



Project no. 224609

Project acronym: DEHEMS

Project title: Digital Environment Home Energy Management System

| | | | | |
|-------------|----|---------|----|-----|
| Instrument: | CA | STREP ✓ | IP | NOE |
|-------------|----|---------|----|-----|

ICT - Information and Communication Technologies Theme

Deliverable reference number and title (as in Annex 1):

D6.3: System Deliverable for Cycle 3

Due date of deliverable (as in revised Annex 1): T0+37 30th June 2011

Actual submission date: 23rd September 2011

Start date of project: 1st June 2008

Duration: 38 months

Organisation name of lead contractor for this deliverable: Hildebrand Technology Ltd

Revision [V1.0]

| 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) | |



Document Control

Work Package Leader: Hildebrand Technology Limited

Document Owner: Hildebrand Technology Limited, Joshua Cooper

File Reference: D6.3_Deheims_Deliverable.pdf

Date: 23 Sept 2011

Version: V1.0

Version Control Record

| Version | Date | Author | Comments |
|---------|--------------|----------------|-------------------------------|
| V0/1 | 10/06/2011 | Joshua Cooper | |
| V1.0 | 22 Sept 2011 | Joshua Cooper | |
| V1.0 | 22 Sept 2011 | Martine Tommis | Proof reading and formatting. |

Table of Contents

| | |
|---|----|
| Abstract | 5 |
| Background and Motivations | 6 |
| Household Requirements | 6 |
| Customer Premises Hardware | 6 |
| System Design..... | 7 |
| User Interface Layer | 8 |
| Business Logic Layer | 9 |
| Consumption Data Collection..... | 9 |
| Dashboard functions | 10 |
| User & System Admin..... | 10 |
| DEHEMS Object Layer..... | 11 |
| Integrated Application Layer | 11 |
| Physical Layer | 11 |
| Application Scenarios | 12 |
| Module Documentation | 18 |
| Conclusions..... | 20 |
| Future Work..... | 20 |
| Appendix I – Installation Guide | 21 |
| Appendix II: Sensor and Data Collector Images | 25 |

Table of Figures

| | |
|---|----|
| Figure 1 UML diagram of the DEHEMS Cycle 3 System | 7 |
| Figure 2 Individual household dashboard showing power and energy consumption. | 9 |
| Figure 3 Timeseries graphs showing household usage compared by appliance..... | 10 |
| Figure 4 Physical system diagram for the DEHEMS system | 11 |
| Figure 5 Scenario 1 | 13 |
| Figure 6 Scenario 2 | 14 |
| Figure 7 Scenario 3 | 15 |
| Figure 8 Scenario 4 | 16 |
| Figure 9 Scenario 5 | 17 |
| Figure 10 Sensor and Transmitter | 25 |
| Figure 11 Radio (433 MHz) receiver | 26 |
| Figure 12 Circuit board for the data gateway device based on the Infineon processor | 27 |
| Figure 13 Back of data gateway board showing available ports. | 27 |
| Figure 14 Plugwise pass through plug measuring appliance level usage..... | 27 |

Abstract

The DEHEMS project collects sensor data from households and presents analysis and real time data back to end users through a web based interface for the purpose of changing behavior patterns towards energy efficiency. Cycle 3 of the DEHEMS project will be installed in 250 homes in the UK and Bulgaria with the system described in this document.

In Cycle 3, a new API, gas usage, temperature sensing, semantic layer and hardware package was developed to incorporate feedback from previous cycles, explore a federated architecture and to test the gas estimator model.

Background and Motivations

The DEHEMS project is using Living Labs as a methodology for research and development for both an energy efficiency system and the behavioral effects the system triggers. Cycle 3 has the purpose of testing the ability to measure gas, test new gateway hardware, use the API for publishing to various presentation and social networking systems external to the dashboard and create a set of interactions between end users. As in Cycles 1 and 2, it is important that the sensor and customer premises equipment be low cost, easy to install and compatible across UK and European electrical systems. There were improvements made to the installation process and simplification of the in home hardware that were suggested as a result of Cycle 2 feedback, however only small system changes were made at the device level outside of the support of new sensors; much of the system work was integration server side with advanced functionality.

A further iteration of foundational infrastructure has been laid for a technical system that can be upgraded over time, following a logical architecture set within work package 2. It focused on adding security features to syndicate data to other intelligent systems. Hardware produced as a part of this Cycle had the advantage of being used in Cycle 2, therefore Living Labs had familiarity with installation. We have made significant changes to the system design in order to dynamically deploy drivers for various types of sensors.

It is also emerging that there is an interesting role for cloud computing in the way that models are applied to sensor data whereby collective intelligence and large computational power are available - not only for consumer applications, but also for smart grid and distributed generation paradigms.

Household Requirements

As with the previous cycles, it is a precondition that the house hold has access to the live electricity cable such that a clamp sensor can be placed around it and that the transmitter unit can be located within a 3 foot cable length. A second precondition is that the household has a broadband connection with ethernet presentation from a modem, router or switch. Normal assumptions are made such that there are two mains plug sockets available for powering the equipment.

Households also have the ability to use appliance level metering and therefore must be able to support the physical locating of the appliance pass through plug.

For gas metering, each home was recruited on a case by case basis for suitability. This includes a meter that can be read by OCR, power to the telecommunications equipment and minimum distance placements of telecommunications for ATAC safety requirements.

Customer Premises Hardware

Hardware has been modified from Cycle 2 increasing the number of sensors that can be used, including CT clamps from Current Cost. This eases the installation, procurement and the leverage of installed base. There was also a need to support appliance level monitoring and a generalised architecture for more sensor inputs. The Current Cost sensor unit is still supported as well as a new clamp meter designed and built as a part of the project. The meter increases accuracy and communicates directly with the gateway rather than relying on the additional Current Cost display. The data collector has stayed with the openWRT

enabled embedded router with an Infineon chipset. The software that is on the data collector is custom designed for the project. Further serial inputs are supported with the new software, namely the Plugwise appliance level metering that utilises Zigbee as the local wireless network.

Gas metering ran through a GSM phone solution, integrating data from the OCR readings via a text messaging link. Some control was enabled through SMS responses back to the GSM module as well.

Code location

There is a repository for code located within SVN at <http://svn.dehems.org/repos/dehems>. A user name and password must be granted in order to work with code. Access can be granted from Hildebrand Technology Limited contacts.

System Design

Cycle 3. implementation is taken from the UML diagram below that defines the high level systems architecture. The areas of social networking and ontology services were moved into usable systems, with semantic systems and Facebook integration coming online.

