Small or medium-scale focused research project (STREP)

Energy Demand-Aware Open Services for Smart Grid Intelligent Automation

SmartHG

EU FP7 Project #317761

Deliverable D3.2.2

Second Year Prototype of Home Intelligent Automation Services

Deliverable due on	:	M26
Output of WP	:	WP3
WP Responsible	:	HMTI

Consortium

Participant Organization Name	Participant Short Name	Country
Sapienza University of Rome	UNIROMA1	Italy
Aarhus University	AU	Denmark
IMDEA Energía	IMDEA	Spain
A. V. Luikov Heat and Mass Transfer Institute of the National Academy of Sciences of Belarus	HMTI	Belarus
ATANVO GmbH	ATANVO	Germany
Panoramic Power	PANPOW	Israel
Solintel	SOLINTEL	Spain
SEAS – NVE	SEAS	Denmark
Kalundborg Municipality	KAL	Denmark
Minskenergo	MINSKENG	Belarus
Develco Products A/S	DEVELCO	Denmark

Document Information

Version	June 5, 2015, 15:01
Date	June 5, 2015
Contributors	UNIROMA1, AU, IMDEA, HMTI, ATANVO, PANPOW, DEVELCO
Reason for release	Second year review
Dissemination level	Public (PU)
Status	Final

Project title	Energy Demand-Aware Open Services for Smart Grid Intelligent Automation
Project acronym	SmartHG
Project number	317761
Call (part) identifier	FP7-ICT-2011-8

Work programme topic addressed	
Challenge	**6**: *ICT for a low carbon economy*
Objective	**ICT-2011.6.1** *Smart Energy Grids*
Target Outcome	**d**) Home energy controlling hubs that will collect real-time or near real-time data on energy consumption data from smart household appliances and enable intelligent automation.

Project coordinator	Enrico Tronci
E-mail	`tronci@di.uniroma1.it`

To make this deliverable suitable for public dissemination, test-bed data have been anonymized.

Contents

List of Figures

List of Tables

List of Acronyms

COP Coefficient of Performance

CSV Comma Separated Value

DAPP-H Demand Aware Price Policies for Homes

DAPP-K Demand Aware Price Policies for Substation-Level Energy Storage Control

DB&A Database and Analytics

DBService Database Service

DSO Distribution System Operator

EBR Energy Bill Reduction

EDN Electric Distribution Network

ESS Energy Storage System

EUMF Energy Usage Modelling and Forecasting

EUMF-H Energy Usage Modelling and Forecasting for Homes

EUMF-K Energy Usage Modelling and Forecasting for Control

EUR Energy Usage Reduction

EUR-H Energy Usage Reduction for Homes

EUR-K Energy Usage Reduction for Control

GIAS Grid Intelligent Automation Service

HECH Home Energy Controlling Hub

HIAS Home Intelligent Automation Service

IAS Intelligent Automation Service

IBR Inclining Block Rate

MILP Mixed-Integer Linear Programming

PEV Plug-in Electric Vehicle

RESTful REpresentational State Transfer

RPi RaspberryPi

SaaS Software as a Service

ToU Time of Usage

Executive Summary

The main objective of the SmartHG project is to develop effective Intelligent Automation Services (IASs) to minimise users energy bill for end residential users while optimising operation on the grid for Distribution System Operators (DSOs). This deliverable, together with Deliverable D4.2.2, describes the second year effective implementation of such IASs. Namely, in this deliverable we focus on the IASs working on the residential user side, i.e., on the Home Intelligent Automation Services (HIASs). Instead, Deliverable D4.2.2 focuses on the Grid Intelligent Automation Services (GIASs), which work on the DSO side. Following the design described in Deliverable D3.2.1, the GIASs developed in the second year are the following: the Energy Bill Reduction (EBR), Energy Usage Reduction for Homes (EUR-H), Energy Usage Modelling and Forecasting for Homes (EUMF-H), Energy Usage Reduction for Control (EUR-K) and Energy Usage Modelling and Forecasting for Control (EUMF-K) services. To this aim, first of all we describe the overall architecture of the prototypes, which is the same as the GIASs architecture described in Deliverable D4.2.2. Then, one section is dedicated to each prototype, describing what it does and how it must be used. W.r.t. the first year version of the HIAS prototypes, the second year version of all HIAS prototypes changed not only to reflect the changes in the design (described in Deliverable D3.2.1), but also in the overall architecture. Namely, all HIAS prototypes are now accessible from a Web service, which allows either a Software as a Service (SaaS) usage (submit an execution and wait for completion) or software download (for services with real time requirements). Finally, we also describe the implementation of the Home Energy Controlling Hub (HECH), which (together with the Database and Analytics (DB&A) service) allows IASs to access to measurement data from the residential homes.

The effectiveness of the prototypes described here, which are based on the algorithms described in the HIASs design (Deliverable D3.2.1), is evaluated, together with the GIAS prototypes, in the second year iteration of the SmartHG IASs evaluation (Deliverable D5.2.1). In the third and last project year, the prototypes presented here will be improved by both implementing the new improved algorithms proposed in the future version of the services design, and by tailoring such prototypes to directly work on the project test-beds in Minsk (Belarus), Kalundborg (Denmark) and Central District (Israel). This will allow SmartHG IASs to be effectively released as the major final product of SmartHG project.

Chapter 1

Retrospect

In this section we briefly recall the main achievements obtained in the first year version of the SmartHG Home Intelligent Automation Services (HIASs) prototype implementation.The detailed list of all advancements of the second year version of such prototypes w.r.t. the first year version is instead reported in Section 6.

The first year prototypes for all HIASs to be developed within the SmartHG project, namely Energy Bill Reduction (EBR), Energy Usage Reduction (EUR) and Energy Usage Modelling and Forecasting (EUMF), plus two testing protocols for communication between home smart devices and Intelligent Automation Services (IASs), where developed following the first year specification and algorithms design. They were used to perform the first year evaluation. Nearly all prototypes were developed on Linux using Open Source Software. As a partial exception, EUR and EUMF services where developed on Windows. However, since they are based on Java, they could be easily ported to other platforms.

The major challenges identified for the HIAS prototypes (in common with the Grid Intelligent Automation Service (GIAS) prototypes) is to provide each service in a more user-friendly way, e.g., by setting up a Web service for each HIAS.

