COSMOS

Cultivate resilient smart Objects for Sustainable city application

Grant Agreement № 609043

D7.1.2 Use Cases Scenarios Definition and Design (Updated)
WP7 Use cases Adaptation, Integration and Experimentation

Version:	1.0
Due Date:	31 December 2014
Delivery Date:	15 March 2015
Nature:	Research
Dissemination Level:	Public

Lead partner:	Hildebrand Technology Limited
Authors:	Joshua Cooper
Internal reviewers:	Sergio Fernández Balaguer Andres Recio Martin Saima Iqbal

www.iot-cosmos.eu

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° 609043

Version Control:

Version	Date	Author	Author's Organization	Changes
0.1	15 Jan 2014	Joshua Cooper	HILD	Initial additions for UrbisAPIs
0.2	15 Feb 2015	Joshua Cooper	HILD	Updated use cases with Camden consultation
0.3	12 March 2015	Joshua Cooper	HILD	Circulated draft for submission
1.0	15 March 2015	Joshua Cooper	HILD	Draft for submission

Annexes:

Nº	File Name	Title
1	D7.1.1.doc	D7.1.1 Use Cases Scenarios Definition and Design (Initial)

Table of Contents

1 Introduction

This document is an update to the initial Use Case Scenarios Definition and Design document published in Year 1 of the COSMOS project. It further develops the use cases for COSMOS as well as showing how those use cases have been tested with end user input. We go further in showing three areas of focus for COSMOS application partners:

1. how the IoTA reference architecture can be realised into a platform that offers the adaptability and scalability for smart cities; this effort is demonstrated through UrbisAPIs a prototypical smart city instance of COSMOS,
2. leveraging the peer to peer and low resource device goals of COSMOS as social objects through an open source radio level communications protocol named OpenThings and then integrating that with UrbisAPIs
3. creation of a scalable system for converting pull based public information into published events through work on VEPROT which will also integrate into UrbisAPIs

The goal of making a central feature of UrbisAPIs is to have a concrete place that COSMOS comes together as a COSMOS platform instance that end users can then participate in the innovation cycle. Year 1 set the general requirements, whereas Year 2 sets specific requirements for a platform and specific subsystems that can be used by end users.

Let us re-iterate key issues from Year 1. Smart City applications are wide ranging and have many practical implications for deployment:

- IoT promises to be an effective technology for higher quality, more efficient and new city services despite some of the implementation challenges.
- To limit project risk, cities that are using IoT are still doing so with “island” implementations, that is to say business functions are not integrated using IoT and IoT infrastructure is not being leveraged between business functions

From a city perspective, COSMOS has a mission of:

- improving the utility of IoT systems across multiple applications,
- reuse existing instrumentation,
- provide tools for creating new applications in this integrated paradigm and
- demonstrating new financial benefits in the applications deployed

These goals are not without challenges and much work will go into technology behind the scenes. It is our approach that Work Package 7 will present use cases in a progressive fashion, with a first iteration prioritising ease of implementation and foundation over complexity while developing two further iterations Deliverables 7.1.2 and 7.1.3 progressing both the sophistication and, by virtue of the our integrated approach, the impact.

Three cities are participating in the trial Madrid, transport, London energy, Taipei future integration and assistance in understanding the situation.

In this Work Package deliverable we will

- description of UrbisAPIs and show the functional design and features of the platform
- introduce new use cases from Camden and Taipei test beds
- show general features of city management and organisation

- update on the background material describing the current situation and available resources in each of the cities
- show the use cases and how they are enacted and modelled within UrbisAPI s
- describe the work that has been carried out in making devices that are COSMOS enabled, including work on the OpenThings protocol and how those are compatible with UrbisAPIs
- articulate some of the challenges that were discovered and any anticipated challenges ahead

The value of this deliverable should be to show how a city can use UribisAPIs and specifically implement city solutions with COSMOS as city management and application developers.

The outcome of this deliverable is to inform the project, but also give future Smart City projects some working models for delivery of projects.

