



INFSO-ICT-224557

BeAware

Boosting Energy Awareness
with Adaptive Real-time Environments

Instrument:	CA	STREP	IP	NOE
--------------------	----	-------	----	-----

ICT - Information and Communication Technologies Theme

D5.5 Public Summary of Service Models

Due date of deliverable (as in Annex 1): October 30th 2010

Actual submission date: December 14th 2010

Start date of project: May 1st 2008

Duration: 36 months

Organisation name of lead contractor for this deliverable: ENG

Revision: 1

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	



European Commission
Information Society and Media



Programme Name: ICT

Project Number: 224557

Project Title: Boosting Energy Awareness with Adaptive Real-time Environments

Partners: COORDINATOR: TKK (FI)
CONTRACTORS:
Helsinki University of Technology, TKK
BaseN Corporation, BaseN
Interactive Institute II AB, II
Engineering Ingegneria Informatica, ENG
University of Padova, UNIPD
Enel.si, ENELSI
IES Solutions, IES
Vattenfall Research and Development AB, VRD

Document Number: D5.5

Work-Package: WP5

Deadline Date: 30-october-2010

Date of Delivery: 14-december-2010

Title of Document: D5.5 Public Summary of Service Models

Author(s): Pasquale Andriani, Giampaolo Fiorentino (ENG)

Responsible Partner: ENG

Reviewer: Luigi Briguglio (ENG)

History: 14-december-2010: Internal Review Passed and Release

Availability: Public



Table of Contents

Executive Summary	4
1. Introduction	5
2. The Features of Service Layer	8
2.1. Consumption Knowledge Base	9
2.2. Message Service.....	10
3. Web Services API Details	14
3.1. Consumption Manager Web Service	15
3.2. Configuration Manager Web Service	17
3.3. Advice Manager Web Service	19
3.4. Quiz Manager Web Service	21
3.1. HouseholdProfile Manager Web Service	23
3.1. UserProfile Manager Web Service	24
3.1. Community Manager Web Service	25
References	26



Executive Summary

This document is delivered together with the relative software prototype. Starting with D5.4, the D5.5 aims at extending and consolidating the “Final Web Service Infrastructure” in order to report features undergoing some changes, as well as providing a summary of the whole Service Layer and its models, defined and implemented during the whole BeAware project.

Afterward the analysis of the trials outcome, the Service Layer undergoes some changes aimed at making it easier and more engaging. Indeed some of primary game rules have been changed i.e. the conditions determining the passage to the next level and the score calculation.

In this perspective, the following documentation exposes in detail the involved components updates, which are message-service and consumptionKnowledgeBase. Moreover, this deliverable gathers the list of all web services provided. In addition, it describes how the service layer will approach the next Trial.



1. Introduction

This document aims at giving a comprehensive description of the whole BeAware System's features set pointing out all services provided by Web Energy Service Infrastructure. The main goal of the project is to educate the people to the virtuous usage of the appliances. To reach this goal, BeAware uses both some charming ambient interface and the game metaphor that aims to engage the user through quizzes, social network and competition between several households.

Each household has one or more members who are the users of the BeAware System. Each user can control the real consumption of the household's appliances. In addition, s/he can play Energy Life by reading the tips and by replying to the quizzes proposed by the BeAware's system. These tips and quizzes are the mean to teach users for using the appliances fruitfully and for boosting energy best practices. All that is made possible thanks to an articulate architecture. The following overall picture of the BeAware Platform showing the links between the different layers gives a first architecture overview.

Figure 1 - The BeAware Energy Life Layers



In the above Figure 1, the Service Layer represents the natural bridge between the Sensing Platform, which provides consumption measurement data from sensor nodes, and the User Application Layer, which provides information to the user and makes him/her an active stakeholder into the consumer energy conservation chain. For clarity, Figure 1 shows the BeAware Platform overview: data coming from the Sensing Platform and power provider network (i.e. BaseStation and Data Store) are elaborated from the Service Layer and fed to User Application layer (i.e. Application Layer and Community).

Figure 2, on the other hand, shows an overview of how the BeAware System puts its idea in practice. Some wireless sensors have been applied to some appliances. The sensors detect the consumption energy by the appliance and send the detected data to a BaseStation that, working in harness with Service Layer, makes a first analysis of received data. The processed data arrive to the Web Service Layer (WSL) that renders them for the Mobile and the ambient interface.