Chapter 2

Introduction

This deliverable describes the second year prototypes for all Home Intelligent Automation Services (HIASs) to be developed within the second year iteration of the SmartHG project, namely Energy Bill Reduction (EBR), Energy Usage Modelling and Forecasting for Control (EUMF-K), Energy Usage Reduction for Control (EUR-K), Energy Usage Modelling and Forecasting for Homes (EUMF-H) and Energy Usage Reduction for Homes (EUR-H). The second year version of the algorithms for the above services have been described in Deliverable D3.2.1. Moreover, these prototypes are used for the evaluation phase described in D5.2.1.

Table 2.1 shows the main properties of the prototypes discussed in this deliverable. For all services, a Web service has been developed and deployed. For the EUR-K, EUR-H and EUMF-H services, a Software as a Service (SaaS) Web service has been developed. A user, after authentication, may interact with the corresponding service through such Web service, by uploading inputs, requesting executions and retrieving outputs. For the EBR and EUMF-K services, which have real-time constraints, the Web service only allows to download the software and the documentation, upon authentication.

Prototype	Web	SaaS	Software Download	RESTful
EBR	Yes [1]	No	Yes	Authentication only [2]
EUMF-K	Yes [3]	No	Yes	Authentication only [4]
EUR-K	Yes [5]	Yes	No	Yes [6]
EUR-H	Yes [7]	Yes	No	Yes [8]
EUMF-H	Yes [7]	Yes	No	Yes [9]

Table 2.1: HIAS prototypes synopsis. A Web page collecting all links above is here [10]

2.1 Outline

This deliverable is organised as follows. Section 3 shows the common architecture on which all second year HIAS prototypes relies upon. Then, Section 4 describes, for each HIAS, the Web service interface which has been developed in the second year. As for the Home Energy Controlling Hub (HECH), which works as an intermediate between devices installed in a residential home and the Intelligent Automation Services (IASs), the second year prototype is described in Section 5. The overall results of this deliverable are summarised in Section 6. Furthermore, Section 6 describes in detail the advancements

Table 2.2: Correspondence between SmartHG tasks inside WP3 and sections of this deliverable

Task	**Task Name**	**Sections**
T3.1	Design and Development of Open Standard Internet-based communication between Home Devices and IASs	Section 5
T3.2	Design and Development of Open Standard Internet-based communication between IASs	Section 3
T3.3	Design and Development of home EUMF service	Sections 4.2 and 4.4
T3.4	Design and Development of home EBR service	Section 4.1
T3.5	Design and Development of home EUR service	Sections 4.3 and 4.4

of this year HIAS prototypes w.r.t. the first year prototypes of the same services, discusses the current limitations and plans future work. Finally, Table 2.2 shows the correspondence between SmartHG tasks inside WP3 and sections of this deliverable.

Chapter 3

HIAS Prototypes: General Architecture and Usage

The goal of tasks 3.1, 3.2, 3.3, 3.4 and 3.5 of WP3 is to design, implement and deploy the SmartHG Home Intelligent Automation Services (HIASs), i.e., the services to be proposed to the residential users inside an Electric Distribution Network (EDN). The design of such HIASs is described in Deliverable D3.2.1. In this deliverable, instead, we focus on implementation and deployment. To this aim, analogously to the description of Grid Intelligent Automation Service (GIAS) prototypes in Deliverable D4.2.2, first of all we categorise the HIASs as follows.

Software as a Service (SaaS) These are services which do not have real-time constraints, and thus may be deployed as Web services. Such Web services must allow residential users to upload inputs, request an execution, and download (and visualise, if applicable) outputs. The HIASs which fall in this category are Energy Usage Reduction for Control (EUR-K), Energy Usage Reduction for Homes (EUR-H) and Energy Usage Modelling and Forecasting for Homes (EUMF-H).

Real-time services These are services which control some device (or serve as an auxiliary service to control some device). The main real-time HIASs is the Energy Bill Reduction (EBR) service, which every t minutes (1 hour in the evaluation shown in Deliverable D5.2.1) must compute a charge/discharge action for an Energy Storage System (ESS) and a Plug-in Electric Vehicle (PEV) installed on a residential home. Moreover, as a auxiliary service for EBR (and also for the GIAS Demand Aware Price Policies for Substation-Level Energy Storage Control (DAPP-K)), also the Energy Usage Modelling and Forecasting for Control (EUMF-K) service is a real-time service. Real-time services cannot be invoked from a Web interface (as this the real-time requirements will fail), thus they must be downloaded, installed and run from a local host owned by the user.

Given this, we designed the HIAS prototypes to have the same architecture used for the GIAS. Thus, we refer the reader to Deliverable D4.2.2 for a general description of both the architecture and of its use cases.

Chapter 4

HIAS Prototypes User Manual

In this section we discuss in detail each Home Intelligent Automation Service (HIAS) from the users perspective, i.e., i) for the Software as a Service (SaaS) services, we describe the Web service, and ii) for the real-time services, we describe both how to download it and how to launch an execution on a local machine.

4.1 EBR Web Service and Tarball File

In this section, we describe both the Energy Bill Reduction (EBR) Web service [1] and the EBR tarball archive file which may be downloaded from [1]. For the design of the service and the nomenclature used, we refer the reader to Deliverable D3.2.1. As for the Web service, it is necessary to authenticate to a login page (see Figure 4.1). Upon authentication, there is only one available action: to download the EBR tarball file, namely `ebr-1.0.tgz` (see Figure 4.2). Note that the user manual for the EBR tarball file (namely `ebr-1.0.pdf`) is also available for download.

Once `ebr-1.0.tgz` has been downloaded on a Linux machine, it is possible to unzip it by typing on a terminal the following command:

```
tar xfz ebr-1.0.tgz; cd ebr-1.0
```

This creates a new directory named `ebr-1.0`, with the following directories: `aux_files`, `example_files` and `src`. Moreover, an explanatory `README.txt` file is provided, together with a wrapper Bash script `launch.sh`.

