct orders from an energy manager device etc.
Smart Meter	Equipment for measuring and reporting aspects of electrical energy usage of the building as a whole in order to participate in various contracts and security facilities: active or reactive energy, power, time... In this case it includes embedded electronics to allow the device to communicate such readings / measurements over a wired or wireless network. A meter records its measurements in the form of index (one per tariff period type) or load curves. Some smart meters have a switching device to limit the power output in conformance with the contracted subscribed power. It could also be equipped with a HAN interface for delivering information (index, tariff period, etc.) to devices in the HAN.
Smart Plug	One device connected within a building to measure or control energy. This Device with wireless communicating capabilities, designed to be attached to a socket and to provide power to a single appliance. The Smart plug is then able to measure energy consumption and to provide readings of instant power and accumulated consumed energy. The Smart plug functions, in many respects, as a Smart Meter for a single appliance but may also accept commands to turn on / off the power.
<b>Energy related Generation (Examples)</b>	
PV Panel	<p>It uses to harvest solar power to produce electricity.</p> <p>In case of PV panel associated to a building, it contributes to the building's autonomy in the event of a power failure.</p> <p>In case of association with batteries/fuel cells, it electricity production serves to recharges.</p>
Solar Panel	Used to harvest solar power to heat water. Hot water can be an energy store and thermal energy is, of course, a form of energy. The system should also include appropriate metering and communication functionality to allow readings on the water temperature (and maybe, solar intensity) be reported over a software interface. The network will likely be wired (wireless connectivity is not important for this component).
Wind turbine	Used to harvest wind power to produce electricity. Same rationale as in the case of the "PV Panel" equipment.
<b>Energy related Storage (Examples)</b>	
Battery	Stores electrical power in the form of chemical energy. Contributes to the building's energy reserves that can be drawn upon in the case of a power

<b>Component</b>	<b>Role</b>
	failure.
Fuel cell	Stores electrical power in the form of fuel chemical energy. Same rationale as the Battery component.
Hot Water Tank	Used to store thermal energy for hot.

## 8.2 Communication infrastructure

### 8.2.1 Generic infrastructure (TI)

Even if the needs and the management of the energy change in the various specific domains (*home, residential, office, public building and data center*) the generic functional principle of the architecture as a whole is substantially the one illustrated in the Figure 90, where are shown the communication infrastructure and its operating flow that can be described in the following way:

- Communication layer allows the generation, transmission and distribution infrastructures to achieve three key objectives: intelligent monitoring, security, and regional load balancing. Using two-way communications, data can be collected from the smart meters located throughout the grid and transmitted directly to the grid operator’s control room. In this way the distribution of energy arrives at the electronic meter of the various domains in part already balanced.

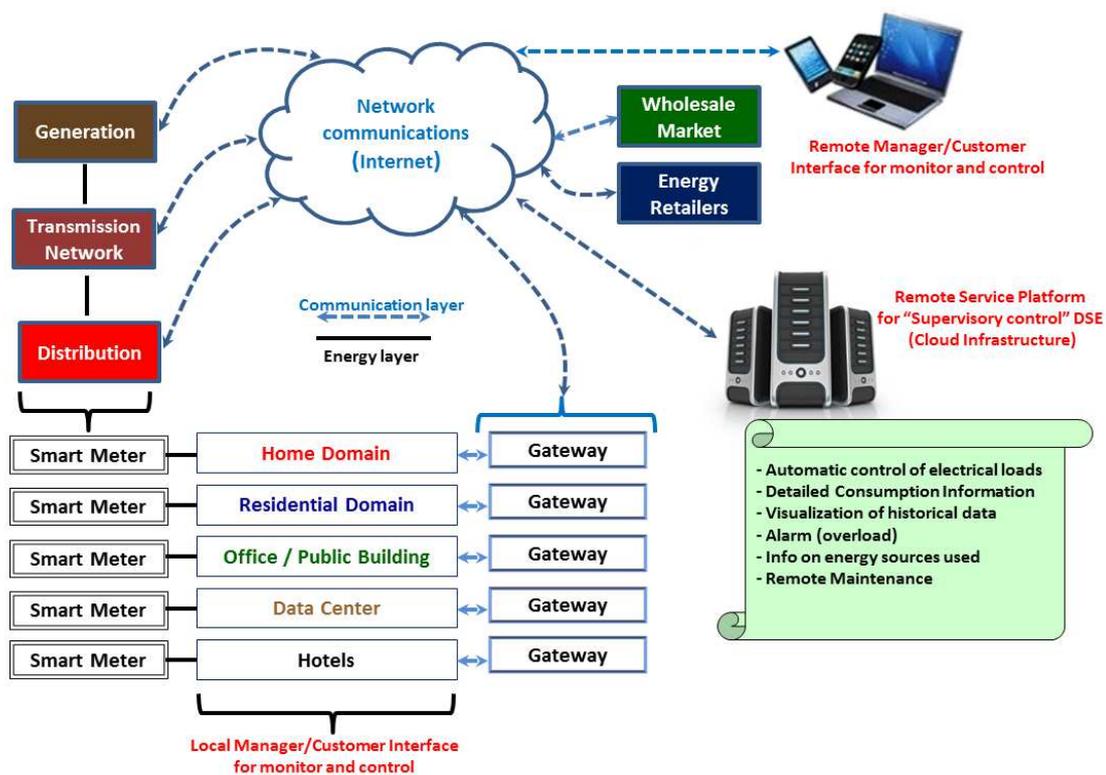


Figure 90: Communication in domain specific infrastructure.

- Information from the smart meter is distributed to the loads (*different for any specific domain*) that will adjust their cycles according to the available power and the energy tariff in order to optimize the consumption and to reduce the energy bill.
- The gateway, with typically an OSGi (Open Service Gateway initiative [7]) framework collects all the data sent from the specific domain network and forwards them outside thanks a broadband always-on connection giving the possibility to display the info about energy on the WEB portal or a smart-phone.
- The remote Service Platform elaborates the collected data and monitors and controls a plurality of individual entities and it includes the “Supervisory control” and the “Demand Side Manager” DSE. Finally, it provides service oriented interfaces for the development of third-party applications.

For each specific domain, the Remote Service Platform is unique for all the gateways, and so it requires a lot of computing power, especially to run complex algorithms on the analysis of consumption and to estimate future needs of the final customer. For this reason, the service platform is today typically installed in remote data centers in cloud computing environment that will be described in sub-chapter 8.2.3.

Communication networks can be wired or wireless networks, each solution provides different advantages and disadvantages.

- **Wired networks** are those that require a wired structure at building level. This infrastructure may already be present but if not, should be provided to interconnect the different component infrastructure inside the building or to interconnect it to other smart building components outside the own building.

The most important features that wired networks provide are more bandwidth and more reliability. On the contrary, the main disadvantage is the need to extend wires in an already existing building and the inconveniences this may lead to.

- **Wireless networks** have experienced a major growth in last years and they have largely replaced the fixed networks, especially in smart building environments. The main features of these networks are:
  - Fast deployment
  - Does not require the installation of a wire infrastructure
  - The available bandwidth and reliability of these networks is lower than fixed networks, but nowadays those features are changing.

There are several classes of wireless networks technologies:

- Cellular technologies: GPRS, UMTS, LTE, ...

They are based on the use of a SIM card providing a long range communication (Kms). They have been widely deployed during late years obtaining a good coverage. Roaming and billing functionalities are also covered.

Main problem of using these cellular based technologies are the high power consumption of the devices, but in the smart building domain this problem doesn't seem to be very important as they already have electric power infrastructure available.

- WLAN technologies:

Even the coverage of these technologies is lower than in the cellular one, it is acceptable in most of the urban areas where buildings are located. They provide a medium range communication (100 – 200 m).

The main problem of this technologies is the need of taking care of the penetration of the signal through walls, water, ... and the interferences due to the use of high frequencies (2.4 or 5.2 GHz).

As in cellular technologies, the power consumption is still high.

- Short range technologies:

These technologies, as IEEE 802.15.4 family, reduce the power consumption from the above technologies allowing small size and low cost devices. But coverage is an issue that needs to be taken into consideration when deploying these networks.

The communication is in a short range (10 – 50 m)

- RFID / NFC:

These technologies provide a very short range communication (cm – some meters).

## **8.2.2 Domain-specific infrastructure (TI)**

Home, offices, hotels, data centers and more in general each space in today's buildings have its own requirements. And the infrastructure that supports the needs for lighting, heating, cooling, ventilation and water is complex. The building consumes power and generates costs that must be managed proactively if a portfolio of buildings is to achieve maximum efficiency. So, for each specific domain, in the following sub-chapters will be described how the communication infrastructure can help to reduce and optimize the power consumption.