Figure 2 - The BeAware Energy Life Outcomes



The WSL plays a key role in this scenario: it is a data manifold that moves together all detected data and transforms them into a BeAware Energy Life. The BeAware Energy Life is the technical outcome of the BeAware project. Conceived as a game, the Energy Life's aim is to improve the energy awareness in the households and boosting energy best practices. Basis, for the BeAware Energy Life are the social-cognitive models defined by the UNIPD team and refined during the project thanks to the results from the trials. These trials, with their fruitful feedbacks, helped us to understand where the Game concept could be improved and enriched. Their punctual feedback together with the study of users

behaviour led UNIPD team and us to recognise the weak points in order to refine them. Our study's outcomes brought some not intuitive game's behaviour to light.

Actually, the recent studies done by the UNIPD team gave attractive results: the BeAware System through Energy Life helped the user to save energy while having fun. Nevertheless, the same studies brought out some critical discoveries e.g. the rule description for the level passage was not easy to understand for the end user. So ENG and UNIPD team together with the rest of the Consortium have refined these conditions and some rules in the BeAware Game's logic.

This document illustrates the refinement of the Service Layer according to the trials results, showing the BeAware system components' new features and the updated ones. In appendix the list of current Service Layer APIs has been provided.

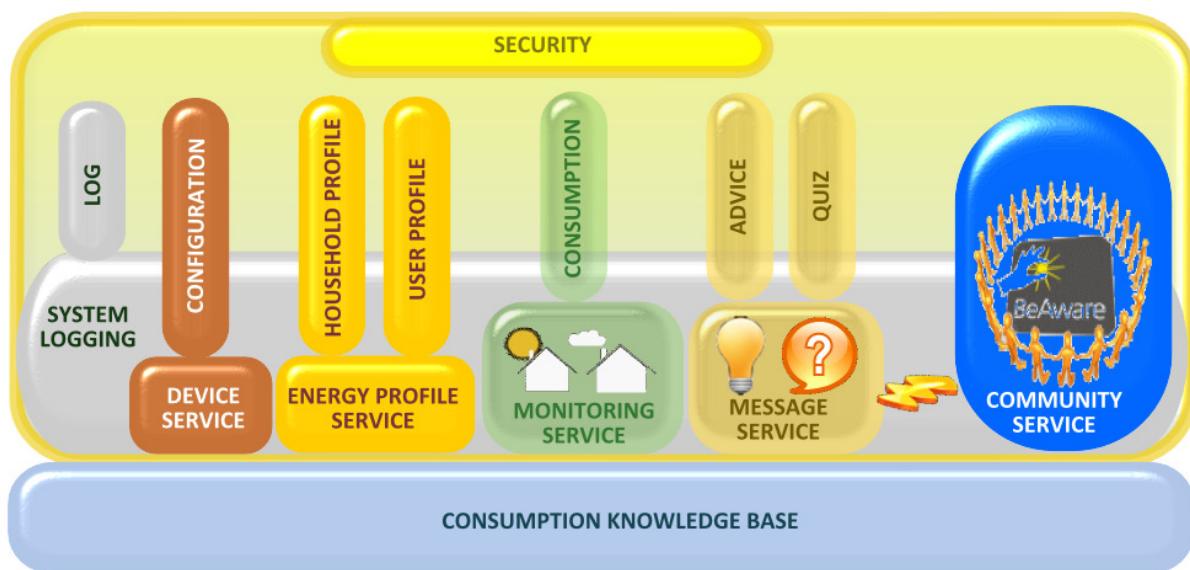


2. The Features of Service Layer

Trials results have been analysed and received by the Service Layer leading to some updates on some of the Service Layer architecture components and related features.

For seek of clarity, Figure 3 shows the Service Layer Architecture with the current components.

Figure 3 - Service Layer Architecture



Three main areas can be recognised from the figure:

- Key Components (Device Service, Energy Profile Service, Monitoring Service, Message Service) –they are the components making the core of the BeAware Application Layer, in this first area,
- Community – it’s an add-on to the Key Components core. It gives an added value to the Key Component area by enriching it with the access to the BeAware Community
- Consumption Knowledge Base – it’s the most important component. It’s responsible for managing the interactions with the sensors Layer and reading as well as processing the energy consumption data. Further details will be discussed in the Release Notes paragraph.

Moreover, a security mechanism has been identified that is responsible for the encryption and authentication.



Just some of the above components have been affected by modifications resulting from trials feedback. They are described hereafter.

