Project no. 224609

Project acronym: DEHEMS

Project title: Digital Environment Home Energy Management System

Instrument: *Please tick*	CA	STREP ✓	IP	NOE

ICT - Information and Communication Technologies Theme

Deliverable reference number and title (as in Annex 1):
D2.15: Future Requirements Evaluation.

Due date of deliverable (as in revised Annex 1): 31st July 2011
Actual submission date: 15th September 2011

Start date of project: 1st June 2008 Duration: 36 months

Organisation name of lead contractor for this deliverable: University of Salford

Revision [V2.0]

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)	

DEHEMS – Digital Environment Home Energy Management System

D2.15: Future Requirements Evaluation.

Document Control

Work Package Leader: The University of Salford

Document Owner: Professor Grahame Cooper

File Reference: Dehems_Deliverable_D2.15.pdf

Date: 15 Sept 2011

Version: 2.0

Version Control Record

Version	Date	Author	Comments
V0.1	14/09/11	Grahame Cooper	First draft.
V0.2	14/09/11	Grahame Cooper	Second draft.
V1.0	15/09/11	Martine Tommis	Formatting for submission
V2.0	16/09/11	Martine Tommis	Addition of information re federation protocol (p9) and correction/proof reading

Table of Contents

1 Executive Summary

This document sets out the research objectives for DEHEMS Cycle 3 and the user requirements for Cycle 3, taking account of preliminary evaluation of Cycle 2, and provides the basis for designing the Cycle 2 DEHEMS system and the Cycle 2 living lab research instruments.

Further analysis of Cycle 2 questionnaires and focus group transcripts will be carried out as the data becomes available and this may result in further refinement of the information contained within this document, particularly as it affects the actual running of the living labs during Cycle 3. This will result in updated version of this document being produced in the lead-up to the Cycle 3 living labs

This document contributes to milestone M3, which is due to be completed at the end of September 2010 (T0+27) and represents formally the cut-over from Cycle 2 to Cycle 3.

2 Activities Carried Out to Achieve the Deliverable

The Description of Work contains the following statements in relation to this deliverable.

Section 2.3: Future Requirements says: "Following the end of Cycle 3, future user and system requirements will be identified and contributed as a component of the final evaluation."

Deliverable 2.15 is described thus: "Future Requirements Evaluation – documentation of future requirements for a DEHEMS type system, following Cycle 3. This will form an input into the Commission Evaluation Report."

The purpose of this deliverable is taken to be to provide an outline specification for a DEHEMS-like system that might be deployed as a practical contribution to improving domestic energy usage behaviours, based on the experiences gained in the three development and usage cycles of the DEHEMS project. The inputs for this work focus primarily on Cycle 2 and 3, given that Cycle 1 was primarily a technical prototyping phase.

The deliverable has been created by distilling out the relevant issues arising from the evaluation reports for the two cycles concerned, along with the original objectives of the DEHEMS system and the design of the existing Cycle 3 DEHEMS system, and using these to produce the proposed set of requirements in the form of system objectives.

3 Inputs to the Future Requirements

3.1 Review of Key Points Arising from DEHEMS System Usage

This section discusses the key points arising from the earlier DEHEMS work that have influenced the proposals contained in section.

1. Users are generally not prepared to spend a significant amount of money on energy monitoring equipment. Irrespective of income, most UK users either would not consider buying such equipment or would not be prepared to spend more than £50, with £20 being the limit for low income households. (D7.6 Appendix 2, Figure 9) For Bulgarian users, the picture is similar, although many would be prepared to spend slightly more than UK users (up to €100) (D7.6 Appendix 2, Figure 10)

2. Whilst involvement in the DEHEMS project did result in a shift in motivation from cost toward ecological sustainability, even after using DEHEMS for a while, users are motivated to change energy usage behaviours as much by cost savings as by environmental considerations. (D7.6 Appendix 2, §2.2.3 & ; D7.7 Appendix 2 §2.2.3)

3. Reliability of the system is particularly important in maintaining long-term engagement of users (D7.6 Appendix 2, §2.3.1 ; D7.7 Appendix 2, §2.6.2 & §3.3), and should be favoured over complexity of the system and user interface (D7.6 Appendix 2, §2.3.2 (focus groups); D7.7 Appendix 2, §3.4). Users are also concerned about the accuracy of the DEHEMS measurements, and have observed discrepancies between the DEHEMS data and readings taken from their billing meter (D7.6 Appendix 2, §2.3.1 (4)).