### **8.2.2.1 Home domain communication infrastructure**

The home communication infrastructure gives the ability to communicate and interact, with people, systems and other objects.

Interconnected devices make possible remote access to information about a device and control of the device.

This enables services throughout the Internet, removing complexity from the home and lowering costs for the service providers.

At the same time, it supports the aggregation of information and control of devices throughout the network. This means that consumers can get a consistent view of their devices, both from home and from mobile devices.

For service providers, it provides an aggregate view of customer characteristics according to criteria such as geographic location, consumption patterns, or types of service.

But this communication infrastructure must have also the ability to make decisions based on data, leading to better outcomes. Intelligent devices support the optimization of their use, both for the individual consumer and for the service provider.

For instance, a utility can send signals to consumers' homes to manage discretionary energy use in order to reduce peak loads. By coordinating this process throughout an entire service area, the utility can optimize the peak reduction, while saving the consumers money on their bill.

The communication infrastructure shown in Figure 91 is intended to identify the main categories of devices in the Home Domain, without any limitation to the possibility for a device to implement functionalities from more than a category.

As an example, an advanced Smart Appliance, provided with a rich user interface, could also implement functionalities typical of a Customer Interface.

In the same way, while typical smart appliances are smart white goods, also a personal computer, able to perform such operations, should be considered an appliance from this perspective.

It is essential to clarify that the structure shown in Figure 91 is reproduced for a single house, while the Service Platform is unique for a set of local apartments and it coincides with the DSE FINSENY "Supervisory Control" (as defined in deliverable D8.2 [8]).

In fact, the "Supervisory control" DSE refers to a lightweight joint control of several individual controlled entities. In other words, this DSE acts not only for a smart-home but also for one or more Smart Buildings.

Anyway the customers are always able to impose own choices by making use of their customer interface.

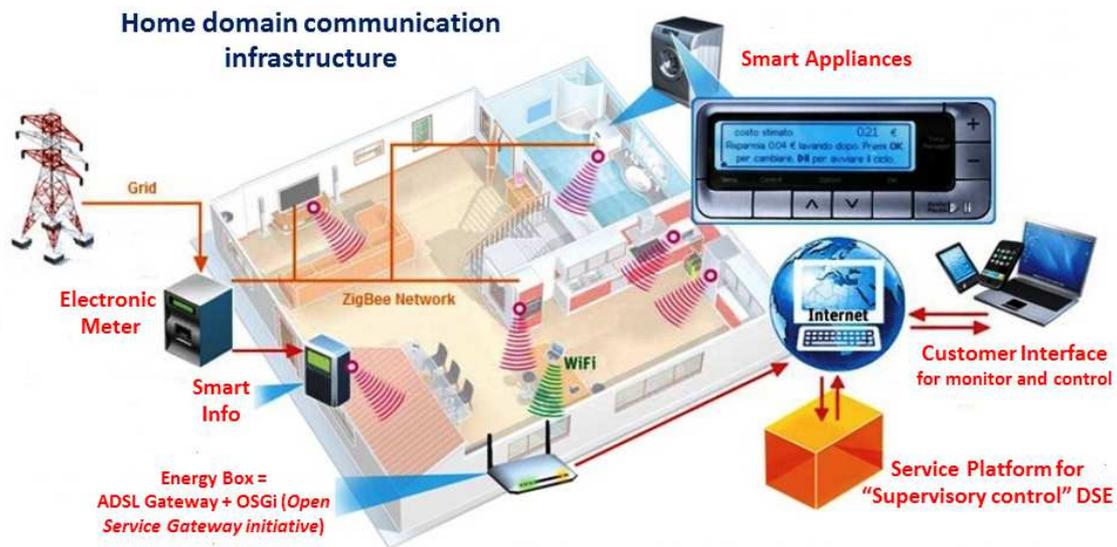


Figure 91: Home domain communication infrastructure.

With reference to the Figure 91, here following is described the communication infrastructure flow:

- The power information coming from the electronic meter are distributed, in wired and/or wireless mode, to the smart appliances that will adjust their cycles according to the available power and the energy tariff in order to optimize the consumption and to reduce the energy bill to the customer.
- A **WSAN** (Wireless Sensor and Actuator Network), typically based on **ZigBee** network, allows to get information from the home sensors and send commands toward home actuators in order to manage all the electric loads in the most efficient way.
- The smart info gathers the data sent via power line communication (**PLC**) from the electronic meter and distribute them wirelessly inside the house.
- The smart appliance receive the data from the smart info and manage their processes according the power availability and in agreement with the user preferences.
- The Energy Box, which is also the HAN controller, is an **ADSL** gateway with **OSGi** (Open Service Gateway initiative) framework and **HAN** (Home Area Network) wireless communication capability. It collects all the data sent from the domestic wireless network and forwards them outside thanks a broadband always-on connection giving the possibility to display the info about energy on the WEB portal or a smart-phone.
- The remote Service Platform manages, together with any the Home Gateway, the HAN devices and provides service oriented interfaces for the development of third-party applications. It monitors and controls a plurality of individual entities and it represent de-facto the "Supervisory control" FINSENY DSE.

- The local customer interface is wirelessly connected at the Energy Box through a **Wi-Fi** connection and it allows to monitor the info about home energy and to control manually the electric loads.
- The remote customer interface (smart-phone, tablet etc.) is connected through **Internet** at the remote Service Platform that allows to monitor the info about home energy and to control manually the electric loads. Moreover, from the remote Service Platform it is possible to receive several VAS (Value Added Services) such as automatic control of electrical loads, overload alarm with instant load shedding, visualization of historical data and info about the kind of energy sources used and the relative price.

**8.2.2.2 Residential domain communication infrastructure**

The residential buildings scenario considers as target system the building shell and common areas of residential buildings, excluding individual apartments/buildings, which are targeted by the smart home scenario.

A residential building can provide communication among automated building systems. The building operator can enjoy a single interface capable of controlling lighting, security, heating ventilating and air conditioning systems (HVAC), fire and other building systems communicating over a single broadband infrastructure, which also supports the occupants/tenants’ voice and data communication needs.

To date, the most commonly used standard communication protocol in the residential building is BACnet. It is a communications protocol for building automation and control networks. It was designed to allow communication of building automation and control systems for applications such as heating, ventilating, and air-conditioning control, lighting control, access control, and fire detection systems and their associated equipment. The BACnet protocol provides mechanisms for computerized building automation devices to exchange information, regardless of the particular building service they perform.

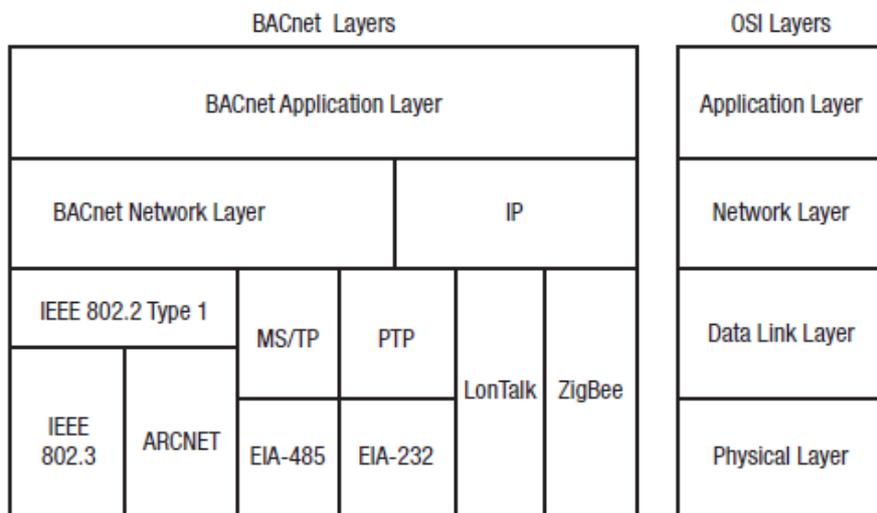


Figure 92: BACnet communication layers

With reference to Figure 92, along the BACnet layers, several kinds of wired (e.g. Ethernet) and wireless (e.g. ZigBee) communication are supported. Moreover, as shown in Figure 93, BACnet is an open technology which means that any manufacturer can use the protocol to transfer data of connected devices like sensors and actuators for fire, lighting, HVAC, etc.