2 UrbisAPIs

As the name implies UrbisAPIs is a platform focused on APIs for city services and represents an instance of the COSMOS platform. From the perspectives of the city test beds, this is where COSMOS can be used and managed for VE developers and application developers. It provides an ease of use for infrastructure owners and operators (both the city and citizen owned assets), virtual entity creators/maintainers and application developers.

This section is an initial product specification that shows the end user perspective of features and functions that can be used once they are registered. The scope for UrbisAPIs is multicity in that as a user your will be able to use services from multiple city infrastructures and model virtual entities that span cities.

Implementation of UrbisAPIs will continue through Years 2 and 3 of the COMOS project and some initial functionality is currently available. A product roadmap will be shown and with staging of features, however an iterative development methodology is being used.

UrbisAPIs can be found at http://www.urbisapis.com.

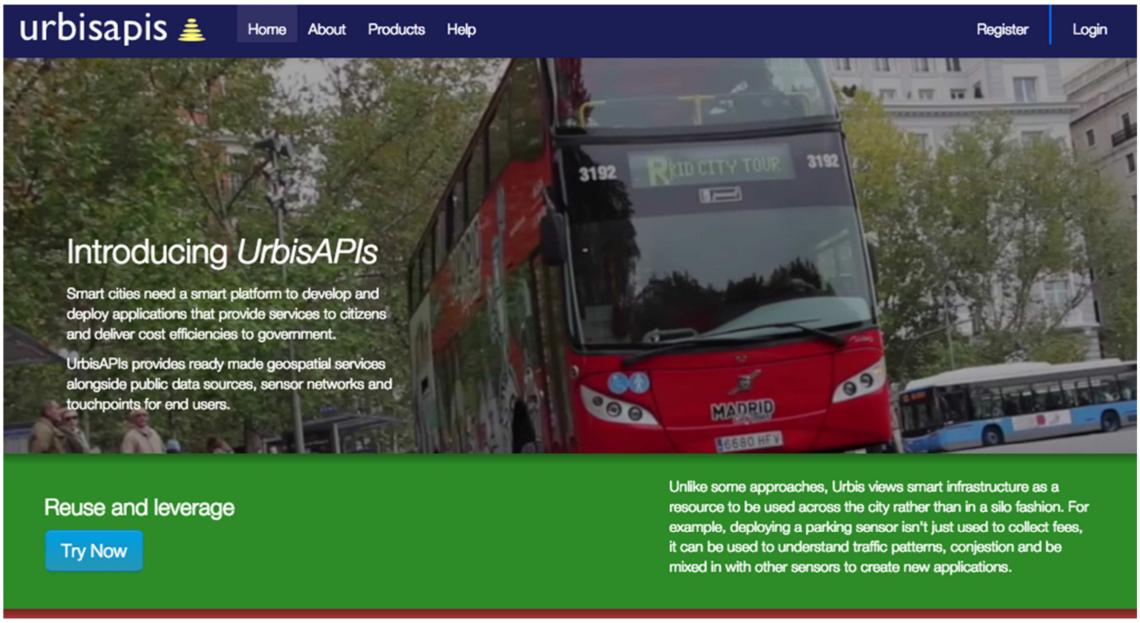

Figure 1 The homepage for UrbisAPIs found at http://www.urbisapis.com

2.1 Purpose

This section provides a design of the UrbisAPIs platform that forms an exemplar instance of the COSMOS project. The UrbisAPIs website provides a user interface for end to end functionality relating to COSMOS with unique functionality of coordination/integration of COSMOS subsystems, higher layer IoT functions, billing, revenue collection, operational management of services below the application layer and distribution assistance for applications.

To show the deployment of the system, the description of the overall system is broken into four deployment locations:

1. **Field assets** – incorporating all of the physical assets that are sensing, actuating and bridging communications; some field assets may have local control and peer level communication, examples will be shown in Section 8 where field asset capabilities are shown for UrbisAPIs; an important and complex field asset is one that is mobile (changing physical location of the asset) or may even be virtually mobile (a process representing a virtual entity may move to different field assets)
2. **Customer facing** – essentially these are applications, however there are special provisions for end users to participate in feedback and interaction with virtual entities, security implications, accessibility, timeliness and the ability to model end users as virtual entities makes them important to model within UrbisAPIs.
3. **Hildebrand hosted** – composed of subsystems, this area is the "cloud" for COSMOS and the core of the UrbisAPIs platform; the UrbisAPIs Portal is the user interaction that is detailed in this section
4. **Payment systems** – this is the interface such that payments collected via other systems can be posted as credits into UrbisAPIs.