4. It is important for users to be assured that the energy monitoring equipment itself, including the communication infrastructure, does not consume significant amounts of energy, and that it does not cause disruption to existing household internet facilites. (D7.6 Appendix 2, §2.3.1; D7.7 Appendix 2, §3.4)

5. In the absence of explicit measures to encourage “socialization”, the most important “added value” functionality of the DEHEMS system is the ability for a household to compare consumption against its previous consumption. (D7.6 Appendix 2, Table 11; D7.7 Appendix 2, Table 20). Households also indicated in Cycle 2 that they found the ability to compare against average consumptions across other households useful (D7.6 Appendix 2, Table 11). Other features that were appreciated included the provision of energy saving tips, and the ability to look at what types of household displayed similar consumption figures to their own (D7.7 Appendix 2, Table 20), but these need to be weighed against the need for reduced complexity.

6. Setting of personal goals and thresholds was considered less useful (D7.6 Appendix 2, Table 11), and requires considerable complexity in the user-interface to be implemented.

7. Access to appliance-level monitoring was seen to be particularly useful by those households that had it, in both Cycle 2 and Cycle 3. (D7.6 Appendix 2, Figure 24; D7.7

Appendix 2, Figure 6, §2.5.2, §2.6.1, & §3.1). This is reinforced by the fact that, in discussing changes made to energy usage as a result of DEHES, users seem to focus on specific appliance-related actions (including permanently connected “appliances”) and decisions to change appliances (D7.6 Appendix 2, §2.4(2); D7.7 Appendix 2, Figure 10). Finally, the quantitative data from Cycle 3 suggests that those households that viewed appliance monitoring data may tend to consume less energy (D7.7 Appendix 2, §2.7.7).

8. It was also apparent that the display of the overall energy consumption of an appliance over a period rather that its instantaneous power consumption can lead to better-informed decision-making (D7.7 Appendix 2, §2.5.2 & §2.6.3). This is relevant in relation to the design of displays.

9. Results from Cycle 3 confirm that it is important to embed the delivery of energy monitoring data within the daily lives of household participants rather than requiring them to log into a special web site to do this (D7.7 Appendix 2, §2.3.1). Users expressed a desire to have “always-on” displays (D7.7 Appendix 2, §2.3.2), and there was a greater level of engagement from those users whose data was integrated into their existing Facebook accounts, both in terms of their qualitative perceptions and in measured data from the system (D7.7 Appendix 2, §2.4.2). Users also expressed a desire for a great deal of flexibility and diversity in the ways that they are able to access their energy usage data (D7.7 Appendix 2, §2.6.3).

10. The introduction of opportunities for socialization in relation to DEHEMS data appears to enhance engagement and motivation. Once users had the ability to compare their energy consumption levels against those of households that they knew (i.e. their friends) (D7.7 Appendix 2, §2.4.2). It is worth noting that this was based on one single social networking platform, albeit the dominant one. There may be benefits to be obtained from expanding this to other online social media.

11. The introduction of team-based incentives, though hard to assess, appeared to have some impact, though not as significant as the online social networking (D7.7 Appendix 2, §2.4.3). However, just one particular incentive model tested in this particular project and it would be useful to allow various approaches to be tried out.

3.2 Observations from the Developer's Point of View

The author has spent some considerable time as a user of the system, as part of the development team, and as a third party developer through the creation of the Facebook application for DEHEMS. This section contains some comments based on those experiences.