In order to run EBR, the main requirement is to have either CPLEX or GLPK installed on the machine (i.e., either the command `cplex` or the command `glpsol` must be available in the system or user `PATH`). Given this, the user may directly launch the Bash script:

```
bash launch.sh
```

This will launch EBR with default settings, which will reproduce the 9 kWh – 2 kW experiment on the Kalundborg scenario described in Deliverable D5.2.1. In order to customise the input, the `launch.sh` script accepts the following command line arguments (note they are all optional: when an argument is not given, the default is used).

`-h`: prints an help message with all arguments and defaults. Des not run EBR.

`-l` h: uses h as the number of hours to be forecasted for each charge/discharge action computation (default is 6).

`-f` f_1: uses f_1 as the Comma Separated Value (CSV) file with the aggregated demand history (default is `example_files/kalundborg/profile.csv`, which may be used as an example for the format).

`-pr` f_2: uses f_2 as the CSV file with the energy tariff for the given home (default is `example_files/kalundborg/pricepolicies.csv`, which may be used as an example for the format when the tariff is Inclining Block Rate (IBR); if instead the tariff is Time of Usage (ToU), then `example_files/central_district/ToU_tariff.csv` may be used).

`-pev` f_3: uses f_3 as the CSV file with the Plug-in Electric Vehicle (PEV) characteristics (default is `example_files/kalundborg/pev.csv`, which may be used as an example for the format).

`-bc` Q: uses Q as the Energy Storage System (ESS) capacity (i.e., maximum energy storage) in kWh (default is 9).

`-br` R: uses R as the ESS maximum charge/discharge rate in kW (default is 2).

`-bde` α: uses $\alpha \in [0, 1]$ as the ESS efficiency coefficient for discharging (default is 0.82). That is, if a discharge action a is sent to the ESS for 1 hour, then the ESS discharges of a kWh, but the energy provided is $a\alpha$ instead of a kWh.

`-bce` β: uses $\beta \in [0, 1]$ as the ESS efficiency coefficient for charging (default is 0.98). That is, if a charge action a is sent to the ESS for 1 hour, then the energy provided to the ESS is a kWh, but the ESS charges of $a\beta$ kWh instead of a kWh.

`-pc` Q_p: uses Q_p as the PEV capacity (i.e., maximum energy storage) in kWh (default is 16).

`-ppr` R_p: uses R_p as the PEV maximum charge rate in kW (default is 13).

`-pce` β_p: uses $\beta_p \in [0, 1]$ as the PEV efficiency coefficient for charging (default is 0.876). That is, if a charge action a is sent to the PEV for 1 hour, then the energy provided to the PEV is a kWh, but the PEV charges of $a\beta_p$ kWh instead of a kWh.

`-cc` C_u: uses C_u as the maximum threshold for power usage in kW (default is 18).

`-cp` P_u: uses P_u as the maximum threshold for power production in kW (default is 6).

`-low` l_{IBR}: uses l_{IBR} as low tariff, in EUR/kWh, when IBR tariff is used (default is 0.1).

`-high` h_{IBR}: uses h_{IBR} as high tariff, in EUR/kWh, when IBR tariff is used (default is 0.3).

`-flat` f: uses f as flat tariff in EUR/kWh (only for comparison, default is 0.2).

`-p` p: uses p as the number of days in the past to be used for forecast (default is 10).

`-pd` p_d: uses p_d as the discounting factor for the days in the past. Format is x_1:...:x_p, and $\sum_{i=1}^{n} x_i = 1$ must hold (default is $\frac{1}{2}$:$\frac{1}{2^2}$:. . .:$\frac{1}{2^9}$:$\frac{1}{2^9}$).

`central_district`: uses the power profile and the ToU tariff for the Central District scenario (overrides options -f, -pr, -bc, -cc).

`ToU`: the price policy is interpreted as a ToU.

`-ndmilps`: do not delete auxiliary files for Mixed-Integer Linear Programming (MILP) problems.

Finally, the output of EBR is the log of the decided actions (for both the ESS and the PEV) and their effect on the resulting home demand. Such output is stored in the CSV file `results/`$\tilde{t}$`/output/results.csv`, being $\tilde{t}$ the time-stamp at which `launch.sh` was started. Namely, `results.csv` contains the following information, for each time-slot in the execution (note that EBR actually starts to compute charge/discharge actions only after $24p$ hours):

- starting and ending time-stamps of the current time-slot t;
- overall home demand $d(t)$ (in kW) without using EBR, i.e., without ESS and where PEV is not managed by EBR;
- ESS action $a_e(t)$ computed by EBR for t (in kW),
- PEV action $a_p(t)$ computed by EBR for t (in kW),
- overall home demand $\tilde{d}(t)$ (in kW) using EBR (that is, after ESS and PEV actions application), i.e., either $\tilde{d}(t) = d(t) + a_e(t) + a_p(t)$ (if $a(t) \geq 0$) or $\tilde{d}(t) = d(t) + a(t)\alpha + a_p(t)$ (otherwise);
- ESS remaining capacity $b_e(t)$ (in kWh), i.e., either $b_e(t) = b_e(t-1) + \tau a(t)\beta$ (if $a(t) \geq 0$) or $b_e(t) = b_e(t-1) + \tau a(t)$ (otherwise), being τ the length of time-slot t;
- PEV remaining capacity $b_p(t)$ (in kWh), i.e., $b_p(t) = b_p(t-1) + \tau a(t)\beta$ (if $a(t) \geq 0$) or $b_p(t) = b_p(t-1) + \tau a(t)$ (otherwise), being τ the length of time-slot t;
- cost of energy $c(t)$ (in EUR/kWh);
- cost of demand without ESS and where PEV is not managed by EBR (in EUR), i.e., $c(t)f$, being f the flat tariff;
- cost of demand with ESS (in EUR), i.e., $c(t)\tilde{d}(t)$.

4.2 EUMF-K Web Service and Tarball File

In this section, we describe both the Energy Usage Modelling and Forecasting for Control (EUMF-K) Web service [3] and the EUMF-K tarball archive file which may be downloaded from [3]. For the design of the service and the nomenclature used, we refer the reader to Deliverable D4.2.1. As for the Web service, it is necessary to authenticate to a login page which is similar to the EBR login page already shown in Figure 4.1. Upon authentication, there is only one available action: to download the EUMF-K tarball file, namely `eumf-k-1.0.tgz` (the corresponding page is similar to the one for EBR, see Figure 4.2).