This document will show the UrbisAPIs Portal functionality and how it is used for the creation, management and operation of virtual entities and applications.

Further to that, roles and user types will be shown with mappings to functionality that the UrbisAPIs system will provide. The creation and management of those roles will be shown.

Finally, this document will show the query and reporting interactions with the devices and device management systems. The detailed internal workings of the sensing, actuating and communications devices will not be covered within this section.

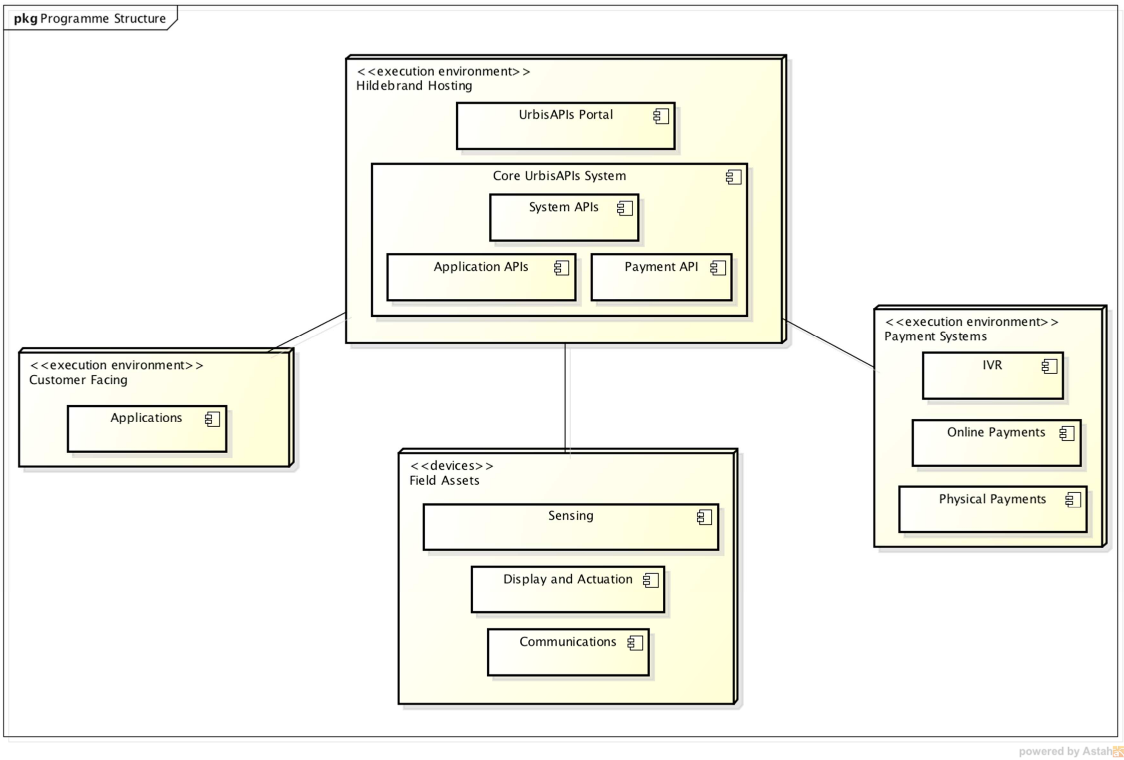

Figure 2 Deployment diagram of UrbisAPIs

2.2 Core UrbisAPIs System

Taking into consideration the very specific area of the Core UrbisAPIs System as show above in Figure 2 Deployment diagram of UrbisAPIs, there are a few high level objectives for any application implemented within UrbisAPIs:

- safe and robust operation of the assets in the field, including the ability to bring new field assets into the system, maintain the working condition of those assets and signal faults that may have wider impact
- representation of a virtual entity within a context of an application
- a system for controlling and managing the feedback to and from the application to support the application with data and events
- robust management of payments and balances where required, including management of exceptions, overrides and credits

The UrbisAPIs Portal provides the management and control via a web based user interface that is hosted by Hildebrand. A concept of a Super User has the capability to configure new users of the UrbisAPIs Portal. The user interfaces described in this document are wireframes to convey the basic functionality presented on a screen. Textual descriptions will compliment and complete the wireframe diagrams and should be read carefully as they may contain more information the diagrams themselves.