1. The availability of cumulative information on energy consumptions over a period seems much more useful than instantaneous power figures, both at the aggregate home level and at the appliance level. However, it is important to be able to choose the granularity at which the data is summarized for display.

2. The accuracy of separate measurement devices is a concern. The DEHEMS partners, having carried out calibrations of a number of electricity sensors in homes, have found discrepancies of up to 20%. This resonates with observation 5 in section 5

3. Queries that are made against the raw DEHEMS system can be extremely slow to complete, especially where aggregation is needed. Significant benefits were obtained from the introduction of a simple data warehousing/OLAP functionality to improve the response and to allow a richer set of choices to be offered to users and developers.

4. The ability for functionality and completely new forms of use through third-party applications could produce a multiplier effect for extending a DEHEMS-like system with limited resources. To support this, the provision of a robust, fast, flexible API, providing data and functionality at a suitably high level of abstraction has been shown to be both feasible and beneficial, allowing two applications to be deployed (Facebook App and Energy Team Challenge App).

5. In the context of online social networking, whilst it is very useful to take advantage of existing online social networks to encourage information sharing, it is equally important to see energy data sharing as a means of building new communities around shared interests in sustainability. This affects the design of social networking applications.

6. Attention has been paid to the architecture of the existing DEHEMS system as a whole, and the system architecture that has been constructed has proven resilient in the context of the DEHEMS project. However, any future development is likely to be heavily dependent on the ability to build incrementally on a relatively simple but extensible system core.

4 Future System Aims and Requirements

4.1 Background and Aims

The general thrust or aim of a DEHEMS-like system remains to be proactive in encouraging behaviour change in relation to energy consumption. It is important in this context that we capitalize on the particular strengths of the DEHEMS approach. The particular strengths that DEHEMS is able to bring to bear have been identified as:

- provision of energy monitoring across a fairly large and diverse user community with data from all households being gathered together at a central point;
- provision of fine grained energy monitoring across households with data gathering and collation at a central point;
- provision of a range of value-added services that capitalize on the above two points by allowing processing and reasoning over a large accumulated data set.

These characteristics provide an opportunity to go beyond what already exists in the following ways:

- interaction with users as a community of households rather than as simply a collection of individual households, allowing users to share information about their energy usages, including making comparisons;
- harnessing of social pressures on an informal basis (through ad-hoc interactions) and on a formal basis (through incentive models) to encourage energy usage behaviour change;
- accumulation of a large dataset allowing longitudinal studies to be performed and allowing trends to be analysed over periods of years, and allowing incremental development of richer services;
- creation of a self-sustaining community of users and developers interested in pursuing such an approach on a long-term basis.
- integration of third-party applications into the DEHEMS platform, including existing online social networking technologies, to allow the development of online communities, and the further extension of the DEHEMS concept and system by the community itself, through third party development.

4.2 Future System Key Requirements

Based on the information presented in section 4coupled with the aims express in section 7, the following key requirements/recommendations are proposed for a future DEHEMS-like system:

a system architecture that supports incremental development based on a ***simple*** core energy usage data gathering infrastructure that provides services defined in a generic manner to support the development of a range of third-party applications, based on the proven architecture used for DEHEMS Cycle 3, and defined in deliverable D2.11;

1. open APIs that support third party extensions of the system with minimum effort:
    - at the data gathering end through adapters to existing and future data acquisition hardware,
    - at the user end through the development of third-party applications as defined in and already tested in DEHEMS Cycle 3,
    - and at the developer end through the creation of additional service layers and APIs providing richer derivative functionality;
2. basic OLAP services should be provided, as implemented in DEHEMS Cycle 3, to support the effective use of energy usage data through the APIs, with derived data calculated according to the approach described in .

3. independent implementations of the core infrastructure able to be deployed in multiple instances by whomever wishes to do so, supported by community-maintained and developed documentation;

4. It is recommended that a federation protocol is provided to allow sharing of information between separate installations of the core DEHEMS system.