2.1. Consumption Knowledge Base

The Consumption Knowledge Base is the component characterising the consumption parameters through the measurements coming from the Data Store of the Sensing Layer.

After the last plenary meeting that the BeAware Consortium had in Catania, the baseline's calculation period has been revised.

Therefore, the last Service Layer release considers the baseline differently. The algorithms for obtaining the Baseline and Saving have been revised, following the suggestions from the Review Panel.

In practice, it has been considered to have as reference value (i.e. Baseline) the energy consumed in the last week, so that data from the past haven't to be considered for the calculation (i.e. Baseline has not memory of the past energy consumption behaviour).

Moreover, the social-cognitive team recognises the importance of using the “calendar week” instead of the last 7 days. Users are able to recognise as the last week, the last calendar week from Monday to Sunday.

In this perspective, the Baseline (\mathcal{B}_7) is defined as below:

$\mathcal{B}_7 =$ = Energy consumed in the last calendar week, from Monday to Sunday. This value is weekly updated.

Moreover, in the I4.2, the daily energy consumption (ϵ_D) is defined as:

$\epsilon_D =$ = Energy consumed in the last day from 6.00 AM to 6.00 AM. This value is daily updated.

In this case, it's important to evaluate how those assertions can impact on the saving definition and how that value needs to be defined and consequently calculated.

If the energy consumed in a day (ϵ_D) has to be considered as the energy consumed from 6:00 AM to 6:00 AM, in this case, it's important that also the Baseline (\mathcal{B}_7) concept has to be calculated only during that same time span. And consequently the Baseline (\mathcal{B}_7) is the consumption in the last calendar week, by using consumption data gathered from 6:00AM to 6:00AM.

Three kinds of the algorithms have been taken into account for calculating the saving.



The analysed algorithms have been:

1. **Day vs. Baseline** - Saving is the comparison of the energy consumed in the previous day with the Baseline. Baseline is calculated as average daily consumption over 1 last calendar week;
2. **Sliding Days** - Saving is the comparison of the energy consumed during last 7 days (included the last one) with the calendar week Baseline.
3. **Calendar Days** - Saving is the comparison of the energy consumed with the energy consumed in the same calendar days of the last week (i.e. comparison of Monday with last Monday, comparison of Monday-Tuesday with last Monday-Tuesday and so on);

Among these ones, the Day vs. Baseline has answered to our simulations (see I.5.2 report). Day vs. Baseline is sensible to any change occurring in one day; by doing so it is able to take in account each little variation on the users' behaviour in order to achieve a careful energy awareness index.

2.2. Message Service

The following sub-sections describe the modifications on the Message Service component's features resulting from trials feedback. Briefly, Awareness Score features as well as rules and conditions (i.e. Level Logic concept) related to the game level logic have been updated, while a new feature related to Smart Advice has been introduced.

Firstly, a short description of the BeAware game's advice and tips is provided - even if they have not been changed - in order to give a clearer and more complete picture about the game logic.

Advice Tips and Quizzes

The user receives tips and quizzes in the language chosen by the user. Currently the Service Layer provides them in English, Sweden, Finish, Italian. Besides, according to WP2 specification, the tips lead the user to a virtuous use of own electrical devices using simple and direct advice that can be FRIENDLY or IMPERATIVE type. Each tip can be specific - associated to an electrical device- or general purpose –associated to a general behaviour.

The Service Layer delivers one random tip every day, so as explained in I4.1. The beUser receives one tip every day even if s/he doesn't use Energy Life: let see an example, if the user can't use the Energy Life for three days and the fourth day s/he opens the Energy Life, the Service Layer notices the event and sends three tips to Application Layer - one for each day. So the beUser can read the tips in order to learn new appropriate behaviours as well as increase his /her awareness score.



Similarly tips, the quizzes are delivered by the Service Layer every three days randomly. Again if the user can't use Energy Life for six days and s/he uses the Energy Life the seventh day the Service Layer delivers two quizzes randomly to the Application Layer. So the user can answer to the quizzes in order to improve his/her awareness score.

Tips and quizzes are distributed through the three different levels (see after). For each level the system takes into account only the advice and quizzes assigned to give level, so that the Service Layer assures that each level has its sufficient amount of tips and quizzes.

A huge different with the previous Service Layer version is the introduction of the multi-user. With the multi-user all that above does not change, each user will act his/her own quiz and tip copy. A beUser will be able to read a tip, even if this tip has been just read to another member of the family. Besides, the user will be able to answer the quizzes without disturbing the other users.