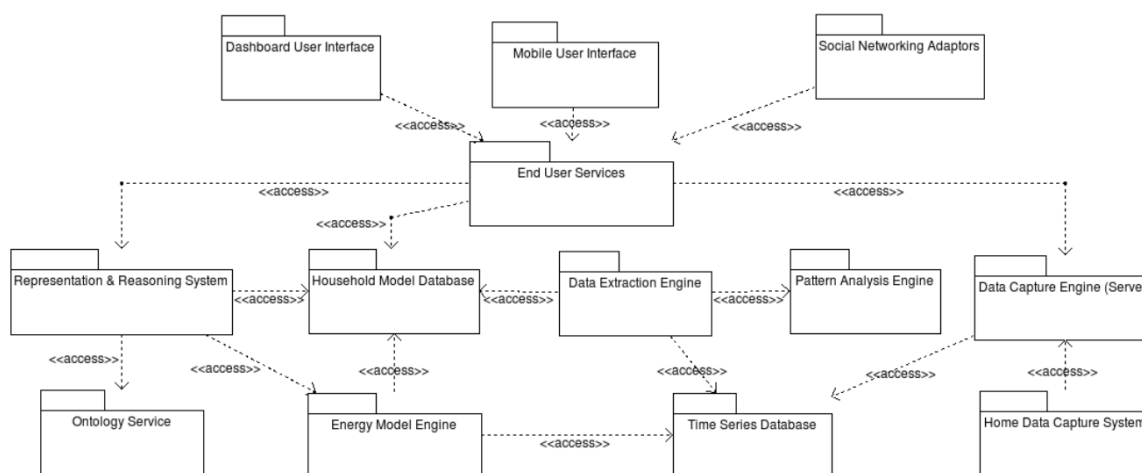


Figure 1 UML diagram of the DEHEMS Cycle 3 System

Third party packages and languages were used for components. These were selected on a preferred basis of open source, robustness and fit for purpose. The following list shows the details:

- operating systems: Linux Centos 5.2 (32 bit for application servers, 64 bit for databases), for data collector, openWRT rev1112.
- database: MySQL 5.0.45 (64 bit), memcached 1.2.5-2 (64 bit)
- web/application servers: Apache 2.2.3 (32 bit)

6.3 System Deliverable for Cycle 3

- application server language: perl 5.8.8 (32 bit), PHP 5.1.6, ModX
- browser: Firefox 2/3 (preferred), Safari 3, IE 7+
- ontology services: Java - Jess and runtime Protege components - software written by Coventry University
- embedded operating system: openWRT
- social networking integration: software written as a Facebook application by Salford University

User Interface Layer

The end user interface was designed for use in a web browser, with data updating into dynamic graphical areas. Flash was used for the dials and other rich controls as it was suggested from Cycle 1 that Flash would provide a better user experience. Flash was combined with other AJAX methods to retrieve data. Two areas were implemented, the Dashboard and the User&SystemAdminUI.

The User&SystemsAdminUI implements registration, password changes, authentication, entitlement and profile management. Registration is done by end users without the need for administrators to pre-configure hardware on a user by user basis.

The user interface requirements were given in WP2 and the design input was given from the WP2 group. The dashboard functions implement the end user requirements for showing power and comparatives to others in the Living Lab in real time. The dial data is updated every 6 seconds and implemented with easing on the dials so that an analogue feeling is created. Other data on the page is refreshed on page reload.

Status information on hardware performance is displayed in the top right with a green indicator showing that the data gateway and data sending are occurring. The status lights update every 5-10 seconds on a change of state. If red indicators are shown, trouble shooting tips are displayed.

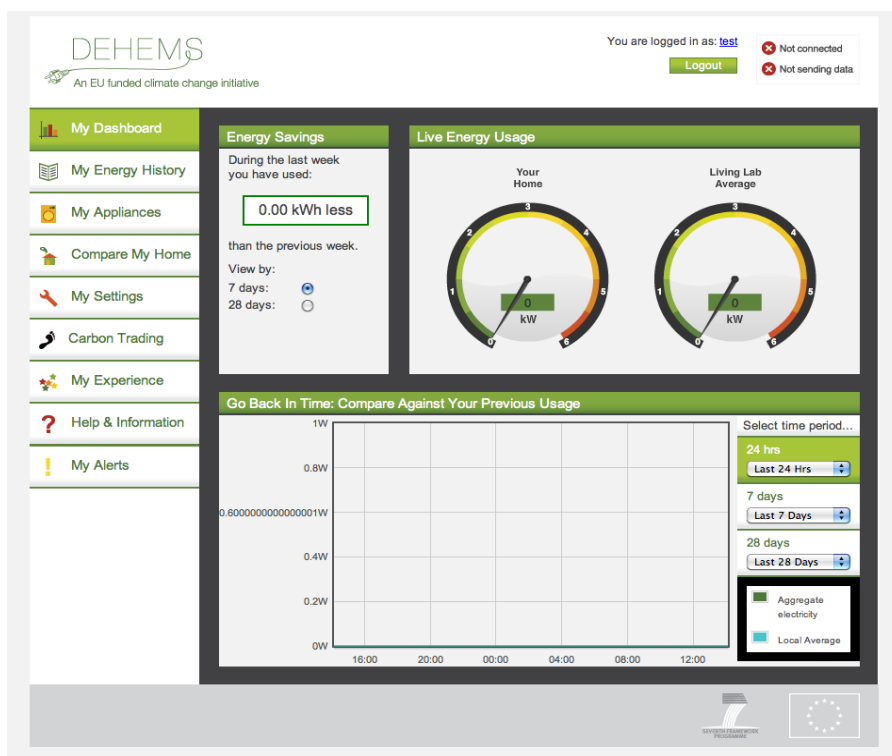


Figure 2 Individual household dashboard showing power and energy consumption.

There was a change to display the individual’s usage over time rather than accumulated measures displayed in Cycle 2.

Business Logic Layer

Interfaces are supplied by the business logic layer in the form of web services. In the Cycle 3 Implementation the business logic layer is exposed to the Internet, so that it can be used by Javascript and Flash components, the data collector device and public web pages. Additional security was added to be able to integrate Facebook applications with an application key assigned by our system.

Consumption Data Collection

The sensor data is collected at the household through a small embedded computer shown in Appendix 2. The data collector brings the serial stream of XML from the sensor unit and makes an HTTP POST to a web service that handles the incoming data. A more detailed description of the consumption data collector hardware and the sensor unit is found in Deliverable 3.7.

The data collector uses openWRT as an operating system with an application, a serial line reader to HTTP client to update data into the central system. The serial line is a USB to RJ-45 connector supplied by the sensor meter manufacturer.

Note the server for the data collector is presently contained in the same perl module as the dashboard functions, Data_Deheems.pm. The check boxes and drop down filter data on dimensions for appliances.