5. basic measurements supported (with reasonable granularities indicated) for each household should be:

- overall electricity consumption (6 seconds to support real-time display; one minute for recording),
- overall gas consumption (hourly),
- multiple channels of individual appliance electricity consumption (6 seconds to support real-time display; one minute for recording),
- inside and outside temperatures (one minute to support future advanced processing and predictive functions);

6. overall electricity and gas measurements should be taken from the billing meters where possible to avoid discrepancies between the system readings and billing meter readings, to eliminate tampering when incentive schemes are in place, and to improve reliability generally;

7. caching of data should be provided at the household data collection point sufficient to store up to 2 days of data during communications failures, with bulk upload to the server upon reconnection;

8. data upload and management protocols should be based on those defined and implemented for the DEHEMS cycle 3 system, which has been tried and tested, with extensions to support caching and bulk upload;

9. a dashboard user-interface should focus primarily on system management, registration and configuration, with basic data presentation, including historical time series; any additional, more complex functionality should be left to external/third-party applications;

10. in order to provide flexibility and maintainability, individual data streams should be associated with the data that is being measured (e.g. a particular appliance) rather than with the sensor that is taking the measurements.

5 Appendix 1. Outline API Definition

The following API calls are required to allow third-party application development on top of the DEHEMS platform. It also supports rationalization of the main dashboard application so that improvements in the performance of the underlying engine will benefit all applications, including the dashboard.

5.1 Conceptual model

A ***Living Lab*** (abbreviated to ***LL*** here) is a collection of households associated with one DEHEMS partner, usually in a single city or area. It has attributes:

- livingLabId (int)
- livingLabName (string)

A ***Household*** has attributes associated with it to include:

- householdID (string [the MAC address of the household's data collector])
- livingLabId (int)
- typeOfHouse ({detached house, detached bungalow, terraced house, …})
- numberOfRooms (int)
- numberOfFloors (int)
- numberOfOccupants (int)

A ***DataStream*** represents a single stream of data, which is associated with one house. It has attributes:

- streamID (int)
- streamName (string)
- dataStreamType ({AggregateElectricity, AggregateGas, ApplianceElectricity, ApplianceGas, OutsideTemp, InsideTemp})
- dataStreamSubtype ({appliance types} plus null)
- units (string)

Each Household will be associated with:

- one DataStream of type AggregateElectricity
- zero or one DataStream of type AggregateGas
- zero or more DataStreams of type ApplianceElectricity
- zero or more DataStreams of type ApplianceGas
- one DataStream of type InsideTemp
- one DataStream of type OutsideTemp

API services

getSessionKey(appID)

Returns a session key to an application that may be used by the application to make signed requests against the API.

appID will be a string of 16 ASCII characters, generated randomly by an authorized DEHEMS project partner and allocated to the app concerned.

getHouseholdAttributes(householdID, sessionKey, hashCode)

Returns an associative array of (householdID, livingLabId, livingLabName, typeOfHouse, numberOfRooms, numberOfFloors, numberOfOccupants).

hashCode will be an application authentication code (digital signature) formed by concatenating *householdID*, *sessionKey* and *appSecret*, then calculating the SHA-256 hash of the resulting string.

sessionKey is a session key that has been obtained by the application from the DEHEMS API using the getSessionKey request.

getDataStreamList(householdID, [dataStreamType], sessionKey, hashCode)

Returns an array of (streamID, streamName, streamType, streamSubtype, units) If optional dataStreamType parameter is supplied, only datastreams of that type should be returned.

hashCode will be an application authentication code (digital signature) formed by concatenating *householdID*, *sessionKey* and *appSecret*, then calculating the SHA-256 hash of the resulting string.

sessionKey is a session key that has been obtained by the application from the DEHEMS API using the getSessionKey request.

getDataStream(streamID, resolution, endTime, numberOfPoints, [cum=false], sessionKey, hashCode)

Returns a sequence of *numberOfPoints* data points associated with consecutive time periods defined by the parameter *resolution* and each determined as the average value of the available data points in its respective time interval. *resolution* defines the time interval length for each data point ({minute, hour, day, week, month, year}). (For electrical energy, the units of the data points will be kW.)