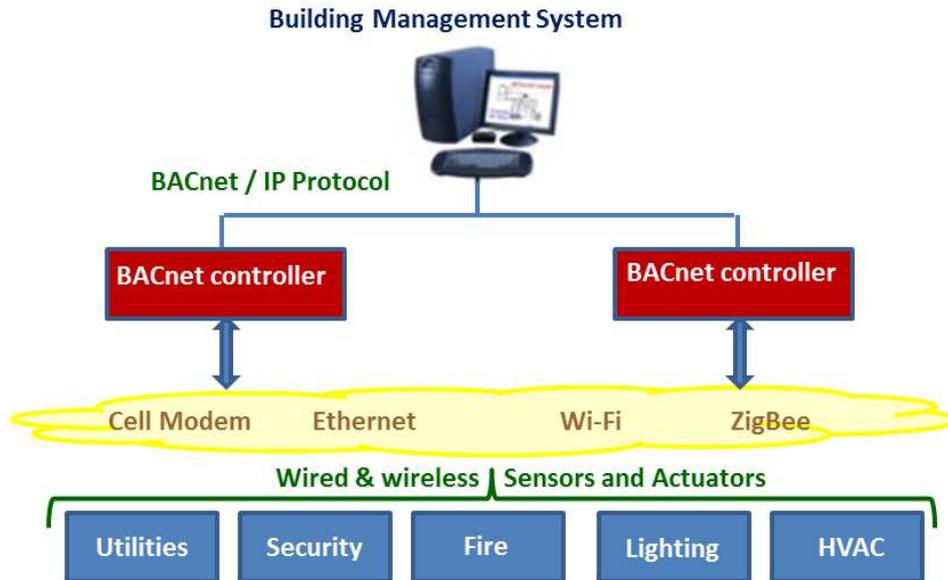


Figure 93: BACnet communication infrastructure

BACnet is mainly used within the following building applications:

- Heating, Ventilation, Air conditioning (HVAC)
- Lighting control
- Security systems and fire alarm systems
- Elevator monitoring
- Pump control and monitoring
- Building Access control
- Energy supply monitoring

As shown in Figure 93, a building manager can remote control the connected devices, through the BACnet controllers. The BACnet controllers are multi-purpose entities with Inputs and Outputs that can be used to monitor and control building services plant or configured to customer specific applications. A BACnet device also offers a variety of services, such as data transfer, scheduling, trending, alarming etc. And the status messages, alarms, operation hours, energy consumption, etc. from the connected devices allow the building manager continually to optimize operation of the building.

### 8.2.2.3 Office/Public Building domain communication infrastructure

Even if the office/public building energy management has several similarities with the Residential domain, the management of energy efficiency in offices and public buildings is a problem with no easy solution. This is because the occupants are not directly involved about the economic aspects (energy bill).

So that some methods for the automatic enforcing better energy consumption have been introduced, like “*whenever workers leave a room, the lights should be turned off*”. Lighting, air conditioning, fire alarm, opened windows etc. have to be controlled as much as possible in an “*automatic way*”. This can be done through the use of sensors/actuators system loops, as shown in Figure 94.

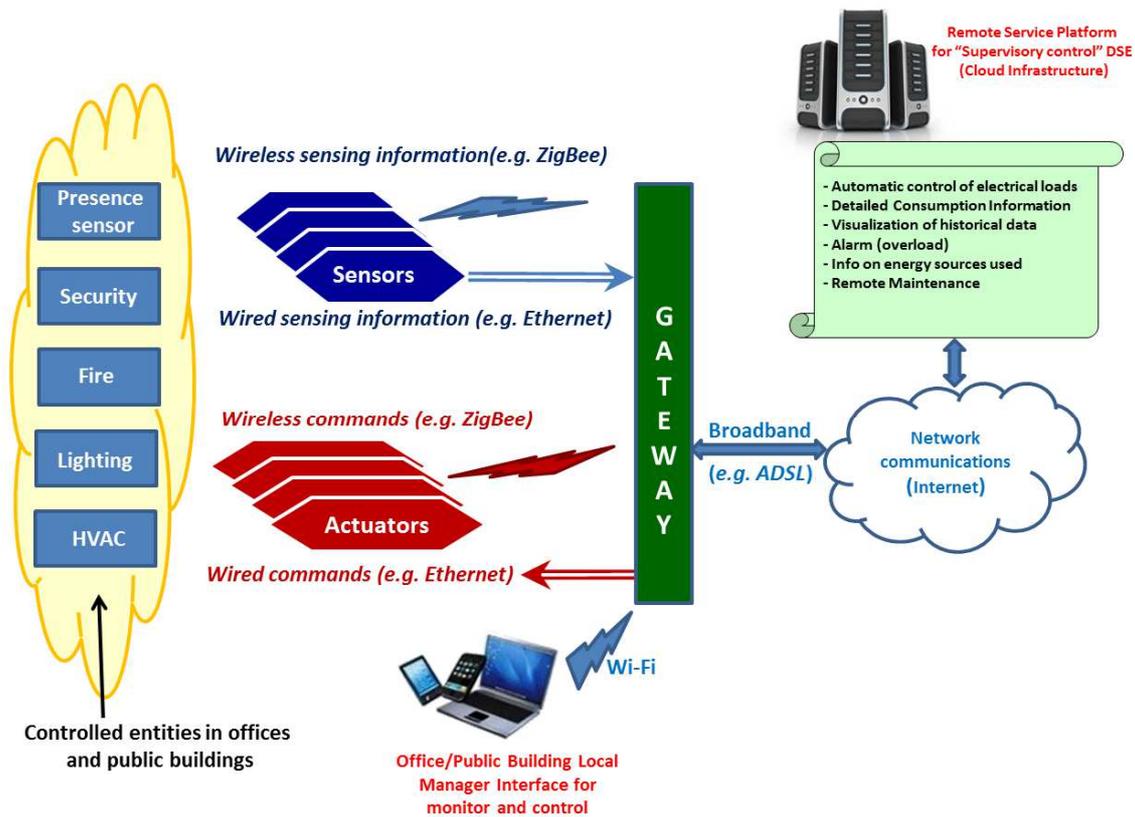


Figure 94: Office/Public Building domain communication infrastructure

Typically, the offices and public buildings communication infrastructure make use of three levels of communication:

- Short range and low throughput communication between the sensors/actuators and the gateway, using well known wired and wireless technologies.
- Medium range and medium throughput communication between the gateway and the Internet world.
- Long range and high throughput communication between the Remote Service Control Platform and the whole Network communications (Internet). Here a high throughput communication is requested because a single Remote Service Control Platform serves, in real time, a lot of offices and public buildings.

The sensors data collected from offices and public buildings are sent in wireless (*e.g. ZigBee communications*) or wired (*e.g. Ethernet*) mode to one or more gateways. The gateway forwards all the sensors data, through a broadband connection (*e.g. ADSL*), to a Remote Control Service Platform connected to Internet.

Always with reference to Figure 94, the Remote Service Platform represents the closure of the system loop between the sensor data (*information*) and control commands (*decisions*) that the platform sends to the actuators in offices and public buildings. To do this, the Remote Service Platform runs algorithms for the “*data filtering*”, “*data mining*” and “*best decision*”.

As instance, IF the sensors data (*in this case presence sensors*) give the information that the room is empty AND the air conditioner is running, THEN a control command is sent to an actuator to turn off the air conditioner. This can be particularly efficient during holidays and weekend when, typically, the offices and the public buildings are completely empty.

Finally, there is a wireless communication (*typically using Wi-Fi*) between the gateway and a local interface (*e.g. smartphone, tablet, notebook etc.*) through which it is possible to direct control in place of any reports of faults and alarms.

### 8.2.2.4 Data Center communication infrastructure

Reality as data centers are among the largest consumers of energy and for this reason the smart grids approach is kept very into consideration.

In FINSENY D4.1 [1] the following key actions were described to contain the data-center power consumption:

- Optimize the conditioning power consumption
- Optimize the free-cooling power consumption
- Optimize the power consumed by servers
- Manage the continuity and autonomy of the service
- Optimize power workload maintaining a high quality of services (QoS)

As well as for the other domains described herein, to achieve the above key actions, it is necessary to use a flexible and efficient communication infrastructure that allows a real-time interaction between the information collected by the sensors and the consequent actions sent to the actuators.

Always referring to the FINSENY D4.1 [1] use cases, the Figure 95 shows the typical communications infrastructure for a data-center. About the sensors/actuators system to manage conditioners, free-cooling, intrusion etc., the info/control data are sent wirelessly using technologies such as ZigBee. The workload optimizer typically uses a wireless communications (e.g. Wi-Fi) to reach, through the gateway and then the Internet connection, remote services for alarm (e.g. excess of computation demand, etc.) and maintenance.