6.3 System Deliverable for Cycle 3

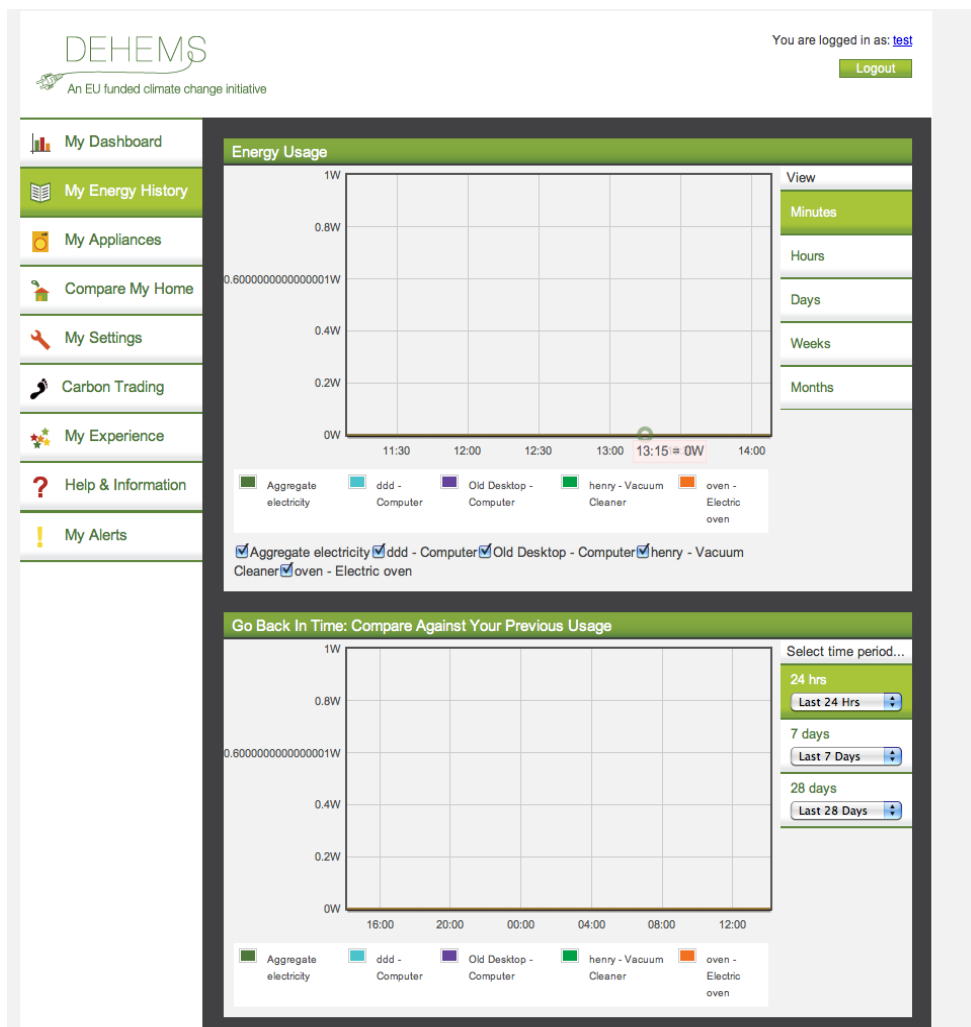


Figure 3 Timeseries graphs showing household usage compared by appliance.

Dashboard functions

Functions for getting data out of the statistical engine and supplying it in the proper format for AJAX display controls is done as a perl module fronted by an Apache web server. This is shown in Figure 4. as the data.dehem.org server. More detail around the functions implemented is found in subsequent sections. Presently the implementation for dashboard functions and consumption data collection are in the same module, Data_Deheems.pm. This is for convenience and can be implemented as separate physical implementations.

User & System Admin

This is implemented through a separate web server infrastructure that enforces user rights for viewing, manages the HTML, CSS and layouts for pages through a PHP content management framework called ModX. A HTTP proxy is configured on the server in order for client code (Javascripted and Flash UI controls)

6.3 System Deliverable for Cycle 3

A minimum of two Linux servers are utilized for the live DEHEMS system, this is to minimally split the load between application and database functions. Load balancing and further partitioning is easily implemented given the architecture found in Figure 4. A duplicate set of servers is available and backups are taken regularly, however, it is the intention to run parallel servers and services with different implementations to be able to compare different systems such as persistent storage.

The physical layer effectively has two databases implementing the Energy Usage Database and the User & System Database and the set of sensors in the field. As with all layers the physical layer is able to be replaced with different technology and several different technologies will be tested throughout the project.

There was an additional set of servers implemented in Bulgaria and a set in Romania, for the purposes of:

- trialling advanced storage, database and aggregation techniques at Rousse
- achieving fault tolerance at leAT
- providing a platform for services such as the semantics to integrate at an SQL level at leAT

Application Scenarios

The physical architecture implements basic storage, retrieval and statistical calculations over a fairly simple system. As much as possible memcached is used to work with data. This dramatically decreases the load on reads from any persistent layer that is implemented.

The use of in memory data storage is strategically either as a cache that is re-populated from persistent data or updated from remote devices. Daily statistics per household and group averages (including all households) are updated every 5 seconds, from persistent data if necessary. Data from devices occurs every 6 seconds and reflects instantaneous readings, therefore no refill of memory from persistent storage is done, we let the next sensor reading populate the memory store.

