DELIVERABLE

Project Acronym: EPIC

Grant Agreement number: 270895

Project Title: European Platform for Intelligent Cities

Deliverable 3.3 Smart Environment Energy Monitoring: Testing Report

Version: 1.0

Authors:

Keith Osman (BCU)

Reviewers:

Wouter Van den Bosch (iMinds)

Wilhelm Stoll (IBM)

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	X
N	Nature	
C	Confidential, only for members of the consortium and the Commission Services	

Revision History

Version	Date	Author	Organisation	Description
0.1	19/05/13	Keith Osman	BCU	Initial draft
0.2	23/05/2013	Wouter Van den Bosch, Wilhelm Stoll	iMinds, IBM	Review remarks
0.3	25/03/13	Keith Osman	BCU	Updated version and put into template
0.4	25/05/2013	Wouter Van den Bosch, Wilhelm Stoll	iMinds, IBM	Final review remarks
0.5	27/5/2013	Keith Osman	BCU	Final Version for consortium approval
0.6	29/5/13	Keith Osman	BCU	Incorporated comments from Michel Dirkx, Deloitte
1.0	29/5/13	Keith Osman	BCU	Final Version

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Table of Contents

List of Abbreviations **4**

1. Introduction **5**

2. Overview of Technical Requirements **7**
2.1 Collection of Domestic Energy Data 8
2.2 Public Building Energy Data Collection 8
2.3 Data Flow Paths 9

3. Technical Testing Procedure and Results **10**
3.1 Technical of Domestic Energy Data 10
3.2 EnviR to WebSphereMQ Communication 11
3.3 WebSphereMQ to Generic Monitoring Framework Communication 12
3.4 WebSphereMQ Subscriber Testing 12
3.5 Testing of Public Building Energy Data Acquisition: Manchester 13
3.6 Technical Testing of Energy Data Web-Services 14
3.7 Technical Testing of Domestic Energy Dashboard 16
3.8 Technical Testing of Domestic Energy Dashboard in Tirgu-Mures 18
3.9 Technical Testing of Public Buildings Web-Services and Energy Dashboard 18
3.10 Tirgu-Mures Deployment: Public Buildings Data Web-Services 19

4. Conclusions **21**

List of Figures

Figure 1: Schematic of Smart Environment Pilot 7

Figure 2: EPIC Domestic Energy Monitoring Schematic 9

Figure 3: Data-flow Paths for Domestic and Public Buildings Energy Data 9

Figure 4: Testing Steps for Domestic Energy 11

Figure 5: Testing of Public Buildings Data Flow 14

Figure 6: Domestic Energy Web Services 15

Figure 7: Domestic Energy Dashboard 16

Figure 8: Public Buildings Web Services 18

Figure 9: Public Buildings Dashboard Comparing 7-Day View of Two Manchester Buildings 19

Figure 10: Public Buildings Dashboard Showing Data From A School in Tirgu-Mures 21

List of Abbreviations

BCU	Birmingham City University
BMS	Building Management System
EPIC	EU Platform for Intelligent Cities
GMF	Generic Monitoring Framework database created by BCU for energy data storage
iMinds	Interdisciplinary Institute For Broadband Technology
MQTT	Message Queuing Telemetry Transport
NAV	Navidis

1. Introduction

The Smart Environment pilot is quite different to the other two pilots and offers much less of the user-interaction found in Relocation and Urban Planning. Smart Environment focuses instead on automating the collection and processing of electrical power data from homes and energy usage data from smart-meters found in public buildings. The data is stored in a large relational database termed the Generic Monitoring Framework (GMF), developed within EPIC specifically for the storage of data from a myriad of sensor types.

Smart Environment was the sole pilot where Internet of Things (IoT) data connectivity protocols were implemented, through the use of the well-known and open-source Message Queuing Telemetry Transport (MQTT) protocol v3.1. This was used for the transport of real-time power data from homes to the Generic Monitoring Framework (GMF) database created by BCU for the storage of energy data. The other pilots did not have a real-time data collection requiring, instead making use of quasi-static third-party and city data; Google maps and 3D city models created by Navidis.

The two energy dashboards offered by smart-environment, one for domestic premises and one for public buildings are therefore relatively simple, providing graphical and numerical indications of instantaneous power and energy used. For domestic premises the energy consumed calculating by integration of the power profile the energy consumed within the display window width. User interaction is limited primarily to selection of the data viewing periods and to the selection of property types or public buildings for comparison.

Whilst the dashboard portlets may be simple, the web-services that expose energy data from the GMF to the energy dashboard and latterly to Relocation and Urban Planning for the integrated scenario are extremely complex and carry out significant amounts of data handling. The technical complexities associated with data acquisition and data transport, particularly for the domestic premises where power-usage updates is received in real-time have been challenging and required that the BCU technical team worked closely with IBM to design and implement a completely new architecture for the domestic energy solution, including selection of the sensors; implementing the internet bridge and data transfer protocols using the IoT protocol MQTT to publish data and the configuring of a message-broker to receive the energy data.

The significant delays introduced by rebuilding smart- environment had several consequences for the project:

1) The closed-group and open-group user-test cycles for smart-environment could not run concurrently with testing of the other pilots but lagged significantly behind.
2) As a result of the delayed testing it was necessary to compress the timescales for smart-environment user-testing, requiring the removal of at least one iteration

3) The open-test group for Smart-Environment needed 50 homes to be provided with energy monitoring equipment installed by Manchester's sub-contractor. This could not start until the technical function of domestic energy had been confirmed
4) The testing of smart-environment with Tirgu-Mures users was concurrent with Manchester users, requiring that domestic monitoring equipment and modified equipment for public buildings was provided to Tirgu-Mures by BCU.