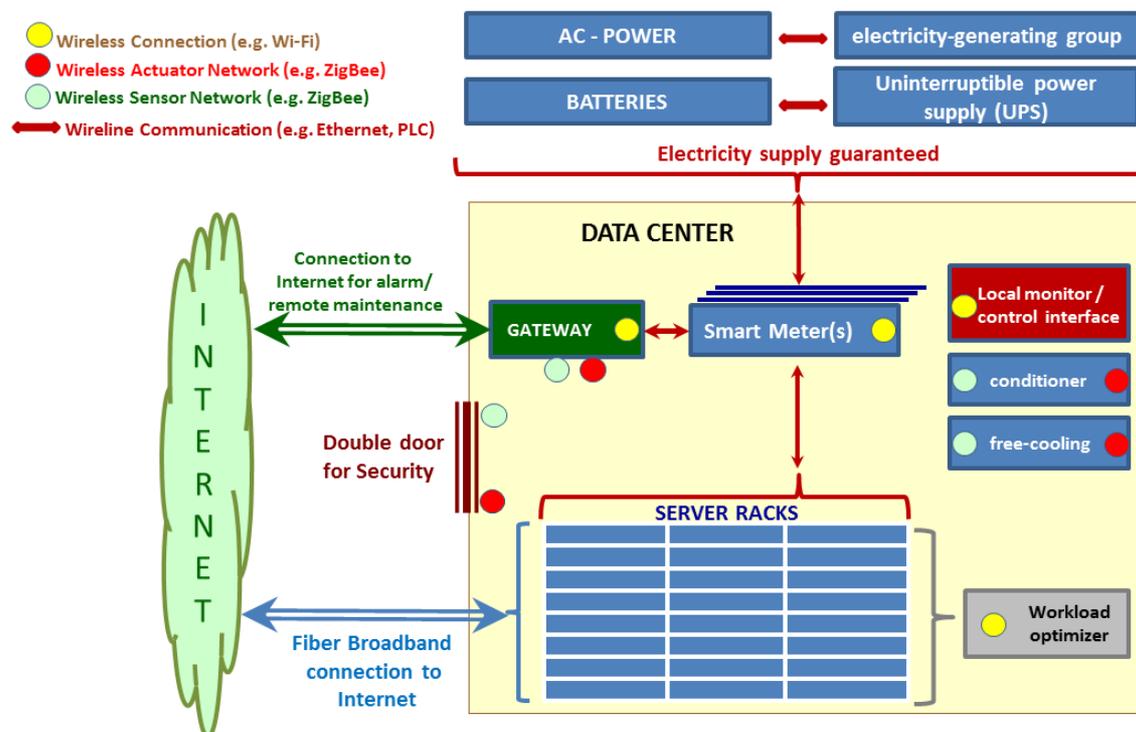


Figure 95: Data Center communication infrastructure

The power metering in real time for each server in the racks is communicated through a wired connection such as Ethernet or PLC (Power Line Communication). Finally, also the communications needed to manage the electricity supply without interruption (even in case of temporary blackout) are exchanged in wired mode.

**8.2.2.5 Hotel communication infrastructure**

Hotel buildings scenario is closely related to residential buildings, with the difference that hotel guests pay a fixed amount of money for a certain level of comfort, and can use as much energy as they may require. Therefore energy efficiency measures such as load shedding must take into account these restrictions and never decrease the level of comfort.

The hotel communication infrastructure shall be fully consistent with the latest industry standards. To enable efficient functional system integration and to provide maximum flexibility and to respond to changes in the building use, the system infrastructure shall support the use of LonWorks, Ethernet TCP/IP and Internet communication technologies. Typically, the system architecture and the relative communication infrastructure for a hotel consist of four levels as shown in Figure 96.

Control Level: it consists of a distributed network of smart control nodes, which are connected to field bus. Nodes shall include all the intelligence of the system. Each node shall be capable of handling several different systems in parallel through flexible distribution of I/O points. Nodes shall be capable of operating autonomously independently of Management Level. For example, all systems must be able to react to alarms on the Control Level without interference from upper levels. All communication shall be event based.

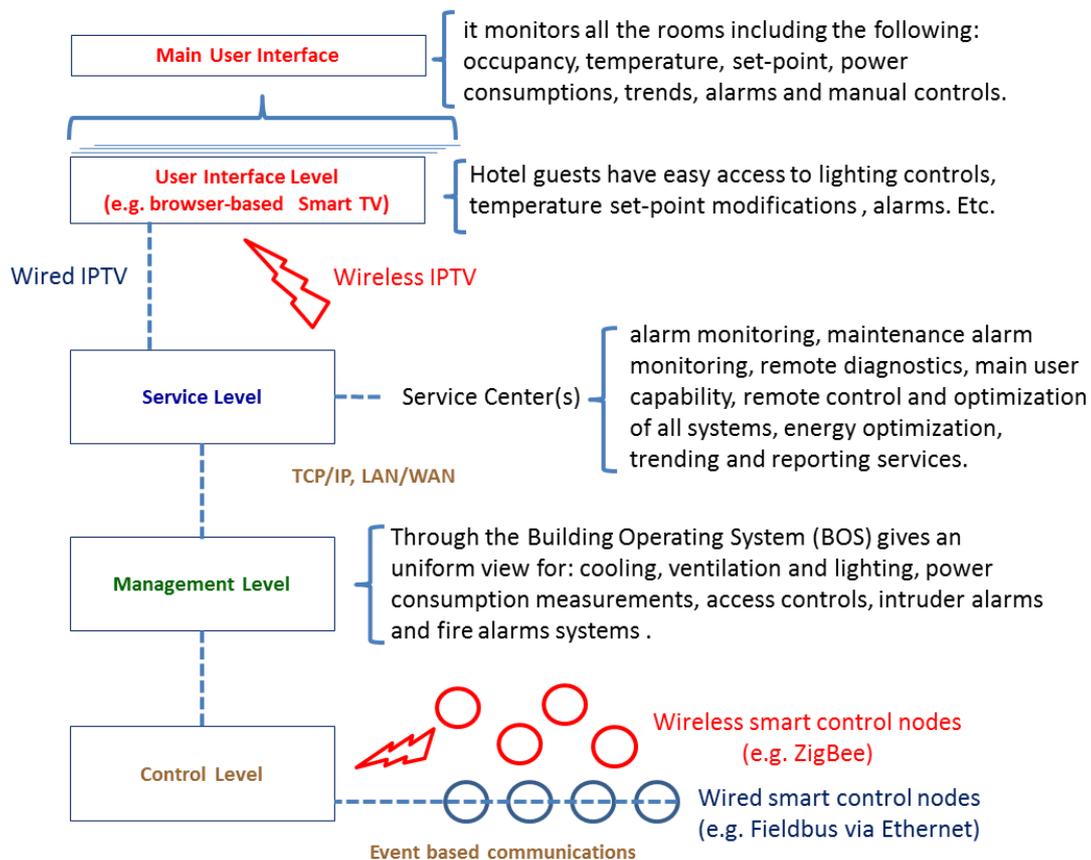


Figure 96: Hotel communication infrastructure

Management Level: it provides a uniform view to all systems through the Building Operating System (BOS). All the systems controls of cooling, ventilation and lighting, consumption measurements, access controls, intruder alarms, and fire alarms systems shall be integrated with the BOS using device drivers. The BOS shall offer at least the following common services to be used by all connected systems: alarms, historical trending, logs and reporting, user profile and role management.

The Building Operating System (BOS) shall include an open interface for other applications to interact with the connected systems. Communication method between BOS and Client applications shall include at least Java Messaging Service (JMS).

Web interfaces shall be used for light-weight clients, e.g. automatically generated browser-based user interfaces in residences for Panel PC's, PDA's or IPTV. The network technology shall be based on the IT standards, such as TCP/IP, and be compatible with latest LAN/WAN technology.

Service Level: it allows the systems to be connected without additional software to one or several Service Center(s), for providing centralized remote monitoring, alarm and fault detection of connected building management and security systems.

The Service Center will be capable of accessing remotely the systems through a standard interface through the BOS. The standard connectivity shall enable providing advanced maintenance and security services, such as security alarm monitoring, maintenance alarm monitoring, remote diagnostics, main user capability, remote control and optimization of all systems, energy optimization, trending and reporting services.

The Service Center shall support connectivity of multiple sites in multi-operator environment. Predefined alarms from connected sites – e.g. intruder alarms, dirty filter notifications or leakage alarms, for example – shall appear in the alarm list with a specified priority. Alarms shall be stored in the central database.

Remote diagnostics of site systems and devices shall enable proactive maintenance of technical systems, energy optimization and efficient management of the infrastructure.

Centralized monitoring of all connected sites with main user capability shall enable e.g. set point changes, manual overdrives and camera controls by using the remote connection.