Finally, each single beUser of a household receives the tips and quizzes in the selected language. S/He can read and answer them respectively in order to increase her/his own awareness score so as move to next level.

Awareness Score

Since each member of the family can handle his/her tips, quizzes, community messages independently to the others, each member of the household (beUser) has his/her own aScore, qScore and cScore.

After the Catania meeting, the calculations of the advice score (aScore), the quiz score (qScore) and the community score (cScore) have been refined. The BeAware Consortium following the Cognitive Team's suggestions set the three following formulas:

$$aScore = \sum \# \text{ read tips}$$

$$qScore = \sum \# \text{ properly answered quizzes}$$

$$cScore = \sum \# \text{ read tips}$$

Moreover the BeAware Household score concept has been removed, in order to make more engaging the Energy life game.

The Level Logic

According to I5.2, there are three levels in BeAware Energy Life. Each level indicates the different energy consumption awareness.

The Level Logic has been updated as following:



- At the first level, LEVEL 0, the Service Layer asks to beUser for setting every existing and sensed device in the households – Configuration Phase - this can be done by one of the family members. After this phase, when that baseline calculation period is passed (i.e. one week), the Service Layer allows the user to move to the next level: LEVEL 1.
- At LEVEL 1, the Service Layer compels the beUser to stay in this level until the number of read advices is less than 20. At this level the user receives 1 advice per day. When the user reads 20 advices, the Service Layer moves him/her to the next level: LEVEL 2.
- The main difference between the level 1 and level 2 is the introduction of quizzes and smart advice (see the next section for further details on smart device). In fact, the beUser is enabled to receive a quiz every day and one advice every three days: the system sends a quiz to the logged users daily, the user replies to the quiz: if the answer is wrong the system sends another quiz to the user, freezing the wrong one for at least 2 days, otherwise if the user replies to the quiz properly the system will mark it as answered and will increase the Awareness Score user to 1.
- At level 3, the Service Layer sends on to the user one quiz every three days and one advice every five days. Furthermore the Service Layer allows the user to send a message to the others community members and at this level the ranking is enabled. The ranking may be done by using the following options: i) Indegree index (i.e. the sum of the all household's members); ii) Quiz Score (i.e. the sum of the all household's members) and iii) Saving Last Day

To summarize the three levels logic,

Level 0

- Baseline Calculation - No feedback.
- Calculation method: Day vs. Baseline: last calendar week (i.e. Monday-Sunday).
- Duration = 1 week from start of game

Level 1

- The user receives 1 advice per day
- The last day means: from 6:00 AM to 6:00 AM of previous day;
- Comparison of last day consumption with Baseline (Saving/consuming);
- Awareness score of a single user is given to the sum of all read advices
- The household's member passes at level 2 if:
 - S/He has Awareness Score of equal or more than 20
- Community disabled
- No constraints for the duration of level



- The user doesn't receive any quiz
- No constraint for the Saving Score
- No Score for the HOUSEHOLD

Level 2

- The user receives 1 advice every 3 days
- The user receives 1 quiz every day
- Smart Advice enabled
- The last day means: from 6:00 AM to 6:00 AM of previous day;
- Comparison of last day consumption with Baseline (Saving/consuming);
- Awareness score of a single user is given to the sum of all read advices, plus the number of properly answered quizzes.
- The household's member passes at level 3 if:
 - S/He properly answered at least 20 quizzes
 - S/He read all advice received at level 2 (at least 6 read advice)
- Community disabled
- No constraints for the duration of level
- No constraint for the Saving Score
- No Score for the HOUSEHOLD

Level 3

- Smart Advice enabled
- The user receives
 - 1 quiz every 3 days
 - 1 advice every 5 days
- Community enabled
 - Message enabled
 - Ranking enabled
 - Indegree index (message number a person receives)
 - Quiz Score (is the sum of the all household's members)
 - Saving Last Day

Smart Advice



According to the social-cognitive models identified in BeAware, the information provided to the user is more meaningful if it is contextualised to the real consumption and behaviour.