A logical architecture is a model that is used to express concepts and label functional process of a system. We are using technique known as the unified modelling language or UML to document components, please refer to external resources for explanations of UML.

This section consists of the following sub-sections:

- Executive Summary - providing a brief summary of the scope of the Heat Metering System and the components involved.
- Assumptions, Issues and Dependencies: describing the various business processes and use-cases which will support the design decisions.
- Logical Architecture and Component Overview - provides a logical view of the various systems required to support the candidate business processes.
- Activity mappings - presents the correlation between business processes and application systems.

2.3 Design Overview

The design approach is user focused with the goal of providing value to the users of the UrbisAPIs system through reuse, sharing and ease of use. When interviewed, possible users of UrbisAPIs were:

- not ICT experts,
- have limited ability to manage software development and software systems implementations
- viewed integration as a major advantage, but had struggled previously in integrating current software systems
- had limited ability to procure a software system that required internal ICT support, but also struggled with security and privacy concerns in outsourcing through software as a service (SaaS)

Our chosen deployment model is SaaS, which means that the high level design considerations can be addressed within the boundaries of SaaS. Security and privacy pose challenges in a cloud environment, especially one where the application layer is outside the control of the city. Where the system design can alleviate concerns with privacy and control choices will be shown as design constraints.

2.3.1. Overview of Users and Roles

From the analysis taken from Section 3 where city operational structures are shown, the following system roles are represented within UrbisAPIs:

Super User – creates and manages all users; has access to all functions

Manager – a role that can be appended to any operational role such that a manager can administer users under the hierarchy, for instance Customer Services Manager can manage all Customer Service users and has the functional rights that the user group has access

Customer Services – view permission on virtual entitles that are within a application entitlement scope

Application Management – view and management of applications that are entitled by a developer, can assign Customer Services entitlements

Finance – view and management of accounting features for virtual entities and API access, can set prices and make financial adjustments to account

Technical Operations – view permissions on system performance indicators like load and number of requests

Application Developer – creates an application and assigns Application Management user entitlement, ability to manage the application and construct new virtual entities, can assign entitlement to technical operations

Data Provider / Maintainer – ability to create new data source and virtual entities and assign permission levels to those sources

3 City Operational Structures

Whereas smart city policy research has developed models for themes, issues and actions in developing a smart city, very little research has been aimed at an operational level. In this section we develop models that can be used for implementation of IoT within a smart city context.

3.1 Background

A thematic approach is the most common way to show the potential benefits of the application of IoT and therefore a smart city approach. This mimics how the city organises budgets and management structure and how citizens would view their various needs from government services.