6.3 System Deliverable for Cycle 3

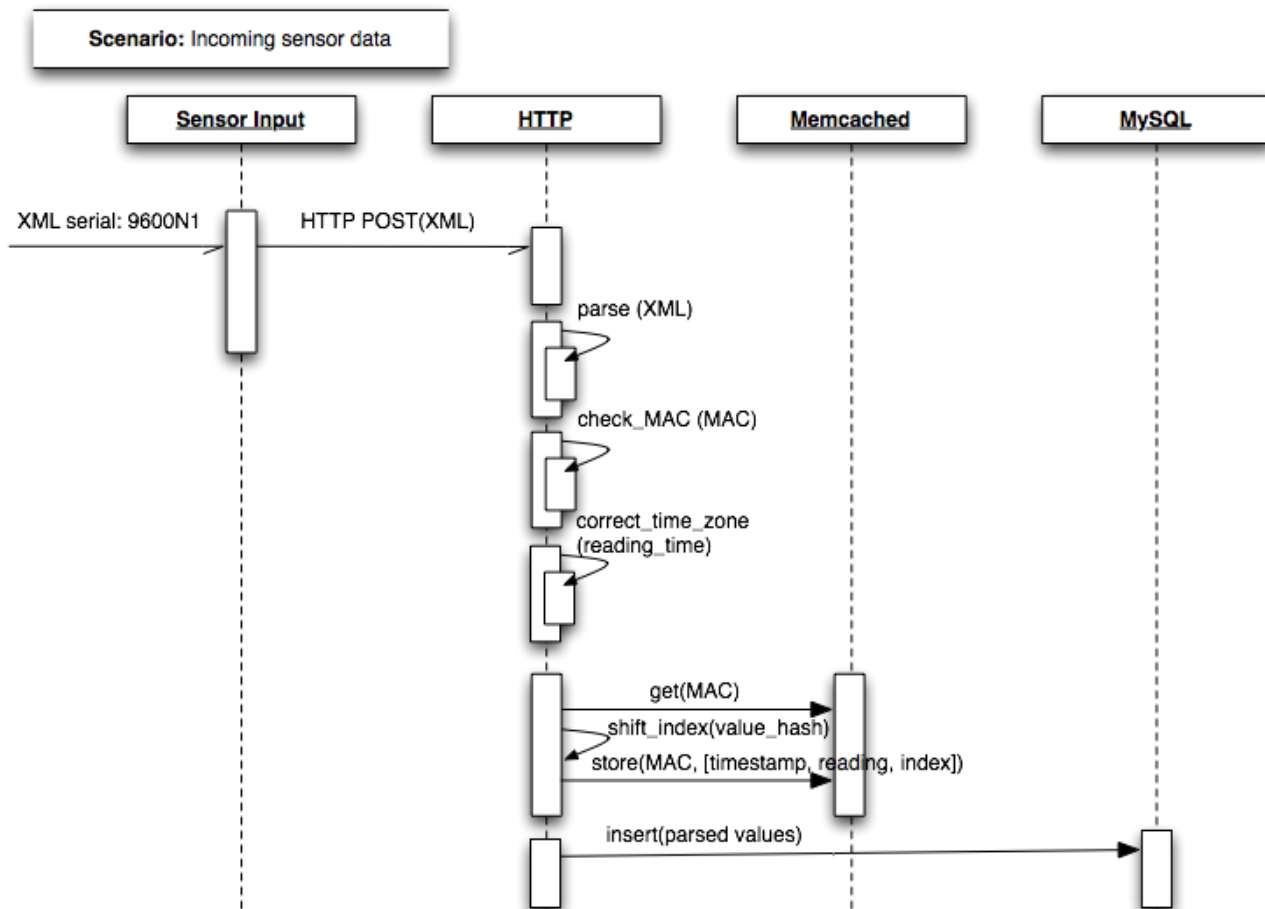


Figure 5 Scenario 1

Implemented in Data_Deheems.pm, data is inserted into structured in memory and persistent data stores.

6.3 System Deliverable for Cycle 3

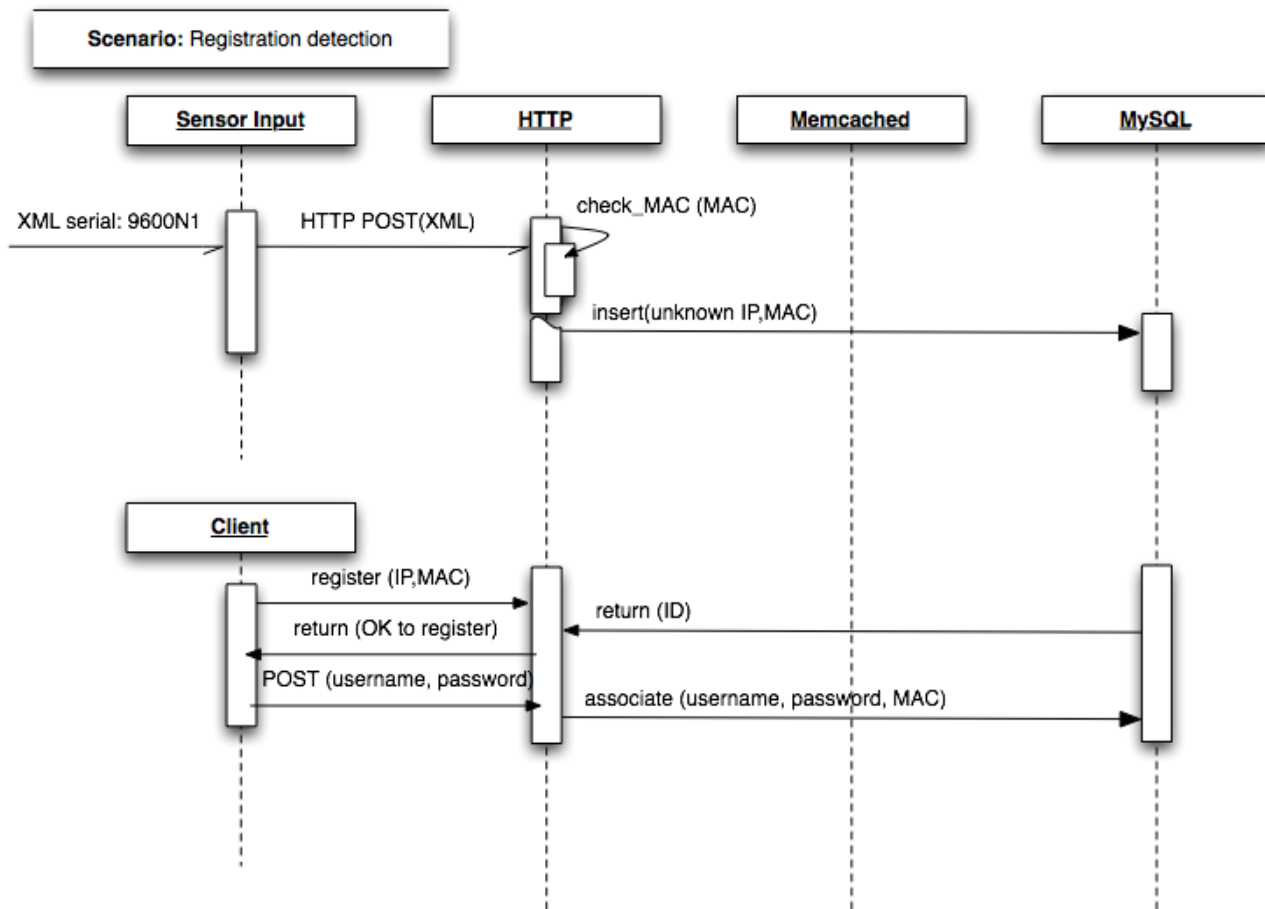


Figure 6 Scenario 2

Data collector boxes are not configured with a user name and password. This that is set by end users when they install their equipment. A simple scheme is used that relies on the IP address from the data collector box to be the same as found from the browser where configuration is completed. The IP addresses being the same, albeit only for this brief period of time, a link between a supplied user name, password combination and a MAC address can be made. Subsequently, only user name and password are required to access information for a MAC address.