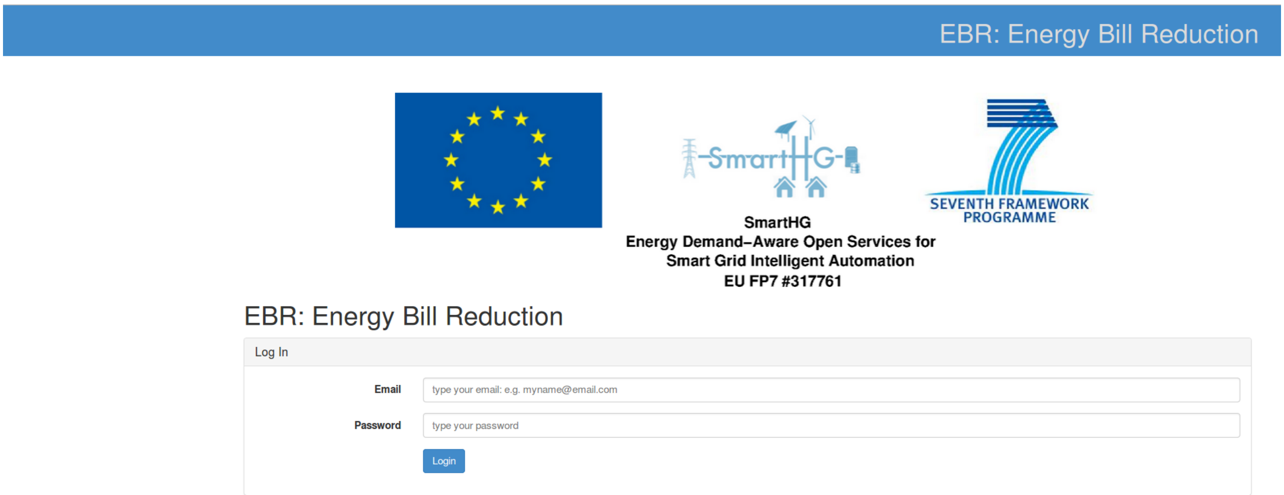

Figure 4.1: EBR service: login page

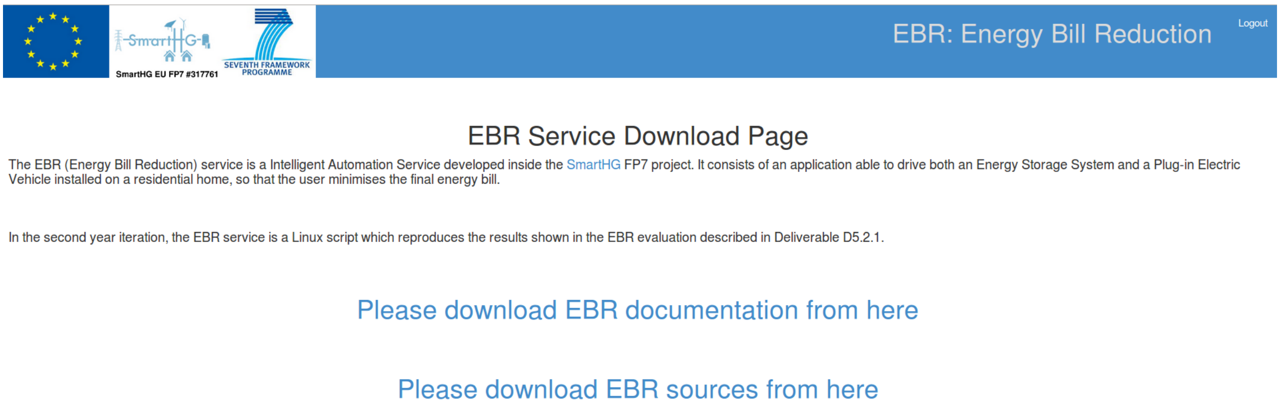

Figure 4.2: EBR service: page for download

Note that the user manual for the EUMF-K tarball file (namely `eumf-k-1.0.pdf`) is also available for download.

Once `eumf-k-1.0.tgz` has been downloaded on a Linux machine, it is possible to unzip it by typing on a terminal the following command:

```
tar xfz eumf-k-1.0.tgz; cd eumf-k-1.0
```

This creates a new directory named `eumf-k-1.0`, with the following directories: `aux_files`, `example_files` and `src`. Moreover, an explanatory `README.txt` file is provided, together with a wrapper Bash script `launch.sh`.

Given this, the user may directly launch the Bash script:

```
bash launch.sh
```

This will launch EUMF-K with default settings, which will reproduce the 500 kWh – 80 kW experiment described in Deliverable D5.2.1. In order to customise the input, the `launch.sh` script accepts the following command line arguments (note they are all optional: when an argument is not given, the default is used).

-h: prints an help message with all arguments and defaults. Des not run EUMF-K.

-l h: uses h as the number of hours to be forecasted for each time-slot in input (default is 24, which is also the maximum value).

-f f: uses f as the CSV file with the power demand history (default is `example_files/profile.csv`, which may be used as an example for the format).

-p p: uses p as the number of days in the past to be used for forecast (default is 10).

-pd p_d: uses p_d as the discounting factor for the days in the past. Format is x_1:...:x_p, and $\sum_{i=1}^{n} x_i = 1$ must hold (default is $\frac{1}{2}:\frac{1}{2^2}:\ldots:\frac{1}{2^9}:\frac{1}{2^9}$).

Finally, the output of EUMF-K is the forecast results. Such output is stored in a CSV file named `results.csv`, which is stored in an automatically created directory named `results/`$\tilde{t}$`/output/`, being $\tilde{t}$ the time-stamp at which `launch.sh` was started. Namely, `results.csv` contains the following information, for each time-slot in the execution:

- starting and ending time-stamps of the current time-slot t;
- power demand $d(t)$ (in kW) from the input, divided in consumption, production and demand (difference of consumption and production);
- for each $i = 1, \ldots, h$, the forecast of consumption, production and demand after i hours. Note that the forecast starts after $24p$ hours.