User Interface Level: the guest room user shall be able to use the system easily with a graphical browser-based User Interface, usable in the IPTV with a normal web browser. The User Interfaces shall provide easy access to frequently needed functionality, such as lighting controls, temperature set-point modifications and alarms.

Moreover it shall be capable of showing consumption values, temperature value and FCU set-point. Guest Room Controls & Monitoring system shall be accessed through the BOS.

Main User Interface shall enable monitoring of all rooms including the following: occupancy, temperature, set-point, consumptions, trends, alarms and manual controls.

### **8.2.3 Virtualization and simplification of the communication infrastructure**

The smart grid technologies collect large amounts of data about power production and distribution, perform real-time analysis and deliver results to both decision makers and consumers.

With the massive amounts of data and information expected to flow across utilities with the implementation of smart grid technologies, the data collection and analysis centers at the utilities themselves may become points of failure, resulting in loss of information system network connectivity.

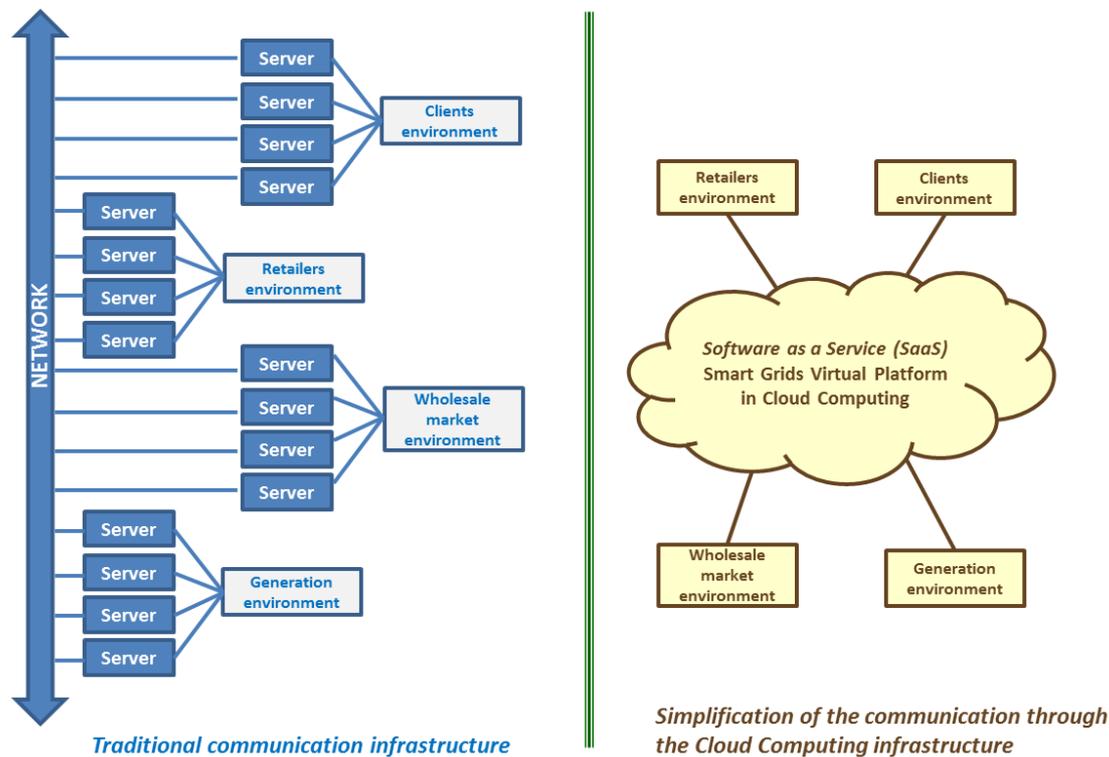


Figure 97: Communication infrastructure simplification by using the Cloud Computing

Today is more and more relevant be able to share information for wide-area and real-time system analysis and visualization. Advanced distributed communications tools can provide the security needed for wide-area situational awareness of the grid network. These communication infrastructure requirements, highlighted in Table 12, are better satisfied if the remote service platform runs in the cloud-computing infrastructure rather than using the traditional client-server model. Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). Cloud computing entrusts remote services with a user's data, software and computation. There are many types of public cloud computing and, in particular, for the demand side manager DSE is typically used the Software as a Service (SaaS), sometimes referred to as "on-demand software". SaaS is a software delivery model in which software and associated data are centrally hosted on the cloud; it is typically accessed by users using a thin client via a web browser.

The development of a virtual cloud-based electric grid information and analysis center, using the “Software-as-a-Service” model, concentrates the flow of communication in a single cloud thus avoiding a dispersion of interconnections typical of the traditional client-server approach (Figure 97). Ultimately, cloud-computing infrastructure will allow for secure virtual information and analysis centers that can distribute results to a large number of clients regardless of their physical locations and with great simplification of the communication infrastructure.

**8.2.4 Communication Infrastructure Requirements**

As explained in sub-chapter 8.2.2, an essential development of the Smart Grid is to extend communication throughout the distribution system and to establish two-way communications with customers through Neighborhood Area Networks (NANs) covering the areas served by distribution substations. In Table 12 are shown the communication requirements as function of the three basic functions in the Smart Grids:

- Energy Trading between producers, retailers and final Customers
- “Supervisory Control” DSE toward the Final Customer
- Network communications among all the Smart Grids actors

FUNCTIONS	OPERATIONS	EXPLANATIONS	ACTIONS	COMMUNICATION REQUIREMENTS
Energy Trading between producers, retailers and final Customers	Forecasting	The energy trading enables a utility to buy energy to meet peak demand or to sell excess capacity.	The Smart grid provides real-time demand and generation information for energy-trading decisions.	Real-time communications between analytics, demand, and generation units is necessary for effective decision making regarding energy trading.
	Market modeling			
	Demand-response programs			
	Risk management			
“Supervisory Control” DSE toward the Final Customer	Final Customer Energy Management (EM)	EM helps Final Customer to Monitor and control the time, amount, type, and level of energy usage.	(EM) data can travel on the Neighborhood area network when smart grid is employed.	Low-cost backhaul communications Methods are necessary for (EM) traffic.
	Metering	Meter data is used for control and billing purposes.	Advanced Metering Infrastructure (AMI) allows for remote meter reads, connects and disconnects an automated outage detection.	Scalable, reliable, low-cost backhaul communications methods are necessary for AMI traffic.
	Demand-Side Management (DSM)	Management of demand-side load.	Smart Grid enables sophisticated demand side managed by integrating EM, AMI data with demand response techniques.	DSM needs reliable communications between EM equipment, AMI, and generation units.
Network communications among all the Smart Grids actors	Communication between regional coordinators	Coordination is needed for better information flow between grids for fault isolation and prevention of fault cascading.	Smart grid enables communication of real-time data between regional control centers.	Secure communications are needed for ICCP (Inter-Control Center Protocol) infrastructure.

Table 12: Communication infrastructure requirements

These communication infrastructure requirements have to be applied in all the five specific domains shown in the Figure 90.

## 9 Security

### 9.1 General approach

Section 9 describes smart domain-specific and use case specific security measures based on the identified security requirements. Security requirements for the FINSENY use cases have been derived based on a threat and risk analysis available in internal report IR 1.4. The substructure of this section reflects these requirements. It is assumed that most of the security measures will not be use case specific in terms of the applied technology. The technology is rather expected to be domain specific or general. Hence, only adaptations may need to be described here. The security measures applied will use security architecture elements as defined in D1.11. Note that D1.11 elaborates on domain specific (Smart Grid) security architecture elements, which may either be defined in the Smart Grid domain or may be adaptations of existing generic security enabler, which are provided by FI-WARE.

### 9.2 Applied Security Technology

The following table summarizes the security requirements as they apply to the building blocks of the architecture defined. Based on this table there are use case specifics discussed in the following subsections. For all requirements, which can be addressed using either domain specific or generic security counter measures, D.11 is referenced.

“S” denotes security requirements that are specific for the corresponding building block, while “X” corresponds to the case when generic security requirements apply to this building block.