6.3 System Deliverable for Cycle 3

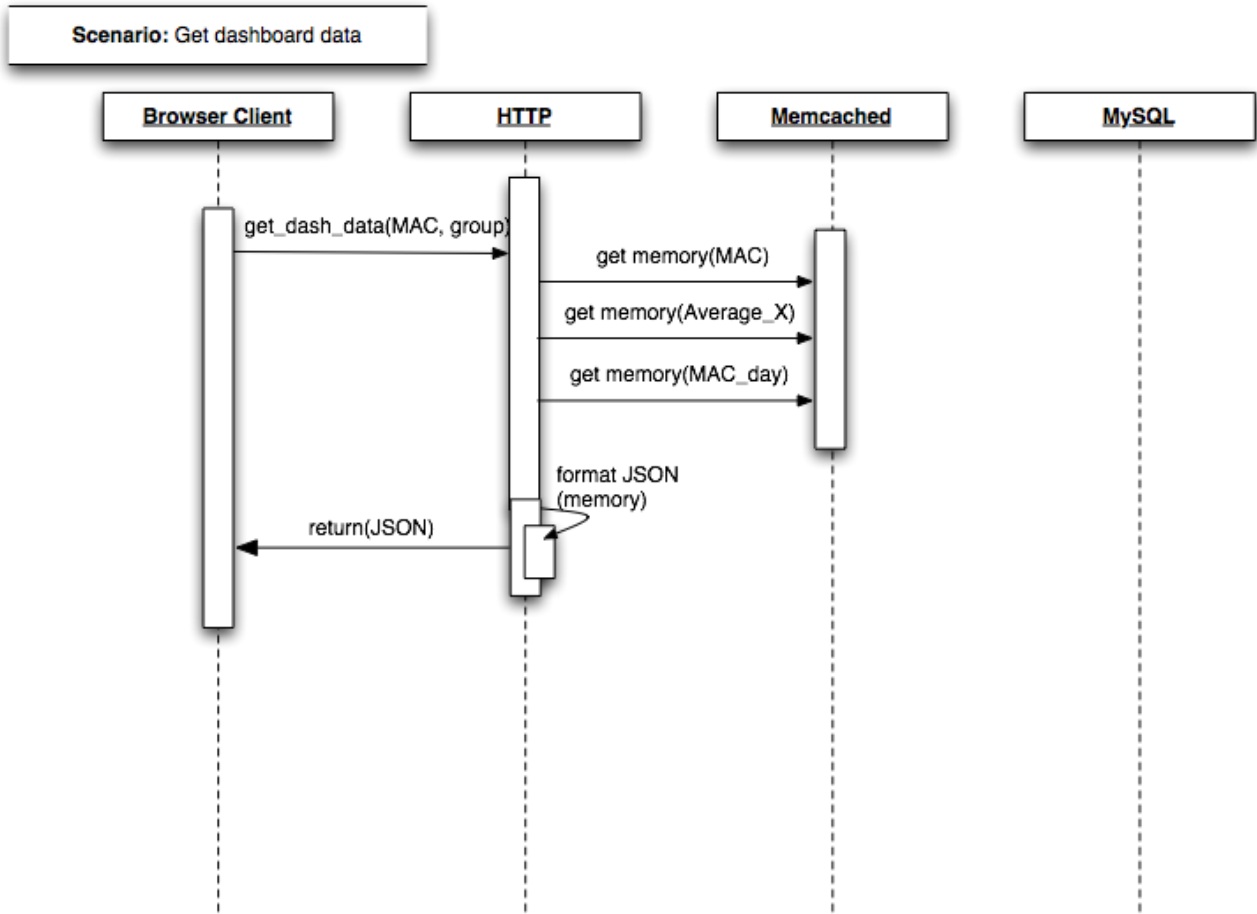


Figure 7 Scenario 3

Retrieving data for the dashboard is done as JSON

6.3 System Deliverable for Cycle 3

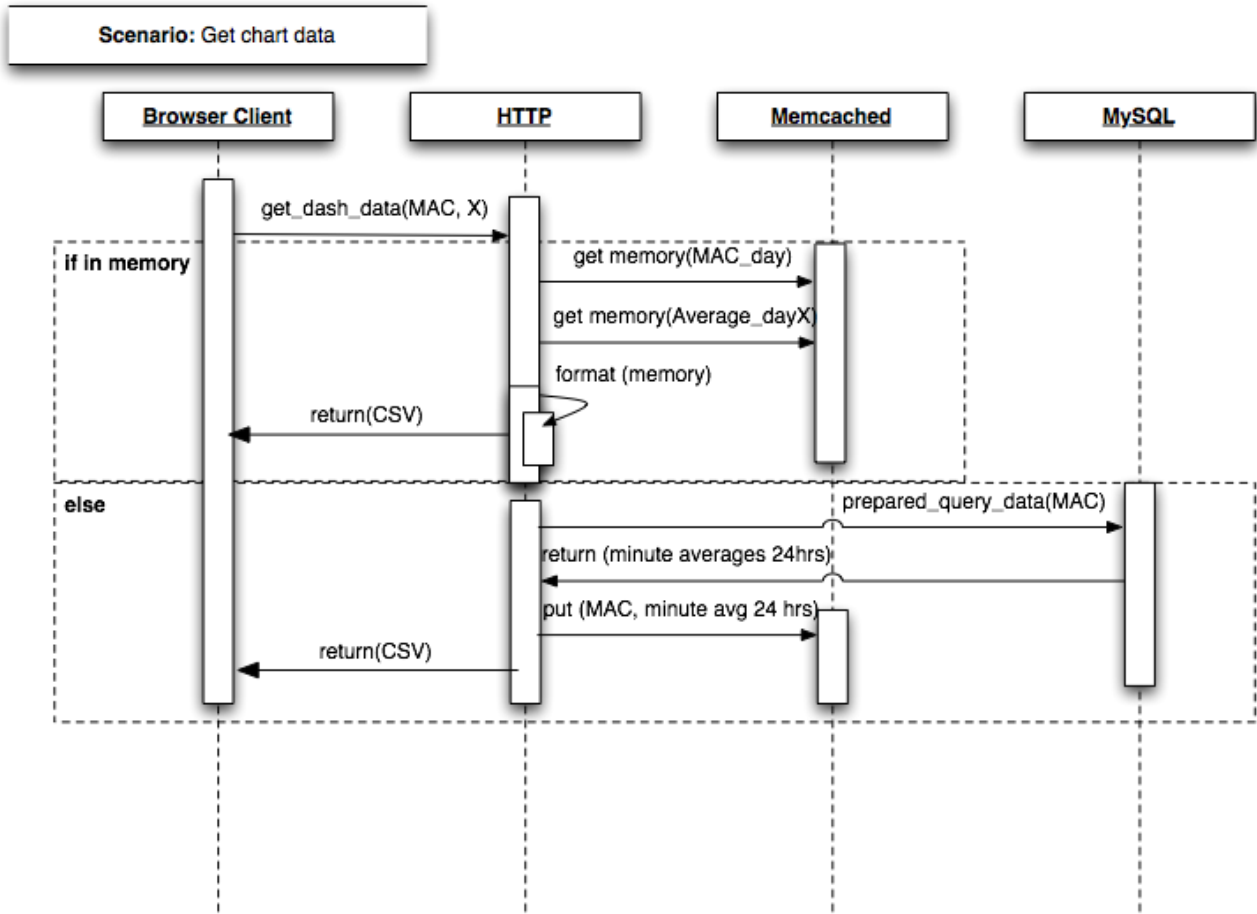


Figure 8 Scenario 4

Detailed chart data returned as comma separated values to power the MIT, SIMILE time plot.