One unintended consequence of the need to create smart-environment totally within EPIC was that pilot integration became an integral part of the development of smart-environment rather than a separate phase of the project, as the BCU technical team started developing on the EPIC platform using a web-service and portlet architecture from the outset, rather than having to integrate into the EPIC architecture elements of pilots which existed prior to the start of EPIC. Therefore smart-environment was the only one of the three pilots to experience the development lifecycle of a technical team setting out to develop for the EPIC architecture, rather than porting an existing application onto the platform.

Because smart-environment had been implemented ground-up using newly created web-services and portlets with no reliance on legacy applications or components, implementation of energy monitoring in Tirgu-Mures for the proof-of-concept phase was actually simpler for smart-environment that for the other two pilots. As the energy dashboards contained no geographic information there was no requirement to create new city models or accesses new property lists or points of interest. However, additional complexity was encountered because it was necessary to create a hybrid solution for public buildings in Tirgu-Mures, which lacked the smart-meters of Manchester buildings. This required modified domestic energy hardware for acquisition of 3-phase electrical power data, which was visualising through the public buildings dashboard and web-services.

This document describes the testing of the Smart Environment pilot in both Manchester and Tirgu Mures, including the different set-ups such as public buildings and domestic energy. Chapter 2 of this document describes the technical requirements that were to be met by the Smart Environment pilot and Chapter describes the testing procedures adopted to ensure correct technical functionality of the data acquisition processes; the web-services and portlets that comprised the energy dashboards. The overall results are then summarised in Chapter 4.

2. Overview of Technical Requirements

The Smart Environment pilot has a level of technical complexity not found in the other pilots due to the requirement to collect real-time electrical power data from homes and energy data from the smart-meters in public buildings. The technical challenges associated with the domestic energy data dashboard were compounded by the withdrawal of Hildebrand who had brought into the project deliverables from a previous FP7 project DEHEMS, where Hildebrand had worked as a technical partner with Manchester. When Hildebrand left the project at M21, none of the work they had done could be reused, in part to its dependence on commercial assets owned by Hildebrand.

The BCU and IBM technical teams worked closely together to design and implement a complete end-to-end system for collecting and storing the energy data updates received from each home every six seconds. The requirement to collect, store and post-process significant volumes of energy data required the design of a back-end database that would be suitable for both domestic energy data and the public buildings energy data and be readily extensible for other sensor data types. The architecture of the database, termed the Generic Monitoring Framework (GMF) allows the storage of a wide range of sensor data. The GMF was developed using the open-source database MySQL and was implemented off the EPIC platform on a server located at Birmingham City University. Web-services to insert data into and extract data from the GMF were developed on the GMF server and exposed to the EPIC platform through "helper" web-services hosted on the EPIC platform. This design follows the standard EPIC solution architecture, where 3rd party data is stored off-platform and exposed by 3rd party web-services running off-platform, accessed by web-services running on the EPIC platform. A schematic of the resulting smart environment pilot can be found is Figure 1.

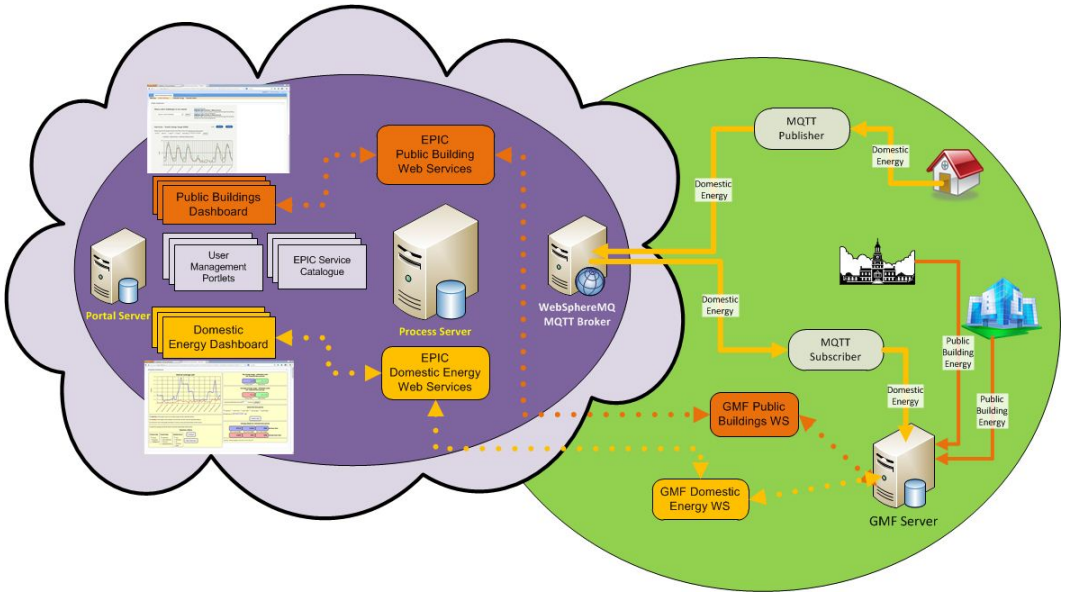


Figure 1: Schematic of Smart Environment Pilot

2.1 Collection of Domestic Energy Data

For domestic energy, the project team was required to develop an in-home solution to capture domestic energy data and relay this to the GMF for storage. Well proven commercially available EnviR monitor manufactured by Current Cost Ltd was used for the in-home capture of electrical power, using a wireless inductive clamp meter attached to the consumer unit and a stand-alone LCD monitor for display of energy data. Because this had no in-built connection to the Internet, the BCU and IBM technical teams worked together to implement an internet data bridge using a combination of customer-firmware in a commercial mini-router and the MQTT protocol for data transfer, supported by the installation on the EPIC platform of the IBM MQTT message broker, WebSphereMQ.