Security Requirement \ Building Block	1: Authentication and authorization	2: Data confidentiality	3: Data integrity	4: Non-repudiation	5: Data backup and recovery	6: System protection components	7: Secure SW/FW Updates	8: Secure Network Design	9: Security Management	10: Logging and Audit	11: Time Synchronization	12: Observation of Policies & Laws	13: Transaction Security
Building Energy management controller	S <sup>3</sup>	S <sup>4</sup>	X	X	X	S <sup>3</sup>	X	X	X	X	X	X	X
Building supervisory control service	S <sup>5</sup>	S	X			S <sup>3</sup>	X	S <sup>3</sup>	S <sup>3</sup>	X	X	X	X
Building Entity virtualization Service	S	S	X	X	X	X	X	X	X	X	X	X	X
Building equipment proxy	X	X	X	X	X	X	S <sup>3</sup>	X	X	X	X	X	S <sup>3</sup>
Building space proxy	S	S	X	X	X	X	X	X	X	X	X	X	X

<sup>3</sup> For residential buildings

<sup>4</sup> For residential buildings

<sup>5</sup> For Data Centers

Security Requirement \ Building Block	1: Authentication and authorization	2: Data confidentiality	3: Data integrity	4: Non-repudiation	5: Data backup and recovery	6: System protection components	7: Secure SW/FW Updates	8: Secure Network Design	9: Security Management	10: Logging and Audit	11: Time Synchronization	12: Observation of Policies & Laws	13: Transaction Security
Legacy systems interface	X	SX	X	X	X	X	X	X	X	X	X	X	X
Building sensors	S	S	X	X	X	S <sup>3</sup>	X	X	X	X	X	X	X
Building actuators	S	X	X	X	X	S <sup>3</sup>	X	X	X	X	X	X	X

Table 13 : Matching of generic security requirements with smart buildings architecture building blocks

### 9.3 Relevance of security requirements to WP4

In this section, we go through the security requirements defined in IR1.4 and discuss whether there are WP4 specific requirements.

#### 9.3.1 Authentication and authorization

Specific requirements arise here for local control of energy management system inside residential buildings, where traditional one-size-fits-all security measures should not be imposed on users in their own home, on top of existing “physical” security such as door locks and keys. Users should be considered to implicitly have the required credentials once they are physically inside the building, which means that contextual security, in this case using location itself, in a set-based model, should be taken as equivalent to a classical security credential. If dedicated physical controls or a specific interface (desktop, tablet, control panel located inside the building) are used, this can be linked directly to the use of the corresponding interfaces. If a mobile device is used, location should be strictly identified as corresponding to being inside the building. In this case it is the location that needs to be authenticated, which requires both a non-spoofable location-determination technology and a secure location management system along the whole chain

This specificity does not apply entirely to non residential buildings, where the building energy management system would be operated by a technical staff. In this case classical security measures would be considered appropriate. Only the non-privileged use of local controls (e.g.; the thermostat in an office) may be relaxed to contextual security.

For data centers, the situation is the opposite and security requirements are more stringent than in other types of buildings. It is preferable to make use of two access modes: besides the traditional access card, an additional biometric identification (fingerprint scanners or retinal scanning) is requested. For the most secure part of the data center see the next requirement.

Before to enter in the most secure part of the data center the authorized person has to be authenticated more times along the path, with increasingly stringent investigations, until at the computer processing room, where servers, mainframes or other critical IT equipment are located<sup>6</sup>.

<sup>6</sup> See section 6.2 of D1.11

### 9.3.2 Data confidentiality

Specific requirements arise here mostly as concerns user privacy for residential buildings<sup>7</sup>. In principle, any detailed data relayed outside the building containing possibly sensitive information about individual users (e.g. about the use of various appliances) should be anonymized before being transmitted. The general principle of “separation of concerns” highlighted in the overall FINSENY smart grid interface means that in normal operation data about what happens inside the building should not need to be transmitted outside the building, as would be the case if for example individual appliances or equipment were controlled directly from the microgrid or distribution network for demand-side management purposes.

For non residential buildings confidentiality requirements relate more to the protection against intrusion and theft than to privacy.

### 9.3.3 Data integrity

This relates mostly to potential tampering of metering or that could be altered or tampered with for fraudulent billing purposes. These requirements are standard for such type of data.

### 9.3.4 Non-repudiation

No specific requirement

### 9.3.5 Data backup and recovery

No specific requirement

### 9.3.6 System protection components

Specific requirements arise from the coupling and merging of the security measures that would normally be used for individual subsystems in a traditional building where these are vertically integrated and each have e.g. their own authentication

In a legacy building, a set of vertically integrated subsystems would usually each use their own security measures, from top to bottom if they do not share any infrastructure, as is usually the case. This would seem to be favorable to security, because compromising the security of one subsystem does not mean compromising the security of others (e.g.; finding the key for the control of the heating does not mean you can control elevators or door locks). This is however, akin to “security by obscurity” arguments. An integrated system should not compromise security as state of the art measures (e.g. multifactor authentication for a non-residential building) can be used at the scale of the overall system whereas an isolated legacy subsystem would most probably use a legacy single-factor authentication.

This could also be true for a residential buildings even if, as said before, traditional security could (and generally should) be relaxed in favor of contextual security in this case: the overall security could be biometric (e.g. 3D face recognition) and replace both traditional authentication by possession (door keys) for physical access to the building and authentication by knowledge (passwords) for access to information systems once the home inhabitant is allowed inside<sup>8</sup>.

### 9.3.7 Secure SW/FW Updates

For most data centers the vast majority of traffic is innocent, so the ability for a system to identify and act on only real attacks frees up tremendous people and processing resources for taking appropriate and proactive actions when real attacks occur. So, it is fundamental to upgrade the SW control systems in order to face the rise of mutating attacks and sophisticated evasions (*including IP fragmentation, SQL injection and polymorphic attacks*).

---

<sup>7</sup> See section 6.6 of Deliverable 1.11 for “Security technologies to protect customer privacy in Smart Grids”

<sup>8</sup> Human to device authentication is discussed in D1.11 section 4.4.2.2

**9.3.8 Secure Network Design**

Today’s data center may use more than 130 protocols and services. Protocols not understood by network intrusion prevention systems can become vectors for attacks. When selecting a data center intrusion prevention system, match up the protocols and services supported by considered solutions with the protocols and services running in that specific data center.

**9.3.9 Security Management**

One of the most significant requirements of data center security is the ability to provide proactive protection without disrupting services. Data center intrusion prevention systems need to be accurate and intelligent enough to understand an exploit and what it is targeting and then take an appropriate response (based on vulnerability, protocol and nature of attack) capable of protecting the server or VM without disrupting the session.

**9.3.10 Logging and Audit**

No specific requirement

**9.3.11 Time Synchronization**

No specific requirement

**9.3.12 Observation of Policies & Laws**

No specific requirement

**9.3.13 Transaction Security**

In the data center is also important to combine the security of transactions with a low latency. So the data center intrusion prevention requires exception-based architectures that are able to focus processing resources on real exploits for maximum protection, not treat all traffic to a one size fits all inspection. Systems that can efficiently parse out innocent traffic and focus on exploits (because of accuracy, protocol fluency, appropriate response capabilities, for example) can perform at much higher levels, producing much less latency and use less processing resources.

**9.4 Specific requirements for data center buildings security**

Data centers are created and secured in a physical and also in a logistical way, to ensure the security of the stored data and equipment. The security represents the most important factor for the data centers. Their target is to manage and run different applications in a way that the business for which they are working to access data files efficiently.

Normally, every type of company has a IT security policy, which is managed by organizations, to control the main information services, such as the Internet connectivity, intranets, LANS, WANS and extranets.

For these reasons, in contrast to the concept of security for a residential building, in the case of the datacenter is necessary to draw a distinction between the “*Physical Security*” requirements and the “*Functional Security*” requirements, as shown in Table 14.

DATA CENTER PHYSICAL SECURITY	
Security requirements	Description
Appropriate wall structure	Use walls lined with Kevlar, with a thickness of at least 30cm: it is a concrete and cheap barrier against the elements and explosive devices.
Barriers at vehicle	Control access to the parking lot and loading dock with a staffed guard