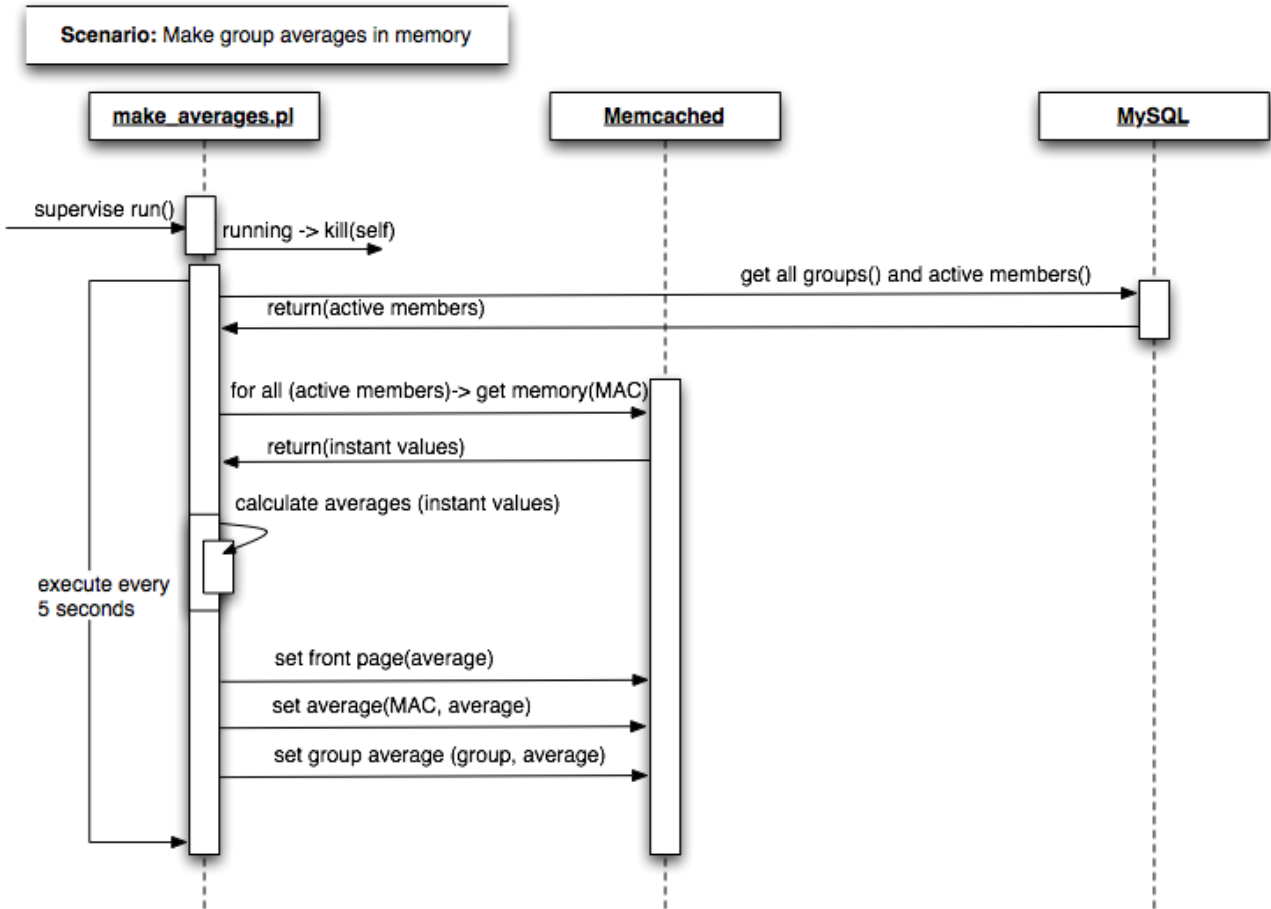


Figure 9 Scenario 5

Constantly running process make_averages.pl cycles through readings contained in memcached.

Module Documentation

Name:

Data_Deheems.pm - perl Apache 2 module for data insert

Description:

This module implements the data insert into memcached and MySQL. Memcached does not have a schema, therefore the key and hashed values are defined in code within this module. The module relies on DBI and XML parsers readily available in CPAN and follows common templates for perl Apache 2 module creation.

Memcached keys are:

MAC MAC address is used as the key for individual households formatted as a 12 character string e.g. 000E2EE0B827

AverageX where X is a string representing a group identifier.

MAC_day where the MAC address is used as the key for individual households formatted as a 12 character string e.g. 000E2EE0B827

Average_dayX where the MAC address is used as the key for individual households formatted as a 12 character string e.g. 000E2EE0B827

Frontpage a single key holding publicly available aggregate information about the entire set of households

Within each key type there is a hash for the value which is a data structure storing data against the key.

MAC keys store a data structure that has the latest time stamp in position 1, followed by 10 data slots for values, these are stored as perl arrays.

AverageX keys store a single integer value representing group average power consumption in watts.

MAC_day keys store minute averaged values of the power for a MAC address, there is a maximum of 1440 values present in a perl array.

Average_dayX keys store minute averaged values of the power for a group defined by string X, there is a maximum of 1440 values present in a perl array.

Frontpage keys store statistics for the whole set of households with the first slot holding the minimum household power in watts for the set, the second slot the maximum, the third slot the count of households, the fourth the sum of all usage across the set and the fifth the average for the whole set.

Prepared statements are implemented for SQL queries against MySQL. The schema is shown in Appendix 1. with the main table implemented as a high volume row store with some tables implemented as aggregates around the row store.

Functions:

`get_dash_data.html` returns a JSON object with the real time household statistics

`get_chart_data.html` returns a CSV text file of time stamps, minute averages of power (watts) for a specified household and group

`get_front_data.html` returns the Frontpage data structures as JSON object

`recxml` records the inbound XML from the data collector

`get_version` simply returns the version number of the module

`working_copies/data_collector` is a function exposed to push new software updates to the data collector box.

Conclusions

This system has been deployed in over 8 households for more the duration of the test cycle. It has been shown that the system is manageable, functional and able to be upgraded while in the field.