The Internet of Things (IoT) functionality contained within the DoW was satisfied by MQTT, which is well-proven open-standard protocol for internet scale communications, ideally suited to low-power embedded sensors and other devices. MQTT is a Publish-Subscribe or "Pub-Sub" architecture, where a message broker sits between the data sources, namely the publishers and the data consumers, namely the subscribers. Pub-sub architecture is ideal for large-scale networks where the publisher is not required to have knowledge of the subscriber, but simply has to publish the data to the message broker, which receives the data-flows from the publishers and relays these to the appropriate subscribers.

The clamp meter does not measure energy consumed but measures the electrical power flowing into the household, proving a reading of instantaneous power in Kw once per six seconds. The power readings are published MQTT messages to the WebSphereMQ message broker on the EPIC platform once every six-seconds, equating to 14,400 power-usage updates per home per day. Within the scope of the EPIC project, it was planned that the homes of at least 10 closed-group users and 50 open-group users would be allocated the necessary monitoring equipment for connection to the smart-environment pilot. A schematic of the domestic energy collection is shown in Figure 2.

2.2 Public Building Energy Data Collection

For public buildings, energy data is normally received as a daily batch update, typically as an email with a spread sheet or text file attachment generated automatically by the remote Building Management System (BMS). A separate daily report is received from each monitored building and the format of the data and form of the attachments can vary significantly, depending on the originating BMS. Within the scope of the EPIC project, it was planned that between 4 and 8 buildings would be connected, including buildings in Manchester and Tirgu-Mures, the latter during the proof-of-concept phase of the project. The public building data is stored in the GMF by the applications running on the BCU GMF server which process the incoming energy data as received.

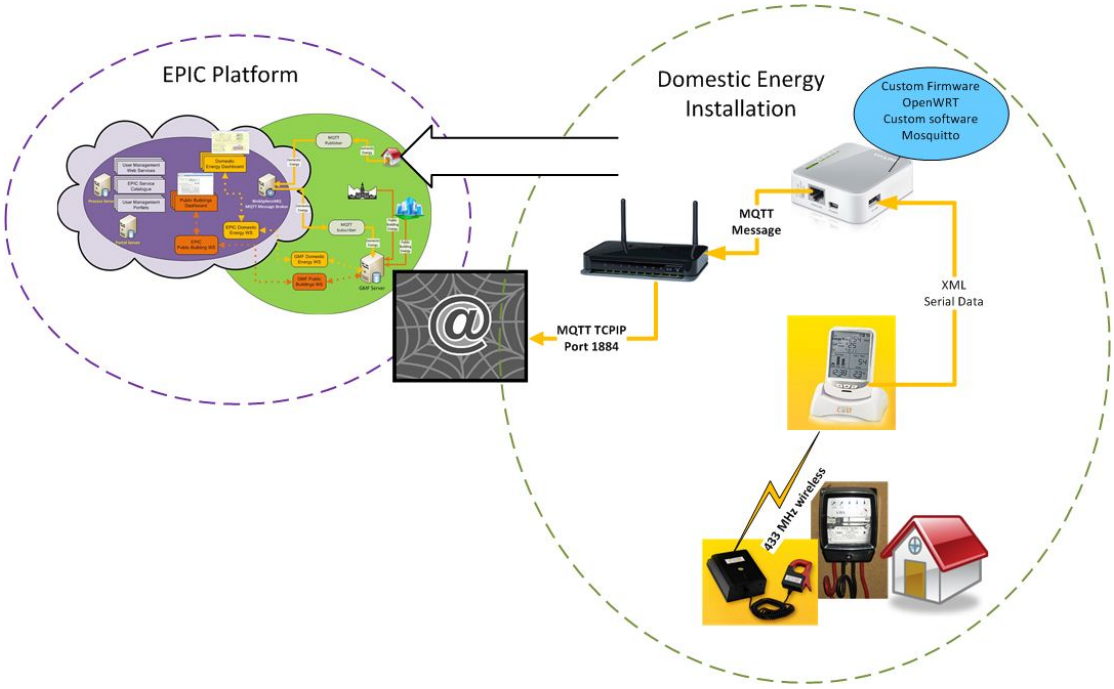


Figure 2: EPIC Domestic Energy Monitoring Schematic

2.3 Data Flow Paths

The data-flow paths for the two separate elements of the Smart Environment Pilot are shown below in Figure 3.

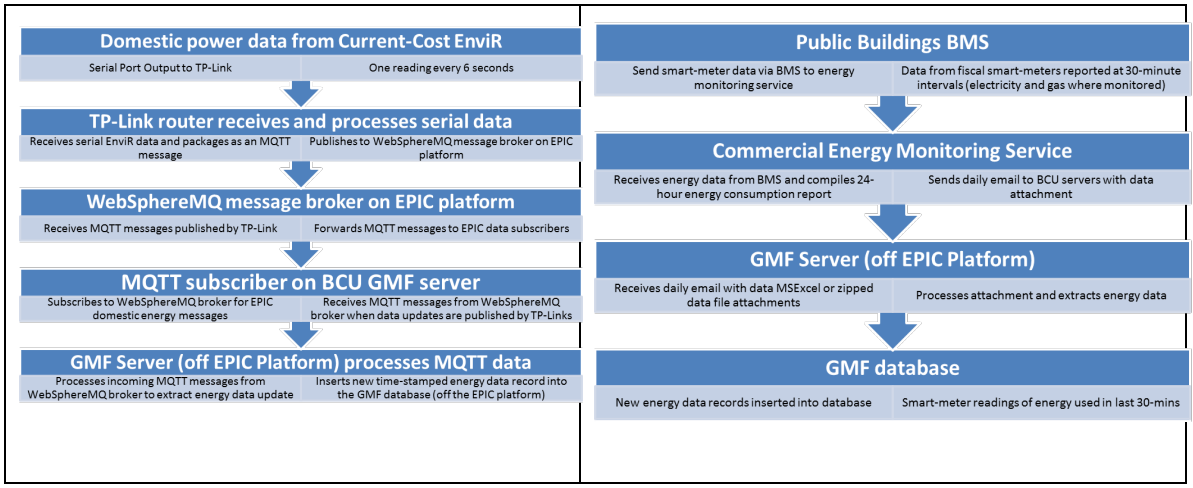


Figure 3: Data-flow Paths for Domestic and Public Buildings Energy Data

The data flow paths for domestic energy and public buildings were entirely separate although the end-point of the data was the same, namely it was stored in the GMF used for all Smart Environment energy data. This meant that testing of the two data flows paths could be undertaken independently.