4.3 EUR-K Web Service

In this section, we describe the Energy Usage Reduction for Control (EUR-K) Web service [5], which shares many features with the Web services of the other SaaS Grid Intelligent Automation Services (GIASs) described in Deliverable D4.2.2. For the design of the service and the nomenclature used, we refer the reader to Deliverable D3.2.1. First of all, the user must authenticate on a login page which is similar to the EBR login page already shown in Figure 4.1. Upon successful authentication, by choosing a link in the left sidebar (the layout is the same as the one in Demand Aware Price Policies for Homes (DAPP-H) service described in Deliverable D4.2.2) the EUR-K Web service allows the user to perform the following actions.

New Execution Request a new execution (see Figure 4.3). In order to submit an execution request, a complete input must be provided. To this aim, the input is organised in the following *scenarios* and parameters:

- The *inside temperature scenario* contains the temperature measurements (in Celsius degrees) taken at different time-stamps inside the home;
- The *outside temperature scenario* contains the temperature measurements (in Celsius degrees) taken at different time-stamps inside the home;
- The *heat pump consumption scenario* contains the power profile of the heat pump, as a sequence of power usage measured in Watts for different time-stamps;
- The *heat pump Coefficient of Performance (COP)*, taken from heat pump specifications;
- The *home outside wall surface* (in m^2).

In order to create a new execution request, it is possible to both choose already uploaded scenarios (i.e., scenarios already present in the REpresentational State Transfer (RESTful) service), or to create new scenarios. If already present scenarios are chosen, it is possible to check the actual profiles contained in the selected scenarios by downloading them as text files. On the other hand, to create new scenarios it is sufficient to upload a file for each scenario. Namely, such file must be CSV file where the first colun is the time-stamp and the second one is the temperature inside the home in that time-stamp (for the inside temperature scenario), the temperature outside the home in that time-stamp (for the outside temperature scenario) or the heat pump energy consumption in that time-stamp (for the heat pump consumption scenario).

Submitted Executions Search, retrieve and check status for already submitted executions (default choice after login). For completed executions, it is possible to download both the complete input and to graph the service output. Such an output is described in Deliverable D3.2.1, and consists of a graph showing predictions on the average daily energy consumption as a function of the overall heat transfer coefficient (U-value). The U-value inversely depends on the thermal insulation and on the total house external surface. The output graph shows one curve for each meaningful value of heat pump COP. EUR-K output is a decision support for users, as explained in Deliverable D3.2.1.

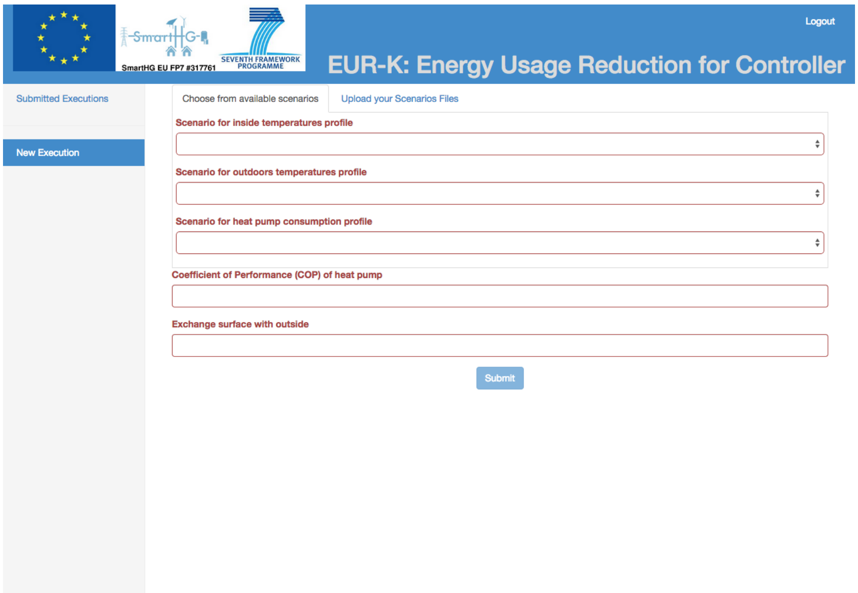

Figure 4.3: EUR-K service: requesting a new execution

Note that, for second year iteration of EUR-K, input scenarios are requested to the user. In order to perform evaluation, we have downloaded such values from Panoramic Power Dashboard for interested houses of our Svebølle test-bed. In order to improve usability of this service, in the next iteration of EUR-K we plan to add a new feature, allowing to directly download input scenarios from Database and Analytics (DB&A) (storing sensors data) or from Panoramic Power Dashboard.

4.4 EUR-H/EUMF-H Prototypes

In this section, we describe the common Web interface used for the Energy Usage Reduction for Homes (EUR-H) and Energy Usage Modelling and Forecasting for Homes (EUMF-H) prototypes. Namely, the Web service for EUR-H and EUMF-H allows to select, on the left sidebar, one home among the ones in the Minsk or in the Kalundborg test-bed. Homes selection is made easier by considering home addresses (see Figure 4.4). Once an home is selected, the following tags are available (see Figure 4.5):

History This allows the user to choose a date in the past and visualise the historical demand in that date, divided in consumption and production. Efficiency for heat pump, when available, is also shown.

Forecast This allows the user to choose a forecast interval and visualise the forecasted demand from the current date to the end of the selected interval, divided in consumption and production. Efficiency for heat pump, when available, is also shown.

In both cases, the data are graphically depicted, as shown in Figure 4.6.

Figure 4.4: EUR-H and EUMF-H Web service: selecting an home.