Testing has been done with simulated load to handle well in excess of 1000 households. In order to study patterns and behavior all data will be kept at its original resolution for future analysis.

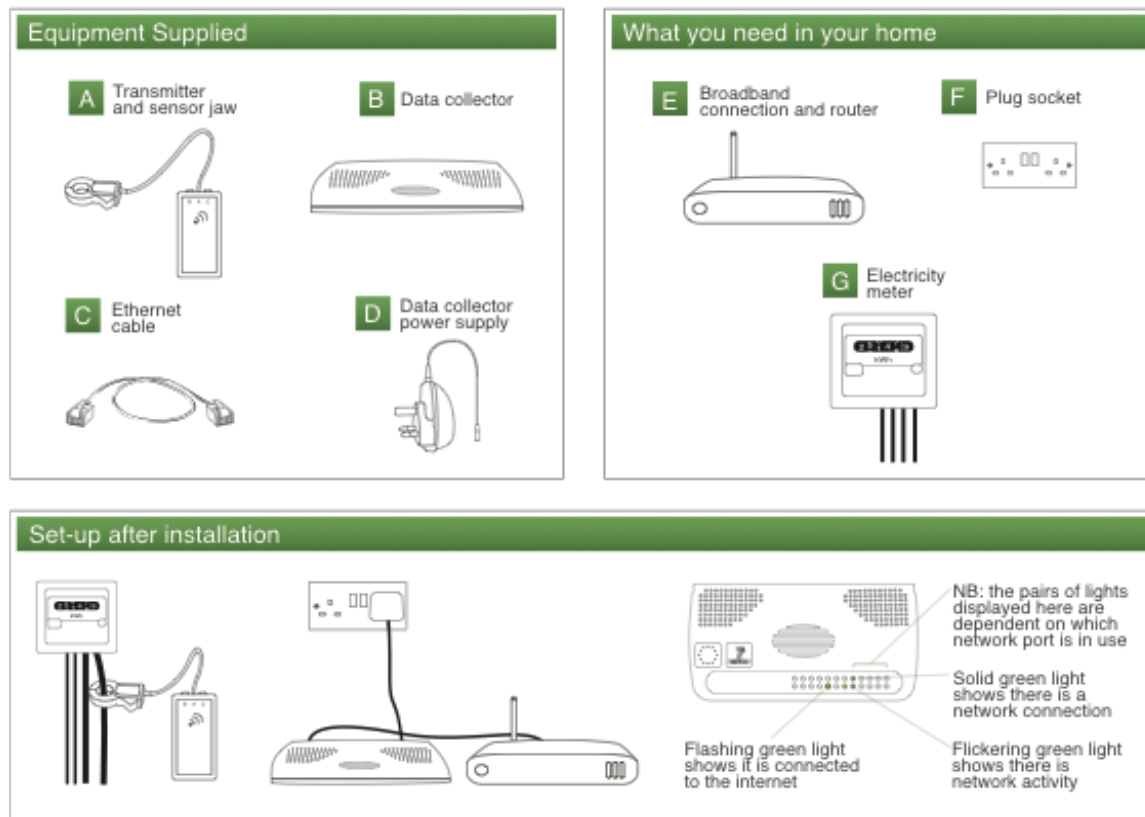
Future Work

A very functional and stable system has been delivered and there has been a lot of interest in expanding user base to bring benefits to a larger number of people. We have also uncovered some interesting areas to research due to the practical limitations of current technology. These areas include:

- scalability to millions of households, the ability for a centralized system to collect analyze and provide data via an Internet based web service
- security models to enforce privacy, legitimacy of data sources and rights to consume
- computational languages to take advantage of the tiered data system that has been designed
- Cycle 2 will also bring new challenges in interfacing with control systems, semantic systems and non-browser based clients.
- Implications for other Work packages
- This deliverable affects all work packages as it represents the physical deployment of the project.

Appendix I – Installation Guide

Equipment Installation Guide



Install the clamp

Open the battery compartment on the transmitter and sensor jaw **A** and remove the plastic around the batteries. A red light should then flash on the front of the unit.

Locate your electricity meter **G** and the fourth cable from the left - most meters have 4 cables, some have 6 but it should always be a thick round wire fourth from the left. Ensure there is enough space around the wire then position the clamp around the wire making sure the jaw is fully closed (see diagram). To work most effectively the jaw should sit perpendicular to the cable i.e. not hanging off the wire. If you have an electricity meter with a different cable set-up note that you will not damage any of the DEHEMS equipment by trying different cables. If still in doubt ask an electrician.

Connect the data collector to the internet

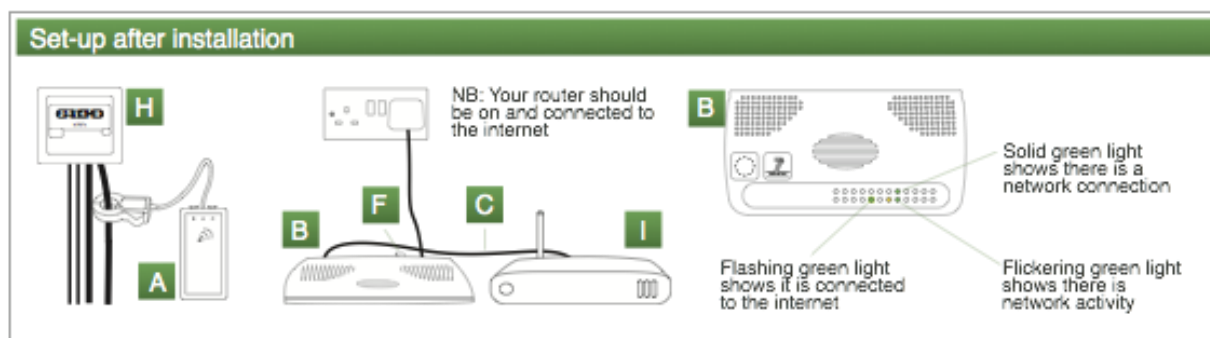
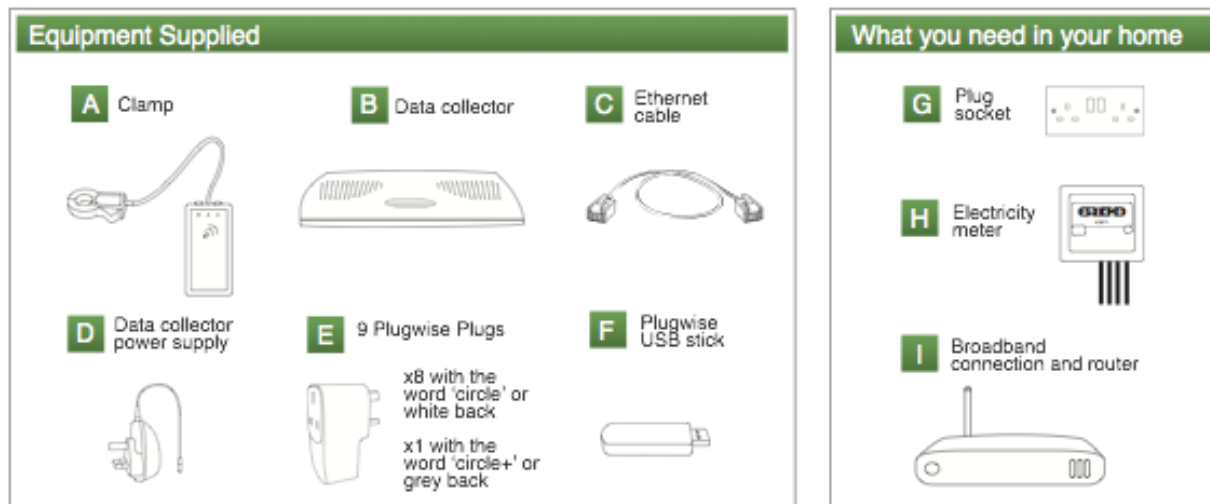
Connect one end of the ethernet cable **C** to the data collector **B** using any of the 5 sockets and plug the other end into your broadband router.