3. Technical Testing Procedure and Results

This section explains the testing procedure adopted for each element of the Smart Environment pilot. The domestic energy data and public buildings data were completely separate data flows so could be tested independently. Once the domestic or public buildings data had been successfully stored in the GMF it could be retrieved and further processed by the relevant web-services. Although the GMF stored all energy data used by the Smart Environment pilot, the portlets and web-services were completely independent of each other and were developed in parallel by two separate technical teams.

3.1 Technical of Domestic Energy Data

The modular nature of the domestic energy data chain allowed testing of individual elements in the path prior to integration of the end-to end-signal path, as shown below:

Signal Path Element	Development & Test Method	Final Deployment
Wireless clamp meter to EnviR	Carry out standard EnviR configuration using LCD display	Standard EnviR installation in user's home
EnviR to TP-Link	Check EnviR serial-port output using serial-USB cable and PC	Serial to USB cable connected between EnviR and TP-Link
Decoding of EnviR XML power data messages	PC based software written in C. Subsequent compilation into TP-Link mini-router.	Added customer processing code to TP-Link firmware
Encoding of power data into MQTT message, adding MAC address and UTC timestamp and publishing to MQTT broker	Prototype on PC using Mosquitto MQTT broker installed on PC. Subsequent compilation into TP-Link mini-router.	Added Mosquitto library to TP-Link firmware. TP-Link connected to user's broadband router for internet connectivity.
Addition of router diagnostics to TP-Link firmware via re-purposing of LEDs to show MQTT connection	Prototype and test in TP-Link connected to a PC	Final firmware flashed into TP-Link for standalone operation. LED shows connection to MQTT broker on EPIC platform
Monitoring of MQTT messages received by WebSphereMQ	Standard remote login and diagnostics on WebSphereMQ	Watchdog timer added to WebSphereMQ for auto-reset in case of problems
Subscriber to MQTT power	Prototype and test on PC using	Code executed and tested

messages	Mosquitto MQTT broker installed on PC.	subsequently on GMF server
Insertion of energy data from subscriber into GMF	Prototyped on PC. Feed validity confirmed from valid MAC	Code executed and tested subsequently on GMF server
Monitoring of Domestic Sensor energy data feeds	Added Status tab to Domestic Energy Portlet Admin view	Web-service running on GMF platform provided last update time for each domestic sensor
End-to-End testing – user's data feed	Blue trace shows user's power profile on Domestic Energy dashboard and timestamp of latest data update (UTC)	Integral part of domestic energy dashboard deployed on EPIC platform.
Status of MQTT broker and GMF database	Gaps in red average trace indicate loss of data from GMF	Either WebSphereMQ down or GMF down – data lost

Figure 4: Testing Steps for Domestic Energy

3.2 EnviR to WebSphereMQ Communication

The TP-Link MR-3020 uses the Open-WRT firmware, a Linux distribution for embedded devices http://openwrt.org) which means that the router firmware can be modified and additional code added to change the functionality of the device. To act as the gateway between the serial data output from the EnviR and the IP address of the WebSphereMQ broker, the firmware was required to receive the serial data; parse it to extract key information; add a UTC timestamp and the MAC address of the router which was in effect the site ID the serial and package this data as an MQTT message and publish it to the broker. The TP-Link was connected between the EnviR and the user's broadband router or home network using the serial-USB cable supplied with the EnviR and a standard Ethernet cable. The software to parse the incoming serial data was written in C language and could be readily tested on a PC before being included in the image for the TP-Link. Mosquitto is a freely downloadable and well-proven open-source MQTT v3.1 broker (www.mosquitto.org) and could be readily included into the TP-Link firmware image to provide the MQTT publishing required.

WebSphereMQ is a telemetry add-on to WebSphere and is a well proven IBM commercial product and add-on to WebSphere. This was installed on the EPIC platform and configured by IBM to accept the EPIC energy data messages published by the TP-Link. Testing of the end-to-end domestic energy data publishing solution, i.e. EnviR to WebSphereMQ, was accomplished by monitoring the messages received by the WebSphereMQ broker to ensure that these were received correctly and intact.

3.3 WebSphereMQ to Generic Monitoring Framework Communication

The GMF subscribes to EPIC energy messages received by the message broker, and these are relayed by the broker as they are received. MQTT defines three levels of Quality of Service (QoS), which controls how diligently the broker/client will try to ensure that a message is received:

- QoS 0: The broker/client will deliver the message once, with no confirmation.
- QoS 1: The broker/client will deliver the message at least once, with confirmation required.
- QoS 2: The broker/client will deliver the message exactly once by using a four step handshake.

Messages may be sent at any QoS level, and clients may attempt to subscribe to topics at any QoS level. Higher levels of QoS are more reliable, but involve higher latency and have higher bandwidth requirements and may involve significant local caching of messages on publisher or broker where connectivity is disrupted.

For domestic energy monitoring, data messages are created every six seconds by each EnviR and in the context of the application, namely understanding cumulative energy consumption, the non-delivery of individual data messages is not consider problematic and guaranteed delivery of messages is not mission critical . Domestic energy usage is calculated by integrating a stream of six-second power readings over a five-minute period, nominally fifty readings, to calculate the average energy used in that period, so the effect of a few missing readings is minimal. It is also important, given the high volume of MQTT messages, that if the remote GMF server were to go off-line for any reason, the WebSphereMQ broker is not required to queue potentially tens of thousands of messages until connection to the GMF is restored, which it would be required to do if a guaranteed delivery QoS was used. Furthermore, from prior experience it is known that many domestic broadband connections may be rather unstable, and subject to periodic disruptions. Again, due to limited internal memory, if the connection between the TP-Link and Broker is disrupted then the six-second data updates from the EnviR are discarded until the data connection to the broker is restored. Many of the TP-Links have continued to send data reliably for several months, and have recovered from drop-outs in the user's broadband without user intervention or resetting.