If *cum*=true then each data point is to be multiplied by the length of the time interval (*resolution*) in hours. (For electrical energy, the units of the data points will be kWh.)

In the case of the hour, day, week, month and year resolutions, the data points returned should relate to fixed historical intervals based on the normal, recognized clock or calendar periods. This will allow data warehouse/OLAP technology to be used to improve the responsiveness of the API.

hashCode will be an application authentication code (digital signature) formed by concatenating *streamID*, *sessionKey* and *appSecret*, then calculating the SHA-256 hash of the resulting string.

sessionKey is a session key that has been obtained by the application from the DEHEMS API using the getSessionKey request.

getLatestData(streamID, sessionKey, hashCode)

Returns, for the specified dataStream, the array: ((hourPc, dayPc, weekPc, monthPc, yearPc), (minuteAvg, hourAvg, dayAvg, weekAvg, monthAvg, yearAvg)) whre the fields with the Pc suffix each indicate the percentage of the respective current time interval that has passed and the fields with the Avg suffix each indicate the average values of all available data points available so far within the respective current time interval.

hashCode will be an application authentication code (digital signature) formed by concatenating *streamID*, *sessionKey* and *appSecret*, then calculating the SHA-256 hash of the resulting string.

sessionKey is a session key that has been obtained by the application from the DEHEMS API using the getSessionKey request.

getLLDataStream(livingLabID, dataStreamType, dataStreamSubtype, resolution, endTime, numberOfPoints, [cum=false], sessionKey, hashCode)

Identical to *getDataStreamAve* except that the data should be based on averages taken across all data streams of the given type within the specified living lab. (Using *dataStreamSubtype* gives the ability, for example, to compare my freezer against the average for all freezers in the LL.) If *dataStreamSubtype* is null, then it should be ignored.

hashCode will be an application authentication code (digital signature) formed by concatenating *livingLabID, dataStreamType*, *sessionKey* and *appSecret*, then calculating the SHA-256 hash of the resulting string.

sessionKey is a session key that has been obtained by the application from the DEHEMS API using the getSessionKey request.

getTips(householdID, numberOfTips, [streamID], sessionKey, hashCode)

Returns an array of (tipID, tipText), each of which is a tip that has been selected for appropriateness based on the parameters given. In the case of the optional *streamID* parameter, this is typically used if the user is currently accessing information about a particular stream to ensure that the tip(s) presented are relevant to the current context of the User.

hashCode will be an application authentication code (digital signature) formed by concatenating *householdID*, *sessionKey* and *appSecret*, then calculating the SHA-256 hash of the resulting string.

sessionKey is a session key that has been obtained by the application from the DEHEMS API using the getSessionKey request.

getNearestNeighbourData(householdID, period, sessionKey, hashCode)

Returns an array of (overallConsumption, typeOfHouse, numberOfRooms, numberOfFloors, numberOfOccupants) for the 5 houses with the nearest consumption higher than theirs and the 5 houses with the nearest consumption lower than theirs over the specified period. The period may be "week" or "month" and refers to the most recent complete week or the most recent complete month respectively.

hashCode will be an application authentication code (digital signature) formed by concatenating *householdID*, *sessionKey* and *appSecret*, then calculating the SHA-256 hash of the resulting string.

sessionKey is a session key that has been obtained by the application from the DEHEMS API using the getSessionKey request.

5.2 Initial Authentication

When users register the MAC address of the household with an application, it is important that the application is able to ensure that the MAC address really is associated with that user. To ensure that the application is genuine at this time, it is important that the DEHEMS server is able to authenticate the application.