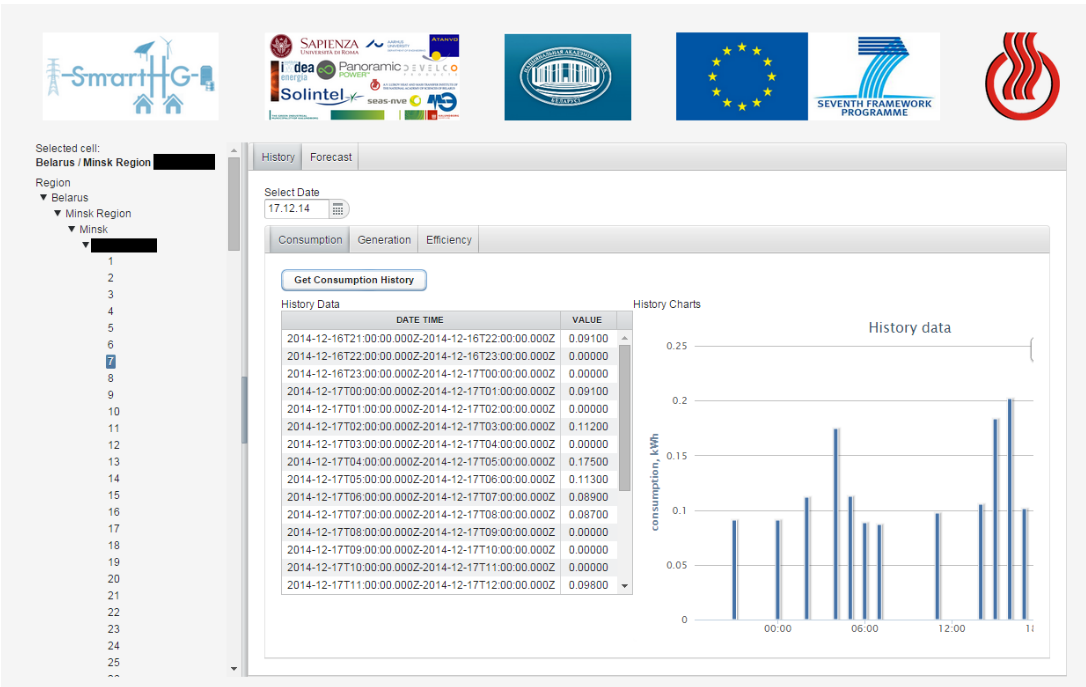

Figure 4.5: EUR-H and EUMF-H Web service: historical consumption of an home in Minsk test-bed.

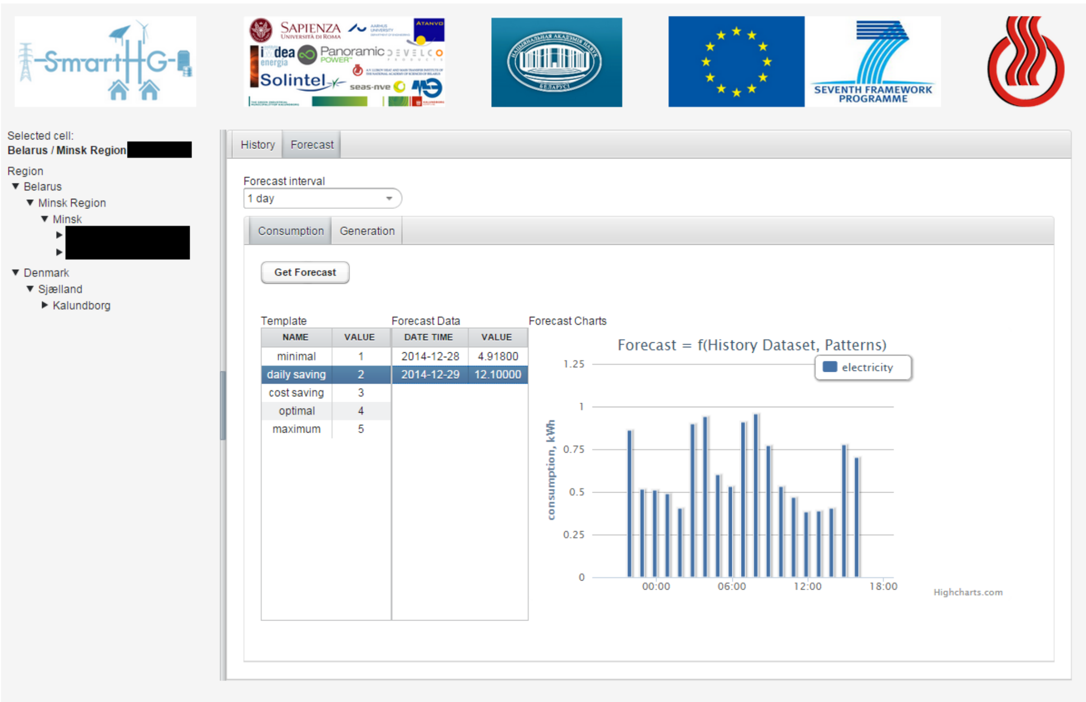

Figure 4.6: EUR-H and EUMF-H Web service: 1-day forecast for energy consumption of an home in Minsk.

Chapter 5

Home Energy Controlling Hub

In this section, we describe a prototype of the Home Energy Controlling Hub (HECH), enabling the communication between home devices and project Intelligent Automation Services (IASs), by using the open protocol designed in Deliverable D3.2.1. Such prototype consists of the following elements.

- RaspberryPi (RPi) as the execution platform (see Figure 5.1);
- DP ZigBee WallPlugs as home devices;
- DP ZigBee gateway to communicate with the remote server (see Figure 5.2);
- SmartAMM as the remote server;
- Database.

In order to show that such implementation fulfil our requirements, we test it w.r.t. three different scenarios:

- the HECH has to automatically configure a ZigBee Wall Plug and control it (see Figure 5.3);
- the HECH has to remotely sense and actuate home devices by accessing the SmartAMM server (see Figure 5.4);
- the HECH has to take measurements from home devices and send them to the Database Service (DBService) (see Figure 5.5).

All such test scenarios were successfully completed, thus the prototype of the HECH developed for the second year fulfils our requirements.