entry points	station that operates the retractable bollards.
Intensive use of surveillance cameras	They must be installed around the perimeter of the building, at all entrances and exits, and at every access point throughout the building. A combination of motion-detection sensors, low-light cameras, cameras with high zoom and standard fixed cameras is recommended. Footage should be digitally recorded and stored offsite.
Availability of redundant utilities	Any utility ( <i>electricity, water, voice and data</i> ) must have at least redundancy two, i.e. coming from at least two different main lines. These lines should be underground and should come into different areas of the building, with water separate from other utilities.
Redundancy of UPS & Backup generators	UPS ( <i>Uninterruptible Power Supply</i> ) redundancy should take into account the amount of total power required to operate the data center and the length of time required to get the backup generator into service. It is fundamental to include the HVAC and emergency lighting power requirements in the power autonomy calculations.
Redundancy of HVAC.	HVAC ( <i>Heating, Ventilation, and Air Conditioning</i> ) should be redundant. They are fundamentals otherwise it is useless to have power and network connectivity if it has to shut down the servers because they can't be kept cool. As detailed in FINSENY D4.1, critical to the operation of the data center, the air temperature, humidity level, and the health of the cooling systems must be monitored using sensors and actuators.
High quality electrical supply	The electronic equipment that the data center hosts can be extremely sensitive to "dirty" power. "Dirty" power is that which has high frequency noise in the line, varying voltages, surges, and other electrical impurities. These electrical impurities can disrupt and even ruin sensitive electronic equipment. The electrical system should be tested for quality of power. If not found to be within acceptable tolerances, power filters can be installed to "clean" the power and protect the data center equipment.
Explosive detection	The guards have to use mirrors to check underneath vehicles for explosives, or provide portable bomb-sniffing devices
Secure air handling	Make sure the heating, ventilating and air-conditioning systems can be set to recirculate air rather than drawing in air from the outside. This could help protect people and equipment if there were some kind of biological or chemical attack or heavy smoke spreading from a nearby fire. For added security, put sensors in place to monitor the air for chemical, biological or radiological contaminant.
Cross-checking access authentication	It is preferable to make use of two access modes: besides the traditional access card, an additional biometric identification (fingerprint scanners or retinal scanning) is requested. For the most secure part of the data center see the next requirement.
Locking down the data center core with security layers.	Before to enter in the most secure part of the data center the authorized person has to be authenticated more times along the path, with increasingly stringent investigations, until at the computer processing room, where servers, mainframes or other critical IT equipment are located.

Protect the building's machinery	Keep the mechanical area of the building, which houses environmental systems and uninterruptible power supplies, strictly off limits. If generators are outside, use concrete walls to secure the area. For both areas, make sure all contractors and repair crews are accompanied by an employee at all times.
Fire protection	The data center should to be equipped with a passive fire suppression system. That is, one that activates automatically with no human intervention. These systems come in two main varieties, chemical suppression systems and sprinkler systems.  There is also a need to have manual gas or sprinkler activation switches, portable fire extinguishers, and floor tile lifters placed throughout the data center. They need to be clearly marked and unobstructed.
<b>DATA CENTER FUNCTIONAL SECURITY</b>	
<b>Security requirement</b>	<b>Description</b>
Use of advanced detection systems to avoid “false positive attacks”.	For most data centers the vast majority of traffic is innocent, so the ability for a system to identify and act on only real attacks frees up tremendous people and processing resources for taking appropriate and proactive actions when real attacks occur. So, it is fundamental to upgrade the SW control systems in order to face the rise of mutating attacks and sophisticated evasions ( <i>including IP fragmentation, SQL injection and polymorphic attacks</i> ).
Match up the resident protocols and services supported	Today’s data center may use more than 130 protocols and services. Protocols not understood by network intrusion prevention systems can become vectors for attacks. When selecting a data center intrusion prevention system, match up the protocols and services supported by considered solutions with the protocols and services running in that specific data center.
Ability to provide proactive protection without disrupting services.	One of the most significant requirements of data center security is the ability to provide proactive protection without disrupting services. Data center intrusion prevention systems need to be accurate and intelligent enough to understand an exploit and what it is targeting and then take an appropriate response (based on vulnerability, protocol and nature of attack) capable of protecting the server or VM without disrupting the session.
Use of exception-based architectures.	Data center intrusion prevention requires exception-based architectures that are able to focus processing resources on real exploits for maximum protection, not treat all traffic to a one size fits all inspection. Systems that can efficiently parse out innocent traffic and focus on exploits (because of accuracy, protocol fluency, appropriate response capabilities, for example) can perform at much higher levels, producing much less latency and use less processing resources.

Table 14: Physical and functional security requirements for a data center.

## 9 Conclusion

This document has proposed an architecture specification that may seem very far removed from an actual ICT architecture instance for one particular building. This genericity is not a liability of this architecture; it is its defining characteristic. The architecture specified in this document has been intended as a common basis, not only for widely different types of buildings (from possibly, individual homes to office buildings, hotels and data centers), but also for a full range of ICT applications operating on the hardware of these buildings. Among these, building energy management, i.e. managing a building as a smart grid endpoint, is but one smart building application, sharing a common infrastructure with other smart building applications that will have access to the same interfaces to control and monitor the building hardware.

We have presented a comprehensive and generic smart building infrastructure, aligned with both the Smart Grid Architecture model and the FI-ware IoT framework, especially the device vs. thing/entity distinction. We have presented a range of home & building energy management solutions that may operate on top of a smart building.

Contrary to microgrids or electric vehicle management systems, the domain of buildings does not start from a clean slate. The present situation of building management systems, with a juxtaposition of closed, vertically integrated dedicated systems, is almost at the opposite end of the spectrum from what is proposed here. A full implementation of this architecture, even in a few select instances, remains a long-term goal, one could even say even an ideal. Prototypes for new buildings can be envisioned in a shorter term, but in this domain, a proposal that would apply only to new buildings would be totally useless.

It is only realistic to assume that, for buildings to become smart and get connected to the smart grid, existing systems will *not* be scrapped and buildings will *not* be retrofitted from scratch. This means a smart buildings architecture should deal with legacy building plant *at all relevant levels*, from hardware to communication to software, in a flexible best-effort way. The approaches used for state of the art software infrastructure are too brittle and ill-adapted for this as they work in an all-or-nothing way: either an entity type or an interface is known and it can get integrated, or it is unknown and nothing can be done with it. Integration of legacy systems and legacy hardware should occur with incremental approximation, taking in all the available information about the relevant entities and matching them with relevant models.

Sharing infrastructure and integrating legacy remain the biggest challenges for future projects to address in this domain.

## References

- [1] FINSENY deliverable D4.1 "Smart Buildings Scenario Definition".
- [2] FINSENY deliverable D4.2 "Coarse grain functional architecture"
- [3] SGCG Report Reference Architecture v1.1, 2011-12-12
- [4] FI-WARE Product Vision: <http://tinyurl.com/7zpu87t>  
(full URL: [https://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FI-WARE\\_Product\\_Vision](https://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_Product_Vision))
- [5] FINSENY deliverable D2.1 "Distribution Network Building Block".
- [6] FINSENY deliverable D3.1 "Microgrid Scenario Building Blocks"
- [7] OSGi: Open Services Gateway initiative: <http://www.osgi.org>
- [8] FINSENY deliverable D8.2 "Experiments and evaluation"
- [9] FINSENY deliverable D8.3 "Selected domain specific enablers specification"
- [10] BeyWatch project: <http://www.beywatch.eu>
- [11] Equinox OSGi implementation <http://www.eclipse.org/equinox/>
- [12] HGI: Home Gateway Initiative: <http://www.homegatewayinitiative.org/>
- [13] OSGi Alliance: <http://www.osgi.org/>
- [14] Reactivhome project <https://reactivhome.rd.francetelecom.com/>

## Index of Figures

Figure 1 : Layers, domains and zones of the SGAM Framework .....	11
Figure 2: Smart Buildings external interfaces with other FINSENY domains and the Energy ecosystem.	19
Figure 3: Prosumer - Weather dependent DG (e.g. PV).....	20
Figure 4: Prosumer – Weather Independent DG & Storage (e.g. Batteries & Small Scale Bio mass Units) .....	20
Figure 5: The FINSENY smart building architecture framework .....	25
Figure 6 : Matching of Smart Building Architecture to SGAM framework.....	25
Figure 7 : Generic functional building blocks of FINSENY Smart Building Architecture .....	27
Figure 8 : Start-up and discovery procedure.....	30
Figure 9 : Configuration of energy, power, and price reporting procedure. ....	31
Figure 10 : Configuration of instantaneous power reporting on appliances. ....	31
Figure 11 : Visualization of price associated to a power profile .....	32
Figure 12 : E@H control disabled: example of sequence diagram with user interaction. ....	33
Figure 13 : E@H control disabled: example of sequence diagram.....	34
Figure 14 : E@H control disabled: Overload warning. ....	35
Figure 15: E@H control enabled: example of sequence diagram with user interaction.....	36
Figure 16 : E@H control enabled: sequence diagram without user interaction.....	37
Figure 17 : E@H control enabled: sequence diagram of reactive control (overload management). ....	38
Figure 18 : Use Primary and Self-Production meters in E@H. ....	39
Figure 19 : Management of Self-Production and primary meter in Energy@home. ....	40
Figure 20: Appliance Management Framework – Global view.....	42
Figure 21: Appliance Management Framework – Base drivers .....	42
Figure 22: Appliance Management Framework – Manufacturers’ drivers.....	43
Figure 23: Appliance Management Framework – BeyWatch appliances’ drivers .....	43
Figure 24: Data Center dashboard application .....	45
Figure 25: Application Management components.....	47
Figure 26: Old and new DCIM approach .....	47
Figure 27: Control Layers description.....	50
Figure 28: Shift of temporary services .....	53
Figure 29: Global architecture of the mixed solving system .....	54
Figure 30: Solving process during one step.....	55
Figure 31: Solution found by the solver .....	56
Figure 32: State model of the washing machine service agent.....	57
Figure 33: Solving algorithm in the agent .....	58
Figure 34: Parameters of a profile .....	59
Figure 35: Optimization using branch and bound .....	59
Figure 36: The component of the mixed solving system.....	60
Figure 37: Structure of services.....	63
Figure 38: Building State Maintainer service. ....	65