In this perspective, the social-cognitive team has identified a number of “smart advice and relative conditions” which may have an important role for improving the awareness of the energy usage and actions carried out by the user in the house. Such as the Advice, also the Service Layer provides the Smart Advice in four different languages: English Italian, Swedish, and Finish. The Smart Advice aims to adjust a specific user’s behaviours directly. The Service Layer does it monitoring the consumption of each single household’s appliance. When it recognizes wrong appliance use, suddenly the service layer sends a smart advice, in the language chosen by user, about how the user is using or used the appliance and how he could use it in a more efficient way.

Each smart Advice has been associated to an event. The service layer groups them in three big categories treated in three different ways.

The Smart Advice can be gathered in three different kinds of behaviour according to its conditions. There are some Smart Advice warning the user about their wrong behaviour during the whole week: i.e. *“This week you consumed much more electricity than the previous **week**, due to the excessive use of air conditioner. Please, try to use it less in order to save electricity”*

Moreover, there are ones warning the user about their wrong behaviour during the day: i.e. *“On Sunday, 7th October 2010, your coffee maker was kept in stand-by mode for y hours; by turning it off when you don't use it, you would save z g CO2 and create less pollution”*

Finally, the third family group is composed of some Smart Advice aiming to improve the wrong user behaviour during the night: i.e. *“On day Sunday, 15th August 2010 at night, your air conditioner produced y g CO2. You can save electricity by airing your home during the night”*

The Service Layer schedules the recognition of the conditions triggering the suitable smart Advice.

3. Web Services API Details

As reported in D5.1 , D5.2, D5.3 and D5.4, Service Layer is composed by different subsystems. This section provides technical documentation about the Web Services API exposed as JSON services. In particular, the documentation highlights the subsystems affected by changes respect to the previous releases (i.e. new methods introduced for adding features and supporting new scenarios).



3.1. Consumption Manager Web Service

Table 3-1 - Consumption Manager

SERVICE	CONSUMPTION-MANAGER-WS
ARTEFACTS	<p>SVN repository: http://davinci.eng.it/svn/BeAware/software/service/consumption-manager/</p> <p>Maven repository: http://davinci.eng.it/repo/it/eng/consumption-manager/ http://davinci.eng.it/repo/it/eng/consumption-manager-ws/ http://davinci.eng.it/repo/it/eng/consumption-manager-client/</p> <p>JSON interface http://test.energyawareness.eu/consumption-manager-ws/ConsumptionManager</p>
OPERATIONS	<p>List<DeviceData> getRealTimeDevicesData(List<String> deviceIds) It provides next to real time info on Devices identified by deviceIds. Each DeviceData object contains informations about current power consumption and device state.</p>
	<p>double getRealTimeHouseholdConsumption() It provides current power consumption of the whole household.</p>
	<p>CumulativeSaving getDeviceSaving(String deviceId) It provides saving for the last day for a specified Device. CumulativeSaving contains information about saving expressed both in KWh and in percentage value.</p>
	<p>List<CumulativeSaving> getDevicesSaving(List<String> devicesId) It provides saving for the last day for a specified set of devices. CumulativeSaving contains information about saving expressed both in KWh and in percentage value.</p>



	<p>CumulativeSaving getTotalSaving ()</p> <p>It provides saving for the last day for the whole household. CumulativeSaving contains information about saving expressed both in KWh and in percentage value.</p>
	<p>double getHouseholdBaseline()</p> <p>It provides the household’s baseline expressed as average KWh daily consumption over last full calendar week.</p>
	<p>boolean setSensingStartDate()</p> <p>This method allows to set the start date of the game for the user/household who invokes it.</p>
	<p>boolean checkSensingStartDateExist()</p> <p>This method allows to check if a start date of the game has been already specified.</p>
	<p>HistoricData getHouseholdHistoricConsumption(long start, long end, String step, int multiplier, String label)</p> <p>It provides Historical consumption data related to the whole household from <i>start</i> to <i>end</i> with a resolution of <i>multiplier * step</i>. <i>Start</i> and <i>end</i> parameter are expressed as the difference, measured in milliseconds, between the current time and midnight, January 1, 1970 UTC; <i>step</i> can be chosen from SECOND, MINUTE, HOUR, DAY, WEEK, MONTH; label is a title for the Historic series.</p>
	<p>HistoricData getHistoricConsumption(List<String> devicelds, long start, long end, String step, int multiplier,String label)</p> <p>It provides Historical consumption data related to the whole household from <i>start</i> to <i>end</i> with a resolution of <i>multiplier * step</i>. <i>Start</i> and <i>end</i> parameter are expressed as the difference, measured in milliseconds, between the current time and midnight, January 1, 1970 UTC; <i>step</i> can be chosen from SECOND, MINUTE, HOUR, DAY, WEEK, MONTH; label is a title for the Historic series.</p>