3.4 WebSphereMQ Subscriber Testing

The subscriber running on the GMF server could be readily monitored to check incoming MQTT messages from the broker and to ensure that these were parsed appropriately and data was inserted currently into the GMF database. The reliability of message delivery from the Broker to the GMF, could be readily checked by comparing the WebSphereMQ logs with the data received by the subscriber and inserted into the GMF database.

The real test has been the stability of the end-to-end connections between each EnviR and the GMF database. Whilst this is difficult to measure directly, the accumulated data in the GMF can be used to gain a measure of stability. The domestic energy portlet displays two traces, namely the user's data where they are a data publisher, displayed as a blue trace and the average data calculated from other similar users displayed as a red trace. Gaps in the blue data appear where missing data means that the five-minute data average cannot be calculated and displayed. This is normally due to disruption of the user's broadband connection meaning that data is not published to the broker, as QoS 0 is used. If the MQTT broker is offline for any reason, then none of the TP-Link messages will be received, so calculation of the average energy of all properties during this period will be impossible. This will show up on the dashboard as gaps in the red average energy trace.

To assist with diagnosis on the domestic energy data chain, a status portlet, visible only to administrators was created, which displayed the status of each of the domestic sensors, displaying the MAC address and timestamp of the last update received by the GMF and a colour-code showing green for online and red for offline. This was designed particularly for the user-test managers who could check the status of a user's energy feed if they were reporting problems with missing energy data on their energy dashboard.

3.5 Testing of Public Building Energy Data Acquisition: Manchester

Unlike the domestic energy data, which required sensor installation and custom firmware developing for the TP-Links, the public buildings used data that was already being collected by the BMS systems and by the fiscal smart-meters commonly used for utility monitoring in public buildings. The Public Buildings dashboard re-used data contained in customised daily reports which were created in the Manchester BMS systems, or in the third-party data aggregation sites used by MCC. Four Manchester buildings were monitored, namely two Manchester City Council buildings including the Town hall and two enterprise academies, all of which were connected to the Manchester BMS network.

As shown in Figure 4, the BMS systems created a separate daily report for each building, and this was send as an email with an MS Excel attachment to an email account monitored by the BCU GMF server. When an email was received, the attachment was removed and processed and the data records were inserted automatically into the GMF database.

Testing of the processing scripts was possible using simulated data and subsequently real data feeds. It is a relatively simple task to automate but long-term stability is 100% reliant on several factors outside BCU's control, including:

- The safe receipt by the BMS systems and external sites of data transmitted from the Manchester smart-meters at 30-minute intervals
- The reliable creation of email reports by the Manchester back-office systems
- The sending email address of the reports remaining consistent to allow safe passage of emails through the BCU firewalls and forwarding to the correct internal addresses

- The data format in the spreadsheet exported by the remote BMS remaining consistent

Signal Path Element	Development & Test Method	Final Deployment
Confirm data report format from Manchester BMS	Manually create a report by remote log-on to BMS	Daily report created automatically
Extraction and processing of report data	PC based software development and test	Deployed to GMF server
Automated processing of emails with attached report	Create processing scripts on GMF server. Send email with data attachment from PC	Deployed to GMF server for auto-processing of emails received from BMS (four received daily)
Insertion of Public Buildings energy records into GMF database	PC based software development and test for SQL queries	Deployed to GMF server

Figure 5: Testing of Public Buildings Data Flow

In general this process has proved highly reliable, with daily reports running as scheduled in the Manchester BMS systems and being emailed and processed by the GMF server. The only problems encountered to date have been due to a change in sender email address meaning that data was not delivered properly via the BCU mail system or errors in the Manchester BMS configuration leading to reports being missed.

However, because all historic data is held in the Manchester BMS and missing data can be readily recovered by manual initiation of a custom report covering the missing period by remote logon to the BMS system. This has also allowed several years of historic data for the period before the start of the EPIC activity to be imported into the GMF.

3.6 Technical Testing of Energy Data Web-Services

Due to the differences in domestic energy data (instantaneous power readings in Kw) and public building data (total energy consumed in 30-minute time frames), separate sets of web-services were created for domestic and public building energy data streams. The domestic energy web-services had not only to calculate and present the smoothed power-profile for the specified property, but also to calculate the total energy used across the specified time-period by integrating the power curve. The domestic web-service was also required to calculate the average smoothed power profile of all properties to provide the red trace on the domestic dashboard and integrate this to calculate the average energy consumed. Domestic energy data was a particular technical challenge due to the large

volumes of data, as 14,400 power readings were received per house in each 24 hour period.