Plug the power supply **D** into the back of the data collector **B** and a suitable plug socket. The data collector should switch on automatically. When starting up 1 green light and 2 amber lights will briefly come on then switch off and there will be no lights showing for around 30 seconds. 1 amber light will then come on and after a pause 3 green lights will appear. When the far left constant green light starts to flash the equipment is working and connected to the internet.

Register on the DEHEMS website

To register to view your electricity usage go to www.dehems.org where you will be prompted to register an account. Whenever you want to view your electricity usage login to www.dehems.org from any computer.

Equipment Installation Guide for Plugwise users Page 1



Install the clamp

Open the battery compartment on the clamp **A** and install batteries. A red light should then flash on the front of the unit.

Locate your electricity meter **H** and the fourth cable from the left - most meters have 4 cables, some have 6 but it usually is a thick round wire fourth from the left. Ensure there is enough space around the wire then position the clamp around the wire making sure the jaw is fully closed (see diagram). To work most effectively the jaw should sit perpendicular to the cable i.e. not hanging off the wire. If you have an electricity meter with a different cable set-up note that you will not damage any of the DEHEMS equipment by trying different cables. If still in doubt ask an electrician.

Connect the data collector to the internet

Connect one end of the ethernet cable **C** to the data collector **B** using the socket with WAN written underneath then plug the other end into your switched on broadband router.

Plug the power supply **D** into the back of the data collector **B** and a suitable plug socket. The data collector should switch on automatically. When starting up 1 green light and 2 amber lights will briefly come on then switch off and there will be no lights showing for around 30 seconds. 1 amber light will then come on and after a pause 3 green lights will appear. When the far left constant green light starts to flash the equipment is working and connected to the internet.

Equipment Installation Guide for Plugwise users Page 2



Register on the DEHEMS website (Plugwise users)

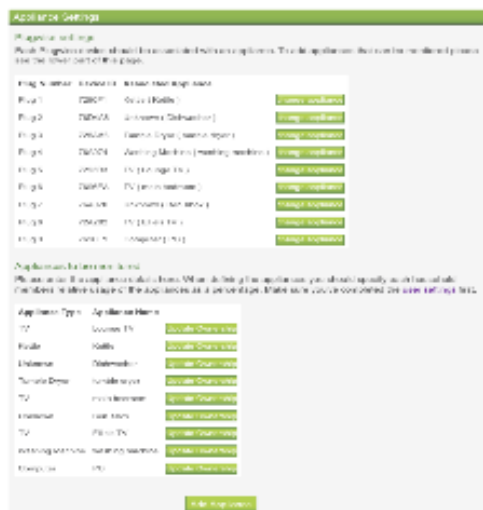
Please note there is no need to plug in any of the plugwise plugs or the USB stick until after you have registered your account and equipment details on www.dehems.org.

To register go to www.dehems.org and click the register button on the left.

- Enter the data collector ID (12 digits long on a sticker on the back of the device)
- For measurement type select 'Clamp and Plugwise'
- For meter type select Omnima
- Enter the clamp serial number (on the back of the clamp starting in OM, converting it to decimal if it is a hexadecimal number (as explained in the briefing))
- To associate the plugwise plugs type in the 6 digit identifier on the back of the plug e.g. 4692AB for each of the 9 plugs. Please note one of the plugs will have the word 'circle+' (as opposed to just 'circle') on it or will have a grey back (instead of white) and this plug should be used for an appliance that is located near to the data collector i.e. in the same room if possible.
- Select your living lab from the drop down menu.

Follow the rest of the account setup and household profile to complete the first stage of registration.

As a plugwise user you will then need to login to DEHEMS and go to My Settings > Household Settings > User Settings and enter the details of each occupant.



Then go to My Settings > Household Settings > Appliance Settings and press the 'Add appliance' button at the bottom. You will be prompted to add details about each appliance you will be monitoring through plugwise. If you are unsure of any details e.g. the model, type the word 'unknown' in the box. At the bottom will be a list of occupant names, for each appliance you will need to assign an approximate percentage use to each person and then click the 'Add appliance' button. Repeat this process until you have 9 appliances listed under 'Appliances to be monitored'

At the top of the page under Plugwise settings you will have a list of 9 Plugs with their device IDs. For each plug you will need to click the 'change appliance' button and select from the drop down one of the appliances you gave details for.

Appliance settings page after the 9 appliances have been assigned

Setting up the Plugwise equipment

Take the Plugwise USB stick **F** and place it into one of the USB sockets on the back of the Data Collector **B**. Now take each plugwise plug and plug the associated appliance (as listed on the 'Appliance Settings' page) into it and into the mains socket. You are now set up and you can view your individual appliance usage from the 'My Appliances' page of www.dehems.org

Appendix II: Sensor and Data Collector Images



Figure 10 Sensor and Transmitter

New current sensor and transmitter developed with 12-bit analogue to digital convertor and more accurate clamp sensor.

6.3 System Deliverable for Cycle 3



Figure 11 Radio (433 MHz) receiver

Radio (433 MHz) receiver is now inside of the data gateway which then interfaces with the home router to send data to servers through the Internet.



6.3 System Deliverable for Cycle 3

Figure 12 Circuit board for the data gateway device based on the Infineon processor



Figure 13 Back of data gateway board showing available ports.

DC power in, 2x USB ports and 5x RJ45 ethernet ports capable of 10/100 Mbps speeds.



Figure 14 Plugwise pass through plug measuring appliance level usage.