3.2. Configuration Manager Web Service

Table 3-2 - Configuration Manager

SERVICE	CONFIGURATION-MANAGER-WS
ARTEFACTS	<p>SVN repository: http://davinci.eng.it/svn/BeAware/software/service/configuration-manager/</p> <p>Maven repository: http://davinci.eng.it/repo/it/eng/configuration-manager/ http://davinci.eng.it/repo/it/eng/configuration-manager-ws/ http://davinci.eng.it/repo/it/eng/configuration-manager-client/</p> <p>JSON interface http://test.energyawareness.eu/configuration-manager-ws/ConfigurationManager</p>
OPERATIONS	<p>List<Device> getDevicesInfo() It provides the user household’s device list. Each Device object has a category, a type, a name, a Picture, a deviceId (also said sensorLabel), an appliance id and a sensorId.</p> <hr/> <p>Device getDeviceInfo(String deviceId) It provides all device’s info by specifying a deviceId (also said sensorLabel).</p> <hr/> <p>List<String> getDeviceCategories() It provides all device categories (existing in the BeAware Repository) as a list of String, according to the user specified language. (i.e. Lighting, Food preparation, Textile care)</p>



List<String> getDeviceTypes()

It provides all device types (existing in the BeAware Repository) as a list of String, according to the user specified language. (i.e. Television, Washing Machine, Microwave Oven)

void assignDeviceToSensor(String deviceId, String type, String category, String name)

It allows to assign a sensor to a device. It is done by assigning a type, a category and a name to a deviceId (also said sensorLabel). Type parameter must come from getDeviceTypes method and category must come from getDeviceCategories; name parameter is a free text user label to characterise his/her own device.

void updateDeviceInfo(String deviceId, String type, String category, String name)

It allows to update type, category and name of a device by specifying its deviceId. As in the method responsible for the assignment type and category must come from the proper method.



3.3. Advice Manager Web Service

Table 3-3 - Advice Manager

SERVICE	ADVICE-MANAGER-WS
ARTEFACTS	<p>SVN repository: https://davinci.eng.it/svn/BeAware/software/service/advices-manager/trunk/advices-manager-ws/</p> <p>Maven repository: http://davinci.eng.it/repo/it/eng/advices-manager/ http://davinci.eng.it/repo/it/eng/advices-manager-ws/</p> <p>JSON interface http://test.energyawareness.eu/advices-manager-ws/AdvicesManager</p>
OPERATIONS	<p>List<SmartAdvice> getAvailableSmartAdvice(String username) It returns the list of the available smart advices for the logged user.</p> <p>List<Advice> getAvailableAdvices () The Service Layer returns an Advice every day for the logged user. This Advice are said Available.</p> <p>List<Advice> getGeneralAdvices () The Service Layer returns a General Advice every day for the logged user.</p> <p>List<Advice> getAvailableAdvicesByCategory(String categoryId) It returns Available advices for specified category of the logged user.</p> <p>List<Advice> getAvailableAdvicesByDevice(String deviceId, int start, int end) It returns the list of the available advices for specified device for the logged user. The returned list gathers the advices from the start to the end position</p>



AdviceCounter getNumAdvices()

It returns an object representing the numbers of all available advices for the logged user

List<Advice> getAllAdvices()

It returns all advices for the logged user

List<Advice> getAdvices(int start, int end)

It returns a partial list of advices for the logged user. The returned list gathers the advices from the start to the end position.

List<Advice> getAllAdvicesByCategory(String categoryId)

It returns all advices for specified category of the logged user.

List<Advice> getAdvicesByCategory(String categoryId, int start, int end)

It returns a partial list of advices for specified category for the logged user. The returned list gathers the advices from start the position to the end position.

List<Advice> getAllAdvicesByDevice(String deviceId)

It returns all the advices for specified device for the logged user.