Once initial data flows had been established from two test properties and data was being stored successfully in GMF, the web-service operation on real-data could be tested. There were four web-services used for domestic energy as shown in Figure 6 below:

Web Service	Usage	Development & Test Method	Final Deployment
getSensorAveragesOverPeriodByID	Domestic Energy Data	Averages power readings from specified sensor. Web-service developed on PC and migrated to GMF server. Tested with real data from GMF	End-point web-service runs GMF server accessed by helper per web-service on EPIC platform
getDomesticSensorAverages	Domestic Energy Data	Averages power readings from all matching domestic sensors. Web-service developed on PC and migrated to GMF server. Tested with real data from GMF	End-point web-service runs GMF server accessed by helper per web-service on EPIC platform
getTotalConsumedOverPeriodBySensorID	Domestic Energy Data	Calculates energy consumed by integrating power profile for specified sensor. Web-service developed on PC and migrated to GMF server. Tested with real data from GMF	End-point web-service runs GMF server accessed by helper per web-service on EPIC platform
getAverageConsumedOverPeriodAllDomesticSensors	Domestic Energy Data	Calculates energy consumed by integrating power profile for all domestic sensors. Developed on PC Initial tests with simulated data Migrated to GMF server. Tested with real data from GMF	End-point web-service runs GMF server accessed by helper web-service on EPIC platform servers

Figure 6: Domestic Energy Web Services

It is important to explain that the Current-Cost sensor provided an instantaneous measure of electrical power flowing into the house at six-second intervals, not energy consumed. Whilst this enabled the power profile to be plotted and the effects of individual high-power appliances such as kettles, electric showers and washing machines to be readily visualised, it did not help with an estimation of the energy consumed. Domestic energy plotted the total power-profile of the house and calculated the energy consumed over the selected display period, providing also an estimate of energy cost if the user entered their per-unit charge. The user was able to select one of five different timescales for the plot and calculation window, namely 1 hour view; 24 hour view; 7 day view; 30 day view and

a 90 day view, which are the commonly used time periods for commercial domestic energy monitors, with 90 days representing a typical billing period.

3.7 Technical Testing of Domestic Energy Dashboard

Prior to release for closed-group and open-group user testing, the development team and others undertook technical testing of the portlets which delivered the domestic energy dashboard. Whilst the controls presented to the user were minimal, mainly to do with selection of the timescale for plotting and the property comparison indicators, a large amount of behind-the-scenes data processing was performed by the web-services. An example domestic energy dashboard view in shown in Figure 7.

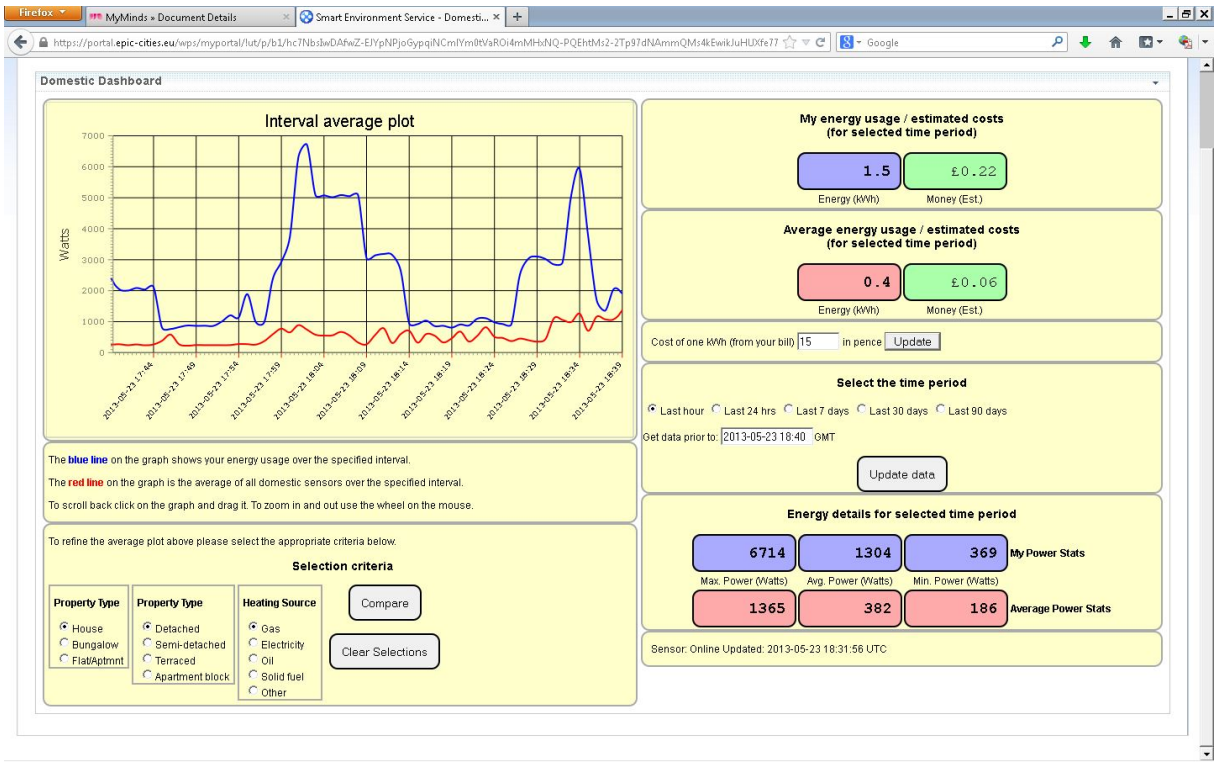


Figure 7: Domestic Energy Dashboard

The default view is the energy usage for the last hour, with the power profile calculate from averages over 1 minute periods plotted top left and the energy used displayed top right. Bottom right shows the display of maximum, mean and minimum power demand over the displayed period.

Compromises were required with all web services, but primarily with those concerned with domestic energy. If we consider a ninety day period, a single house would generate 10 x 60 x 24 x 90 = 1,296,000 separate power readings. Clearly there were limitations on what could be readily plotted and what could be processed in a reasonable time period to

ensure a user-acceptable loading period. The red line is generated from the average of all properties, which reduce in number as the user selects property parameters that represent their own house. Clearly the number of data points from which the average property profile is calculated increases with the number of users, and 50 properties would generate around 65 million readings in a 90-day period.

Domestic power data readings were collected once every 6 seconds, equating to 14,400 readings per property per 24 hour period. To calculate the energy used it was necessary to integrate the power measurements, done simply using an implementation of the trapezium rule. However, it was clearly important in terms of response time, to ensure that data processing was not excessive, leading to long load-times and response-times to user inputs. A different averaging period was used for the power profile and cost calculations, based on the time period selected, ranging from 1-minute for the 1 hour window to 1 day for the 90 day view.