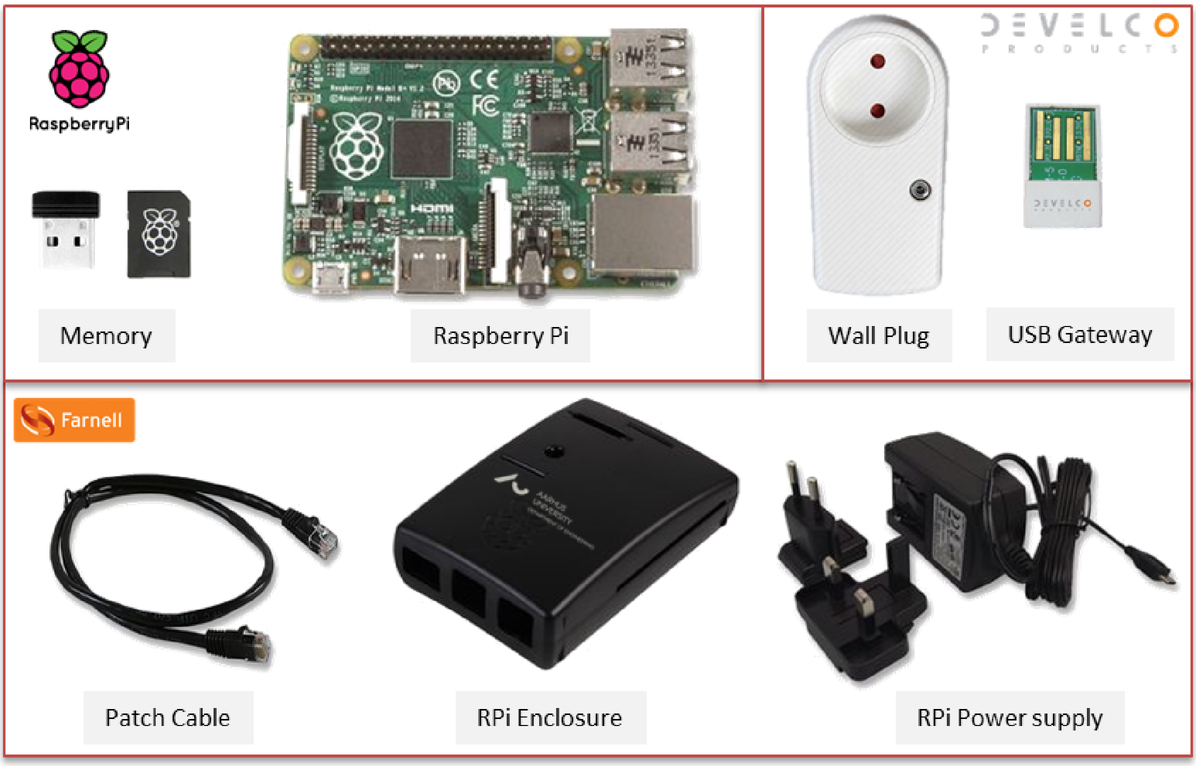

Figure 5.1: Prototype hardware components

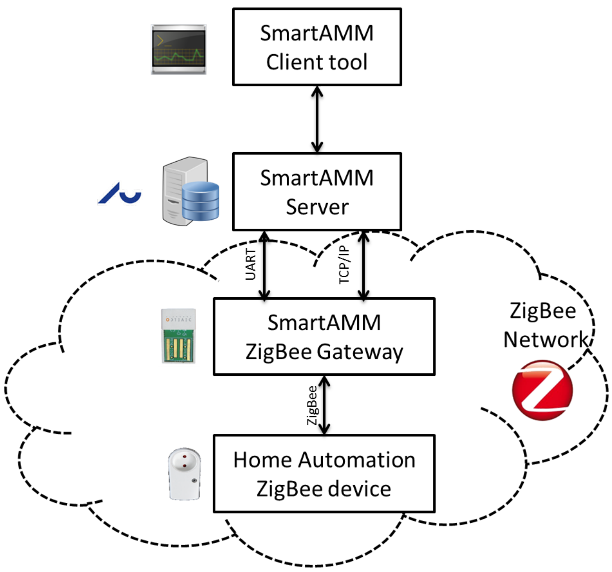

Figure 5.2: Prototype communication schema

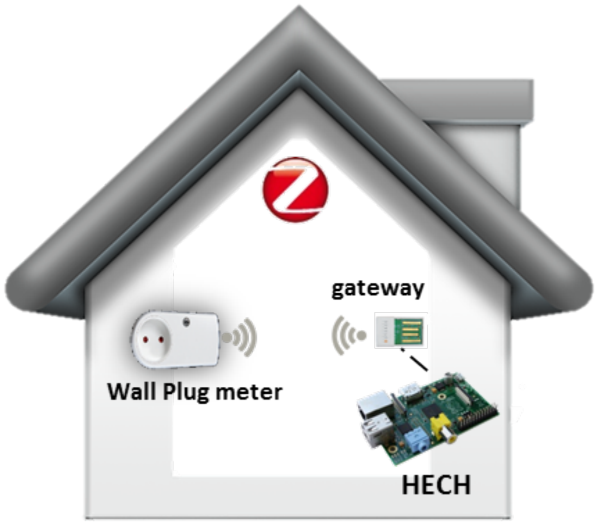

Figure 5.3: HECH test scenario: configuring and controlling a ZigBee Wall Plug

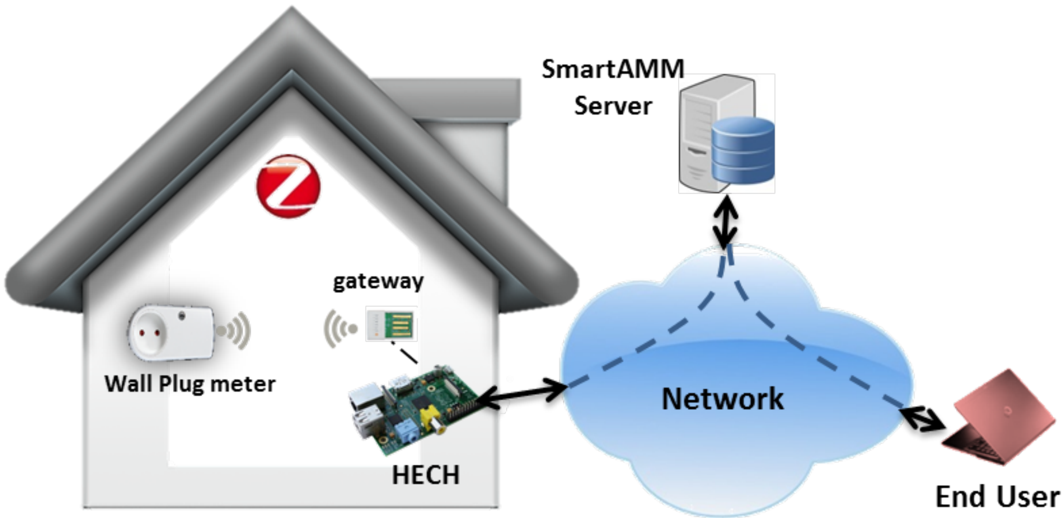

Figure 5.4: HECH test scenario: sense and actuate a home device

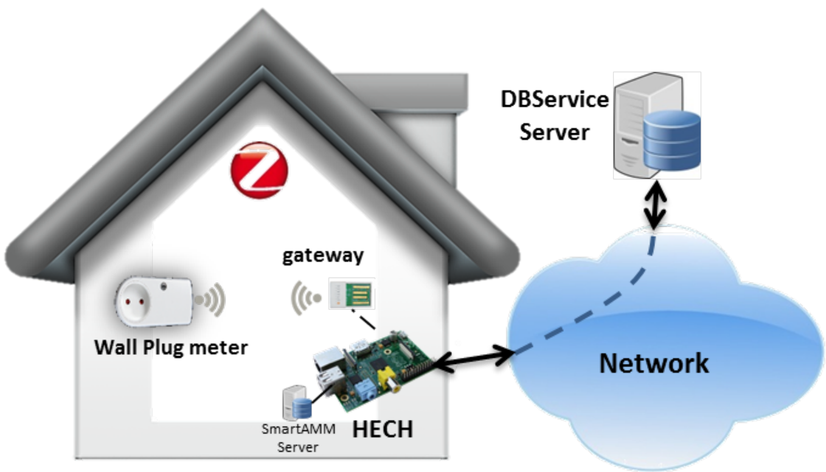

Figure 5.5: HECH test scenario: take measurements and send them to the DBService

.

Chapter 6

Conclusions

In this deliverable we described the second year versions of prototypes for all SmartHG Home Intelligent Automation Services (HIASs) Such prototypes are all based on the corresponding services design described in Deliverable D3.2.1. In the second year iteration of SmartHG, the main goal of such HIAS prototypes, together with the prototypes for Grid Intelligent Automation Services (GIASs) described in Deliverable D4.2.2, is to enable the second year version evaluation of SmartHG Intelligent Automation Services (IASs) described in Deliverable D5.2.1. As a consequence, the positive evaluation of SmartHG IASs in Deliverable D5.2.1 is also a positive evaluation of the prototypes described here. Finally, a further objective of HIAS implementation is to prepare an exploitable version of each HIAS, as will be discussed in Deliverable D7.2.1.

This section is divided in three parts. First of all, in Section 6.1 we compare this year iteration of the SmartHG HIAS prototypes implementation with the first year version of the same prototypes. Then, in Section 6.2 we discuss the limitations of this year iteration of the the SmartHG HIAS prototypes implementation. For each of such limitations, we outline the foreseen work to be done in the third year HIAS prototypes.

6.1 Advancements with respect to First Year HIAS Prototypes

In this section, we discuss the main advancements we obtained with this year SmartHG HIASs implementation w.r.t. the first year version of the same prototypes. To this aim, we discuss the enhancements for each HIAS.

EBR: Since this year design of Energy Bill Reduction (EBR) went through a complete restyling w.r.t. the first year version of EBR (see Deliverable D3.2.1), the EBR prototype had to be re-implemented. Moreover, differently from the first year prototype, a dedicated Web page to download both EBR software and documentation is available upon authentication.

EUR-K: Since the Energy Usage Reduction for Control (EUR-K) service has been developed this year, the implementation of EUR-K is completely new.

EUMF-K: Since the Energy Usage Modelling and Forecasting for Control (EUMF-K) service has been developed this year, the implementation of EUMF-K is completely new.