List<Advice> getAdvicesByDevice(String deviceId, int start, int end)

It returns a partial list of advices for specified device for the logged user. The returned list gathers the advices from the start to the end position

Boolean markAsRead(String adviceId)

It marks the specified advice as read

boolean markAsUnRead(String adviceId)

It marks the specified advice as unread



3.4. Quiz Manager Web Service

Table 3-4 - Quiz Manager

SERVICE	QUIZ-MANAGER-WS
ARTEFACTS	<p>SVN repository: https://davinci.eng.it/svn/BeAware/software/service/quiz-manager/trunk/</p> <p>Maven repository: http://davinci.eng.it/repo/it/eng/quiz-manager/ http://davinci.eng.it/repo/it/eng/quiz-manager-ws/</p> <p>JSON interface http://test.energyawareness.eu/quiz-manager-ws/QuizManager</p>
OPERATIONS	<p>List<Quiz> getAvailableQuizzes() It returns a Quiz every three days. This Quizzes are said Available</p> <p>List<Quiz> getAvailableQuizzesByCategoryId(String categoryId) It returns an Available quiz for specified category for the logged user.</p> <p>boolean isAvailableQuizz() It returns true if a new quiz has to delivery to user.</p> <p>Quiz getAvailableQuizzesByDeviceId(String deviceId) It returns an Available quiz for specified device for the logged user.</p> <p>QuizzesCounter getQuizzesCounter() It returns an object representing the numbers of all quizzes for the logged user</p>



List<Advice> getAllAdvices()

It returns all advices for the logged user

List<Quiz> getQuizzesByCategoryId(String categoryId)

It returns quizzes for categoryId for the logged user

List<Quiz>getQuizzesByDeviceId(String deviceId)

It returns quizzes for deviceId for the logged user

boolean submitQuiz(String quizId,String questionId ,String optionId)

It returns true if the chosen option is correct and false otherwise

3.1. HouseholdProfile Manager Web Service

Table 3-5 - HouseholdProfile Manager

SERVICE	HOUSEHOLD-PROFILE-WS
ARTEFACTS	<p>SVN repository: https://davinci.eng.it/svn/BeAware/software/service/household-profile-manager/trunk/</p> <p>Maven repository: http://davinci.eng.it/repo/it/eng/household-profile-manager/ http://davinci.eng.it/repo/it/eng/household-profile-manager-ws/</p> <p>JSON interface http://test.energyawareness.eu/household-profile-manager-ws/HouseholdProfileManager</p>
OPERATIONS	<p>List<Household> getAllHouseholds()</p> <p>It returns the list of all households being into the system.</p>



3.1. UserProfile Manager Web Service

Table 3-6 - UserProfile Manager

SERVICE	USER-PROFILE-WS
ARTEFACTS	<p>SVN repository: https://davinci.eng.it/svn/BeAware/software/service/household-profile-manager/trunk/</p> <p>Maven repository: http://davinci.eng.it/repo/it/eng/household-profile-manager/ http://davinci.eng.it/repo/it/eng/household-profile-manager-ws/</p> <p>JSON interface http://test.energyawareness.eu/household-profile-manager-ws/UserProfileManager</p>
OPERATIONS	<p>List<Person> getPeople() It returns all members of the logged user’s household</p>
	<p>Person getPerson() It returns the Person Object starting with the username</p>



3.1. Community Manager Web Service

Table 3-7 - Community Manager

SERVICE	COMMUNITY-WS
ARTEFACTS	<p>SVN repository: https://davinci.eng.it/svn/BeAware/software/service/community-manager/trunk/</p> <p>Maven repository: http://davinci.eng.it/repo/it/eng/community-manager/ http://davinci.eng.it/repo/it/eng/community-manager-ws/</p> <p>JSON interface http://test.energyawareness.eu/community-manager-ws/CommunityManager</p>
OPERATIONS	<p>void newMessageSent(List<String> receivers) This method increases the user’s cScore and the receivers’ indgreeIndex if they are at level 3</p>
	<p>List<HRank> getHouseholdRanking(String ORDER) It returns a ordered list in order to the order param. It can be SAVING, QSCORE, INDGREE</p>



References

[WP2-Score] The Awaress Score, internal documentation available on <http://wiki.hiit.fi/display/BeAware/Workpackage+2+Cognition+and+Practices+of+Energy+Consumption>

[I4.1] “Complement to D4.3 –Instantiation and Fine-tuning of Energy Life for Trials” – internal documentation available on <http://wiki.hiit.fi/download/attachments/6389987/BeAware-I4.1-v0.9-20100208.doc>

[I5.2] “I5.2 Complement to D5.5 – Feasibility Analysis for releases after M30” – internal documentation available on <http://wiki.hiit.fi/download/attachments/8389074/BeAware-I5.2-v0.2-20101025.doc>