The averaging has the effect of a low-pass filter on the data, with an increased smoothing effect on the longer view periods. This means that short-duration power spikes which appear in the raw data may be smoothed and ultimately become invisible when viewed on a longer duration window.

The averaging is implemented inside the web-service and the averaging period is passed in the call to the web-service, allowing an averaging interval of 1 minute, five minutes, one hour, 1 day or one week. Typically one-minute intervals are used for the 1 hour view; five-minute intervals are used for the 24-hour view and 1 hour intervals are used for 7 days and 30 days with one-day being used for the 90-day period.

Therefore a major consideration of technical testing once correct operation of the web-services had been confirmed was the trade-off between resolution and load-time for the different plots. Again it is important to reiterate that the domestic energy dashboard is NOT a billing application but to assist the user to identify where their pattern of energy usage across a period, to help understand the large energy drains associated with appliances in their house. Therefore loss of resolution with increased window width is acceptable, but clearly a bill calculation programme would need to maintain the underlying data resolution in order to calculate accurate the energy usage.

After the first phases of user testing, users reported that a scrolling and / or zooming function would be useful for the plot windows. The Dojo GFX cross-browser 2D graphics API was used for the plots, as this integrated very well with WebSphere portal server and provided good functionality. The scrolling ability required that the data loaded behind the window was wider than the display window and compromises were required to trade-off load time against ability to scroll and zoom.

3.8 Technical Testing of Domestic Energy Dashboard in Tirgu-Mures

The deployment of domestic energy monitoring in Tirgu-Mures was simple, using the same equipment as used for the UK users. This has worked correctly, but is subject to internet connectivity drop-outs, leading to data loss and sensors going off-line. As the equipment had been already proven in 60+ installations in the UK and one in Belgium, no Tirgu-Mures specific testing was needed.

3.9 Technical Testing of Public Buildings Web-Services and Energy Dashboard

The public buildings web-services were much simpler than those for domestic energy, primarily because the data received was the amount of energy used in the preceding 30-minute interval, as this data was calculated directly by the smart-meters or BMS sub-metering. The main requirement of the web-services was therefore to allow retrieval of data from a given sensor (meter) over a given range, but an option was provided to provide this as the energy consumed per hour or the energy consumed in the previous week, to allow meaningful comparisons between different buildings with user controlled granularity of data. An option to download the data as a comma-separated values (.csv) file was also provided for users so they could undertake further analysis outside of the Smart Environment pilot.

Web Service	Usage	Development & Test Method	Final Deployment
getWeeklySensorDataOverRange	Public Buildings Energy Data	Returns total energy used in specified week period for specified building. Web-service developed on PC and migrated to GMF server. Final testing with real data from BMS reports and GMF Modified to handle Tirgu Mures data	End-point web-service runs GMF server accessed by helper per web-service on EPIC platform
getHourlySensorDataOverRange	Public Buildings Energy Data	Returns a list of hourly energy usage figures over specified period for a given building. Web-service developed on PC and migrated to GMF server. Final testing with real data from BMS reports and GMF Modified to handle Tirgu Mures data	End-point web-service runs GMF server accessed by helper per web-service on EPIC platform

Figure 8: Public Buildings Web Services

The correct operation of the Public Buildings web-services for Manchester buildings was readily checked and verified as the known input data, delivered as an attached

spreadsheet, could be readily compared with the results obtained from the web-service and displayed on the portlet. Furthermore, energy data from the public buildings could be readily downloaded and checked against the original data. Because the data provided by the smart-meters was in the form of energy consumed per 30-minute window with a timestamp, most of the complexity required in the domestic web-services was not required for public buildings.

An example of the Public Buildings dashboard displaying data from two of the Manchester Academies is shown in Figure 9.

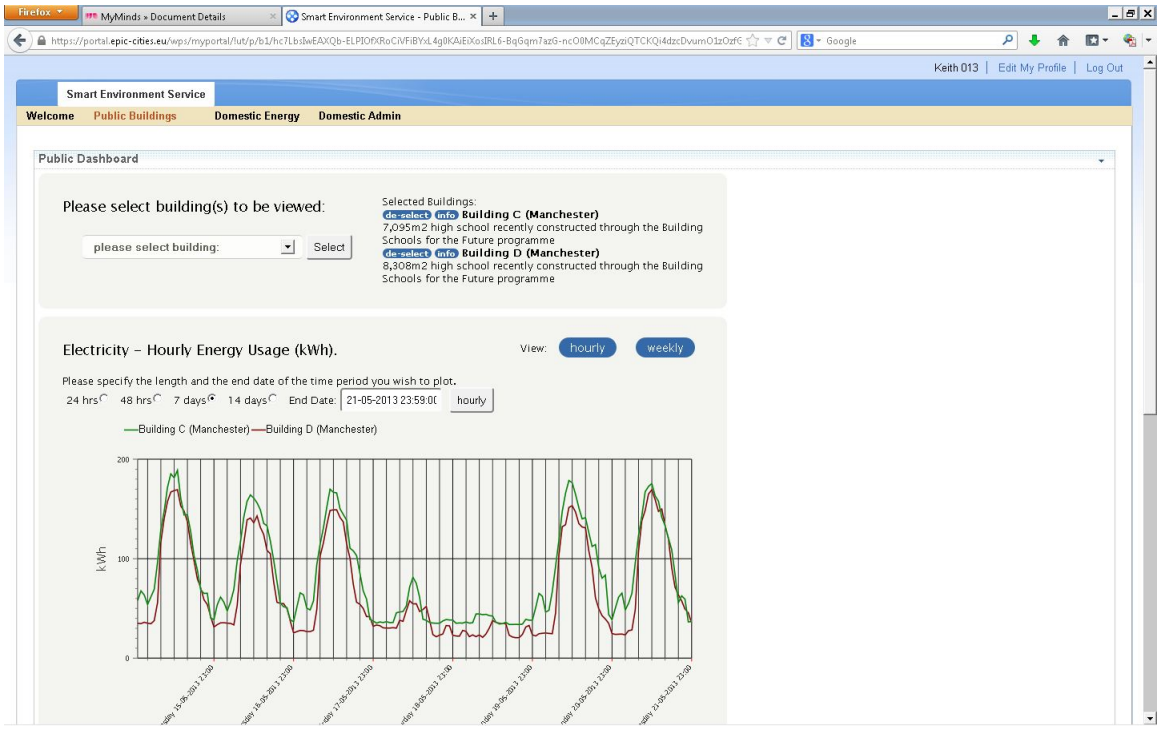


Figure 9: Public Buildings Dashboard Comparing 7-Day View of Two Manchester Buildings