Figure 39: Historization service Data Mining mode .....	66
Figure 40: Supervisory control in service layer.....	67
Figure 41: Conceptual view of physical entity “shadowing” in the building abstraction layer.....	69
Figure 42: Sensor & actuator Interface and their application domains.....	71
Figure 43: Functional layers in TEDS interface .....	72
Figure 44: Application model of IEEE 1451 interface .....	73
Figure 45: Family of IEEE P1451 Standards .....	74
Figure 46: Conceptual view of IEEE 1451.1.....	75
Figure 47: IEEE 1451 logical interface .....	75
Figure 48: Actuator Sensor Interface architecture.....	76
Figure 49: Actuator Sensor Interface operation.....	77
Figure 50: How an Actuator Sensor Interface performs an operation.....	77
Figure 51: EEBus interface layers .....	78
Figure 52: Comparison between standardization architecture and EEBus. ....	79
Figure 53: The EEBus architecture and interface layers .....	80
Figure 54: Component architecture for smart buildings.....	81
Figure 55: Energy Box as WSan interface. ....	82
Figure 56: WSan topology (a) and abstraction of control application (b) .....	83
Figure 57: The three blocks of the Smart Gateway .....	83
Figure 58: Interface layer of the 6LoWPAN gateway.....	84
Figure 59: Message flows between IPv6 hosts, gateway and 6LoWPAN wireless sensor network.....	85
Figure 60: FI-WARE Chapters.....	88
Figure 61: Mapping of FI-WARE GEs to functional building blocks of the FINSENY architecture .....	93
Figure 62: IoT Node Service Stack .....	95
Figure 63: IoT Communications GEs.....	96
Figure 64: IoT Resource Management GEs .....	96
Figure 65: IoT Process Automation GEs.....	97
Figure 66: IoT Data Handling GEs.....	97
Figure 67: CDI GE .....	98
Figure 68: IoT Gateway aggregating devices functionality from devices with/and without computing capabilities.....	98
Figure 69: Architecture of IoT Communications ( <i>framed in red the Front-end GE</i> ).....	99
Figure 70: Complex Event Processing (CEP) GE .....	101
Figure 71: EPN Definition Example .....	101
Figure 72: Big Data Analysis GE.....	102
Figure 73: Interface with Marketplace and Grid .....	102
Figure 74: Example of a Marketplace & Grid specific CEP deployment scenario.....	103
Figure 75 Example ontology of building entities to be represented in building abstraction layer.....	110
Figure 76 : Hybrid Discrete State model examples for home appliances.....	111
Figure 77: SGAM communication layer and OSGP/OSI protocols .....	112
Figure 78: Topological segmentation in WAN, NAN and HAN.....	116

---

Figure 79: WAN mapping: communication protocols and network technology .....	118
Figure 80: NAN mapping: communication protocols and network technology .....	118
Figure 81: HAN/BAN mapping: communication protocols and network technology.....	119
Figure 82: ZigBee Protocol Stack .....	121
Figure 83: Smart Home communication flow .....	121
Figure 84: Residential building communication layers .....	122
Figure 85: BACnet Protocol Stack .....	123
Figure 86: LonWorks Protocol Stack .....	124
Figure 87: Office/Public building communication layers.....	125
Figure 88: Data center communication layers .....	126
Figure 89: Hotel communication layers .....	126
Figure 90: Communication in domain specific infrastructure. ....	135
Figure 91: Home domain communication infrastructure.....	138
Figure 92: BACnet communication layers .....	139
Figure 93: BACnet communication infrastructure .....	140
Figure 94: Office/Public Building domain communication infrastructure .....	141
Figure 95: Data Center communication infrastructure .....	142
Figure 96: Hotel communication infrastructure .....	143
Figure 97: Communication infrastructure simplification by using the Cloud Computing .....	145

## Index of Tables

Table 1: Architectural external interface principles .....	22
Table 2 : Main Building block of the FINSENY Smart Building Architecture .....	28
Table 3: Data Quality identification .....	40
Table 4: Mapping of FI-WARE GEs to functional building blocks of the FINSENY architecture .....	93
Table 5: Communication protocols in smart building applications .....	113
Table 6: Communication protocols in the world .....	117
Table 7: WAN protocol characteristics .....	127
Table 8: NAN protocol characteristics .....	128
Table 9: HAN protocol characteristics .....	129
Table 10: Communications protocol constrains .....	130
Table 11: Component overview .....	132
Table 12: Communication infrastructure requirements .....	146
Table 13 : Matching of generic security requirements with smart buildings architecture building blocks .....	148
Table 14: Physical and functional security requirements for a data center. ....	152

## Acronyms and Abbreviations

<b>ADSL</b>	Asymmetric Digital Subscriber Line
<b>API</b>	Application Program Interface
<b>ATM</b>	Asynchronous Transfer Mode
<b>BACnet</b>	Building Automation and Control Networks data communication protocol
<b>BAL</b>	Building Abstraction Layer
<b>BAN</b>	Building Area Network – synonym for HAN
<b>BEMS</b>	Building Energy Management System
<b>BPL</b>	Broadband over Power Line
<b>CDMA</b>	Code Division Multiple Access
<b>CIM</b>	Common Information Model
<b>CIS</b>	Customer Information System
<b>CPS</b>	Combined Photovoltaic System
<b>DER</b>	Distributed Energy Resources
<b>DMS</b>	Distribution Management System
<b>DNP</b>	Distributed Network Protocol
<b>DNP3</b>	(Distributed Network Protocol) is a set of communications protocols used between components in process automation systems.
<b>DOCSIS</b>	Data Over Cable Service Interface Specification
<b>DRbizNet</b>	Demand Response Business Network
<b>DSE</b>	Domain Specific Enabler
<b>DSO</b>	Distribution System Operator
<b>ebXML</b>	Electronic Business using eXtensible Markup Language
<b>EMS</b>	Energy Management System
<b>FAN</b>	Field Area Network
<b>GE</b>	Generic Enabler
<b>GID</b>	Generic Interface Definition
<b>GPRS</b>	General Packet Radio Service
<b>GPT</b>	General Purpose Technology
<b>HAN</b>	Home Area Network
<b>HGI</b>	Home Gateway Initiative
<b>HVAC</b>	Heating, Ventilation, and Air Conditioning
<b>ICCP</b>	Inter Control Center Protocol
<b>IEC 60870</b>	Standards that define systems used for telecontrol
<b>IEC 61850</b>	Is a standard for the design of electrical substation automation.
<b>IEC 61970</b>	Series of standards deals with the application program interfaces for energy management systems (EMS)

---

<b>IEC 61968</b>	Is a series of standards under development that will define standards for information exchanges between electrical distribution systems.
<b>LAN</b>	Local Area Network
<b>MPLS</b>	Multiprotocol Label Switching
<b>OMS</b>	Outage Management System
<b>OSGi</b>	Open Service Gateway initiative
<b>PAN</b>	Premise Area Network – synonym for HAN
<b>PC</b>	Personal Computer
<b>PHEV</b>	Plug-in Hybrid Electric Vehicle
<b>PLC</b>	Powerline Carrier
<b>RES</b>	Renewable Energy System
<b>SAE</b>	Society of Automotive Engineers
<b>SAN</b>	Substation Area Network
<b>SCADA</b>	Supervisory Control and Data Acquisition
<b>SGAM</b>	Smart Grid Architecture Model
<b>SOAP</b>	Simple Object Access Protocol
<b>SONET</b>	Synchronous Optical Networking
<b>WAN</b>	Wide Area Network
<b>WDM</b>	Wave-division Multiplexing
<b>WiFi</b>	is a mechanism that allows electronic devices to exchange data wirelessly over a computer network using the IEEE 802.11 family of standards.
<b>WiMAX</b>	Worldwide Interoperability for Microwave Access
<b>Wireless mesh</b>	Is a communications network made up of radio nodes organized in a mesh topology.
<b>WMS</b>	Work Management System
<b>ZigBee</b>	is a specification for a suite of high level communication protocols using small, low-power digital radios based on an IEEE 802 standard for personal area networks