EUR-H and EUMF-H: W.r.t. the first year version of their ancestors Energy Usage Reduction (EUR) and Energy Usage Modelling and Forecasting (EUMF), a REpresentational State Transfer (RESTful) service has been implemented for both services. Moreover, the Web service went through a restyling, in order to make it easier to use.

HECH: The Home Energy Controlling Hub (HECH) had to be revised and reimplemented in order to be able to communicate with the new devices from Develco, as well as to be adapted to changes in the open protocol design.

6.2 Limitations of Second Year HIAS Prototypes and Future Work for the Third Year HIAS Prototypes

The main improvements of the third year version of SmartHG HIAS prototypes will consist in implementing the new features (and algorithms) of the third year versions of the HIASs.

For the EBR service, analogously to the Demand Aware Price Policies for Substation-Level Energy Storage Control (DAPP-K) service, the following also holds. The current version of EBR assumes to interact with a program simulating both an Energy Storage System (ESS) installed on a residential home and plug-in requests of a Plug-in Electric Vehicle (PEV) used at a residential home. As a consequence, in the third year we will investigate how to provide interoperability between the EBR and the ESS and PEV in the home, by using the home HECH and actuators from Develco. To this aim, the work done for the EUR-K service will be used to develop a version of EBR able to directly control heat pumps through Develco actuators.

Finally, as for the HECH, the privacy of users has not been currently addressed. Therefore, in the third year we will investigate how to ensure consumers privacy.

Bibliography

[1] "Energy Bill Reduction (EBR) Web Service: mclabservices.di.uniroma1.it/ebr/," 2014.

[2] "EBR RESTful Service: mclabservices.di.uniroma1.it:10005," 2014.

[3] "Energy Usage Modelling and Forecasting for Control (EUMF-K) Web Service: mclabservices.di.uniroma1.it/eumf_k/," 2014.

[4] "EUMF-K RESTful Service: mclabservices.di.uniroma1.it:10003," 2014.

[5] "Energy Usage Reduction for Control (EUR-K) Web Service: mclabservices.di.uniroma1.it/eur-k/," 2014.

[6] "EUR-K RESTful Service: mclabservices.di.uniroma1.it:10006," 2014.

[7] "Energy Usage Reduction for Homes (EUR-H) and Energy Usage Modelling and Forecasting for Homes (EUMF-H) Web Services: http://lesa.hmti.ac.by/smarthg-ui-demo/ui/," 2014.

[8] "EUR-H RESTful Service: http://www.lesa.hmti.ac.by/smarthg-eur/info/open," 2014.

[9] "EUMF-H RESTful Service: http://www.lesa.hmti.ac.by/smarthg-eumf/info/open," 2014.

[10] "All SmartHG Intelligent Automation Services (IASs) Web Page: http://smarthg.di.uniroma1.it/second_year_release/," 2014.