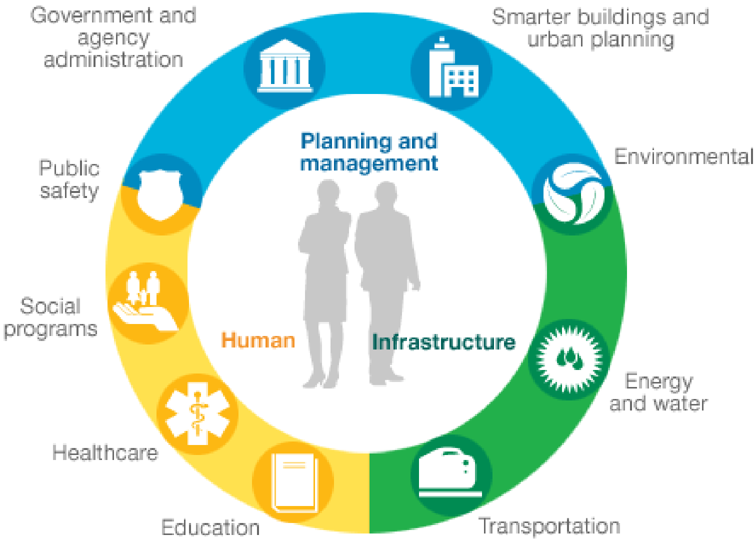

Figure 3 Smart city thematic model, IBM Smarter Cities 2014

Many of the research projects within smart cities Zenn, R2Cities, Pitagoras, Celsius, Insmart, Steep, Eu-Gugle, Step Up, Pleec and Transform for instance are energy system lead with innovation coming from citizens or private business working on the fringe of city operations. The technology deployment model is largely driven by the way a service is procured or how it is financed through risk sharing with systems vendors or community interest groups. This is further a practical consideration as many cities do not own energy infrastructure and merely set policy or policy environments that promote the city's interests.

Other research projects in the smart transport area for cities look at integrating transport for efficiencies, better energy efficiency in public transportation systems, traffic management and emergency services provisions. Here there is more of a direct ownership model of transport assets (roads, buses, track, trains, etc.), large operating budgets and specific technical knowledge required to make these systems function.

The other areas shown in the thematic model exhibit similar affinity towards IoT and ability for the city to directly control influence. Figure 4 shows the relative scale of city government themes ability to implement IoT solutions for smart city applications.

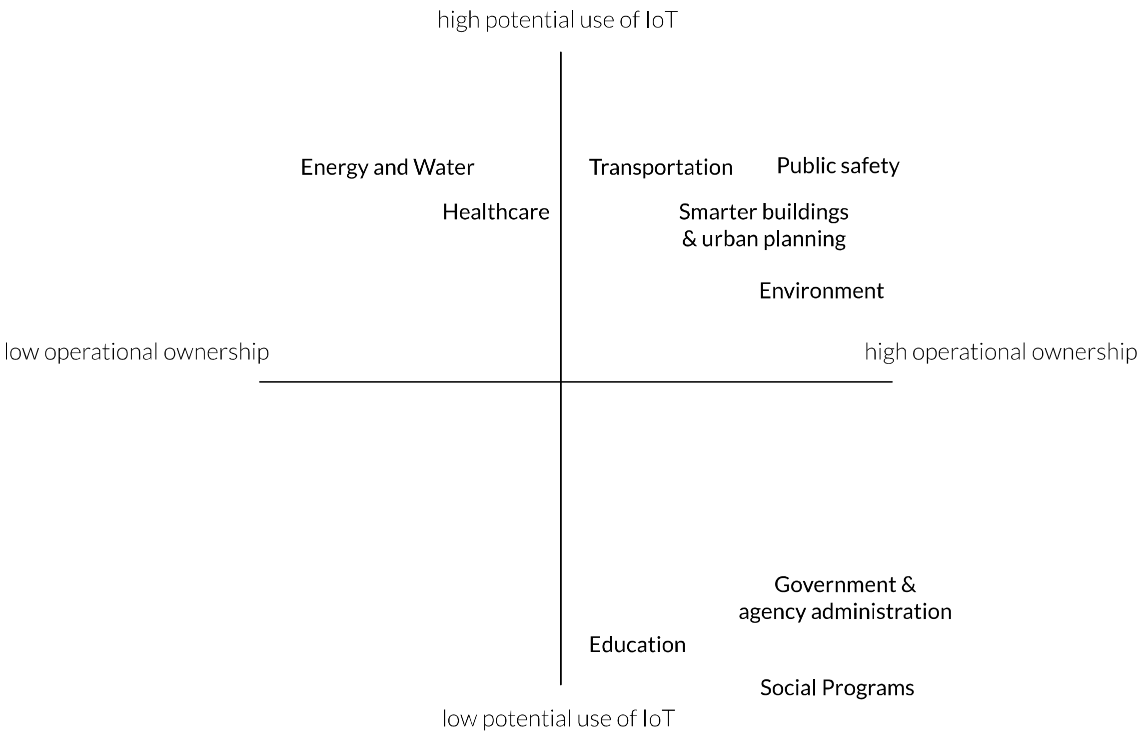

Figure 4 Integration and infrastructure reuse within government organisations are dictated by level of operational influence and use of IoT

3.2 IoT Operational Model for Cities

Our approach has been to show a more operational model of a city like a value chain that would describe the flow of materials and processes to create end user value. This operational model must be generalised enough to be applied to many different city political and administrative systems yet offer some specific guidance on the implementation of IoT within a city.