3.10 Tirgu-Mures Deployment: Public Buildings Data Web-Services

Energy monitoring for public buildings for Tirgu Mures was made difficult because the buildings to be monitored did not have smart meters and were not connected to a BMS which could generate the daily energy usage exports as used for Manchester buildings. The interest of Tirgu-Mures was to compare the electrical energy used by three schools which had 3-phase electrical supplies as is normal for commercial buildings, other than the single-phase used in domestic premises.

The domestic hardware was therefore modified, as the EnviR transmitter could handle 3 separate current clamps, one per phase of a three-phase supply. The firmware of the TP-Links was modified to handle the data from the three inductive clamps (one per phase)

and the Tirgu Mures sensors were identified as Public Buildings in Tirgu Mures. The data from the Tirgu Mures public buildings was transported using MQTT and stored in the GMF but not combined with the other domestic data.

The Manchester public building data comprised energy-used in each 30-minute timeslot, yet the Tirgu Mures public building data was in the same format as the domestic energy, namely the electrical power readings received at 6-second intervals via the MQTT transport.

The Public Buildings web-services were therefore modified to become very smart web-services to handle both Manchester and Tirgu Mures data in a manner transparent to the user, rather than creating separate web-service variants for the Tirgu Mures public buildings. If the data for Manchester sensors was requested this was returned by the web-service in the normal way, from the 30-minute energy reading supplied by the Manchester BMS. If data was requested from Tirgu Mures public buildings sensors, the web-service processed the necessary 30-minute energy consumption figures from the 6-second Tirgu Mures public buildings power readings in the GMF by integration over the requested period. This allowed, for example, a Tirgu Mures building to be compared with Manchester building, which would not have been possible had separate dashboards with separate web-services been created for Manchester and Tirgu Mures.

The modification Public Buildings web-service demonstrated very clearly the advantages of using web-services to expose third-party data. It was not necessary for the user of the web-service to know that the underlying data formats of Manchester and Tirgu-Mures buildings were different, as the complexity of processing the Tirgu-Mures data has handled transparently to the user, who was presented with data in the same form irrespective of source. Figure 10 shows an example of the Public Buildings dashboard displaying data from a school in Tirgu Mures.



Figure 10: Public Buildings Dashboard Showing Data From A School in Tirgu-Mures

4. Conclusions

The Smart Environment pilot was the most complex of the three pilots in terms of data acquisition and data processing required. Technical testing of the Smart Environment pilot covered several parts of their operation: data acquisition; processing and storage; separate web-services for domestic and public buildings and the performance of the domestic energy dashboard and the public buildings dashboard:

Domestic Data Acquisition: The technical performance of the domestic energy monitoring using EnviR, TP-Link, and MQTT has been proven, with in excess of 26 million readings in the GMF. There are still outstanding issues with the stability of user's home broadband, as when their Internet connections are lost data is lost, appearing as gaps in their individual energy profile. This is, however, a well-documented problem and is not hugely problematic, as the Domestic Energy dashboard aims to help users understand real-time power demands energy usage rather than providing a smart-billing application, where any data loss would be problematic.

Public Buildings Data Acquisition: The automated processing of emailed reports with attached files containing energy data has been proven and works reliably. If for any reason the remote BMS does not send a daily report, a manual report can be run remotely to replace the missing daily updates.

Domestic Energy Web-Services: These were the most technically complex and carried out significant processing on the GMF server before the requested data could be returned to the calling service. The need to calculate averages from the 6-second power readings over periods from 1-minute to one-week introduced significant demands on back-end processing power.

Public Building Web-Services: For Manchester data these were relatively simple, as they calculated 1-hour and weekly cumulative energy consumption figures from the smart-meter data. For Tirgu Mures the web-service processed the data from Tirgu Mures public buildings EnviRs to calculate the equivalent hourly and weekly data as per Manchester buildings.

Domestic Energy Dashboard: This has changed little between Phase 1 and Phase 2 testing. The major change has been the implementation of scrolling and zooming on the plot window. This is helpful for users who can move backwards and forwards through the data, but has an associated overhead of increased load-time, due to the extra processing required by the web-services. This would merit some further attention in the future, with possible pre-processing of GMF data to create and store the averaged data rather than creating on the fly, to reduce processing overhead.

Public Buildings Dashboard: This changed significantly in layout but not functionality from Phase 1 to Phase 2, due to use of a single plot window with overlay plots rather than separate plot windows. This eases comparisons between buildings.

Ease of Redeployment: Because smart environment lacked the location-specific and geographical information of the other two pilots, deployment in Tirgu-Mures was largely confined to establishing the correct data feeds. The domestic energy monitoring equipment used in Manchester was supplied to Tirgu Mures households for installation. Public buildings required a modified form of the same equipment due to the lack of smart-meters in Tirgu-Mures. This required that the public buildings web-services were made smart so that they could consume either Tirgu Mures data or Manchester data and present this is a consistent way to the public buildings dashboard. The ease of doing this demonstrated the logic of separating data through web-services from display via portlets.