In order to achieve these aims, a suitable authentication protocol must be implemented that provides at least the functionality of the following calls.

getAuthenticationToken(appID, userID, returnURI, sessionKey, hashCode)

appID will be a string of 16 ASCII characters, generated randomly by an authorized DEHEMS project partner and allocated to the app concerned.

userID will be a string that identifies a user in the context of the application. (E.g. this might be the Facebook user ID number in the case of a Facebook application.)

hashCode will be an application authentication code (digital signature) formed by concatenating *appID*, *userID*, *returnURI* and *appSecret*, then calculating the SHA-256 hash of the resulting string.

sessionKey is a session key that has been obtained by the application from the DEHEMS API using the getSessionKey request.

appSecret will be a shared secret string of 32 ASCII characters, generated randomly by an authorized DEHEMS project partner and allocated to the app concerned.

returnURI will be the URI to which the user's browser will be redirected once the authentication is completed.

This request will be made by the application redirecting the user's browser to the authentication provider service on the DEHEMS server, with the appropriate attributes appended to the URI GET request. The DEHEMS authentication service will verify the *hashCode*, check whether the user is authenticated (logged in) to the DEHEMS server and respond to the application using either the *returnAuthenticationToken* request (if the user is authenticated) or the *returnAuthenticationFailed* request (if the user cannot be authenticated) shown below.

returnAuthenticationToken(userID, macAddr, sessionKey, hashCode)

userID will be a string that identifies a user in the context of the application. (E.g. this might be the Facebook user ID number in the case of a Facebook application.) This is supplied by the corresponding *getAuthenticationToken* request.

macAddr will be the MAC address for the home that is associated with the logged-in user on the DEHEMS server.

hashCode will be an application authentication code (digital signature) formed by concatenating *userID*, *macAddr*, *appID* and *appSecret*, then calculating the SHA-256 hash of the resulting string.

sessionKey is a session key that has been obtained by the application from the DEHEMS API using the getSessionKey request.

appID will be a string of 16 ASCII characters, supplied by the application in the *getAuthenticationToken* request.

appSecret will be a shared secret string of 32 ASCII characters, generated randomly by an authorized DEHEMS project partner and allocated to the app concerned.

This request will be made by the DEHEMS server redirecting the user's browser to the URI provided by the application in the *getAuthenticationToken* request, with the appropriate attributes appended to the URI GET request.

returnAuthenticationFailed(userID)

userID will be a string that identifies a user in the context of the application. (E.g. this might be the Facebook user ID number in the case of a Facebook application.) This is supplied by the corresponding *getAuthenticationToken* request.

This request will be made by the DEHEMS server redirecting the user’s browser to the URI provided by the application in the *getAuthenticationToken* request, with the appropriate attributes appended to the URI GET request.

6 Appendix 2 – Calculation of Derived Data for OLAP support of the DEHEMS API

This appendix describes the algorithm to be used to calculated derived data for be presented at the API on request. The aim here is to provide data in a form and at a level of abstraction that is suitable for presentation to end user applications, with minimal processing load on the server at the time of delivery, and placing the minimum load on third-party applications from both the development and data processing points of view.

This approach has been tested within DEHEMS Cycle 3 and shown to be suitable for use through two deployed applications (Facebook App and Energy Team Challenge App) as well as one proof-of-concept implementation (Java-based mobile phone app).

Intervals represented in the API (and therefore used for aggregation of data stored in the OLAP database) should be clock and calendar intervals as defined in the household's local time zone.

For definitions of the concepts relating to a data stream, and the time intervals to be represented (specifically: minute, hour, day, week, month, and year), see the API definition in .

If the available resolution for a stream is more coarse than the resolution listed here, then the higher resolution should be simulated by adding points between the available ones, with the added points being set to equal the next earlier real point.

The aggregation should be carried out as follows, bearing in mind that the data may be incomplete.

1. The raw points available should be averaged over each minute, with minutes that contain no raw data points being set to as <no data>. In the *minute* resolution, <no data> should be reported as zero in the data stream shown.

2. The *hour* level data should be calculated by averaging the available *minute* level points, ignoring any <no data> *minute* points, with hours that contain no minute level data points being set to as <no data>. In the *hour* resolution, <no data> should be reported as zero in the data stream shown.

3. Each resolution should be calculated from the next finer resolution points in the same manner as described in 1 and 2 above.