Figure 5 shows the relationship between the political, administrative, operational and citizen influence. In democratic systems the loop is closed, but sometimes slow moving.

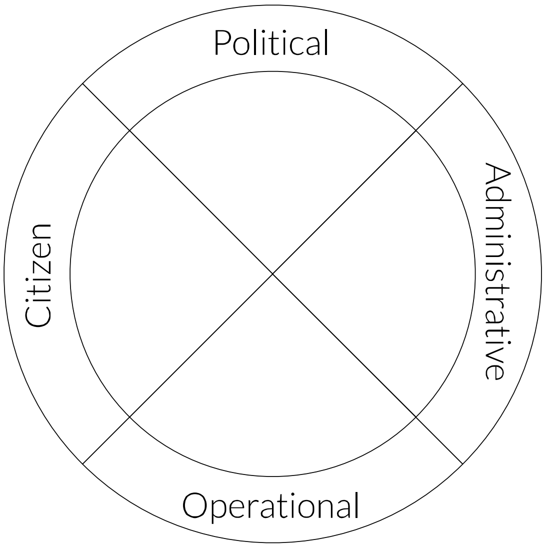

Figure 5 Engagement model for IoT with main city actors effecting service delivery

Taking into consideration the actors in the engagement model the following IoT implications can be described:

Political – the ability to verify policy and budgets with measurable output from IoT systems; the ability to discover the root cause of citizen issues and use data to campaign for new policy; understanding of the inner-dependencies of programmes and assess risk in creating new policy and spending; functions are reporting, simulation and systems reuse

Administrative – taking the policy and programme agenda from elected politicians and turn those into delivery programmes though the use of IoT; verification and budget control as delivery is occurring; ability to procure systems and delivery based on accurate data; new financial models where risk is shared with operator of service (performance contracts) both internally and externally; functions are reporting, cost control, revenue collection and high level management

Operational – service delivery is at an operational level; Figure 6 shows the areas of Inbound Logistics to Service where traditionally a specialist department will create or source a solution that is delivered out to citizens, visitors or businesses within the city; this is the area that will most benefit from reuse and efficiencies the IoT can provide; functions are device management, service creation, application management, user management and integration

Citizen – the citizen has the most to gain from successful use of IoT and like today where citizen's may bring their own smart phone to use city services, they may increase their impact on a smart city by choosing which cars to purchase (traditional, smart, electric), what energy to use (grid, renewable, community generated), lending their data for analysis of services, etc; while citizens may not directly force the choice of IoT technology into a city, the demand for efficiency and integration indirectly puts pressure on politicians and administrators to consider IoT as a possible solution; functions that citizens require are mainly application related, however there could be a future where citizen owned IoT devices are a part of the city's scope for the smart city

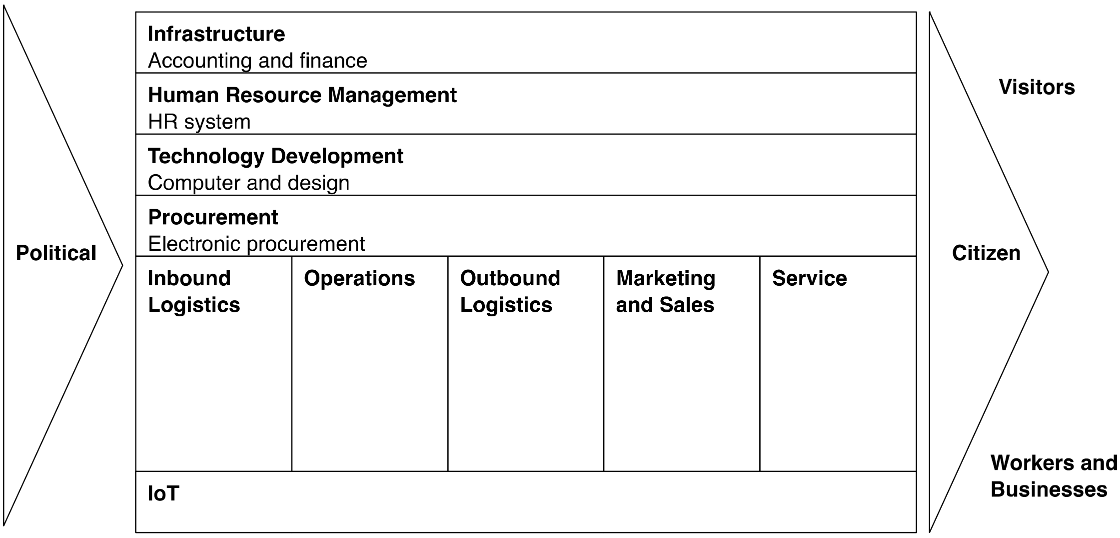

Figure 6 Value chain for smart cities showing where city services operation is situated and where IoT can cut across multiple services to be an integrated, centrally managed component of the smart city

The opportunity that we believe can create the efficiencies, reuse and improved quality is shown as a lateral service offering that can be used by all of the participants within a city value chain.

3.3 City IoT Operational Functions

A set of IoT functional categories are proposed that begin to show where integration between departments logically takes place for deployed sensors, actuators and where a citizen would play a major role in the IoT system's function.

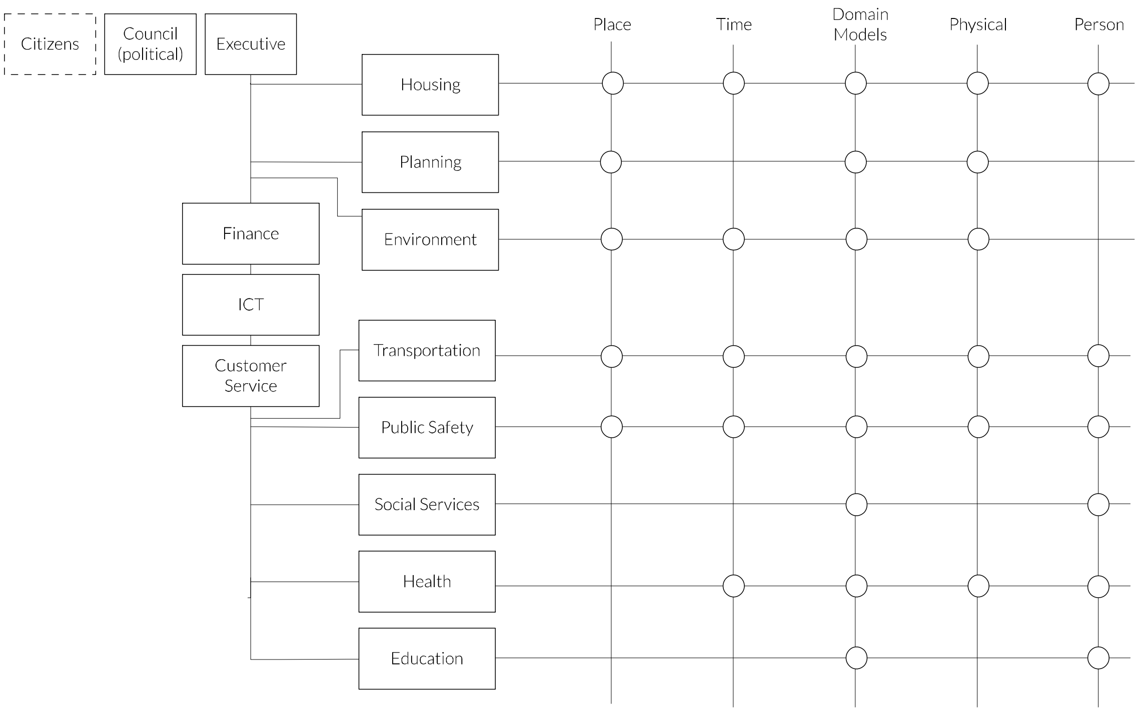

Figure 7 Dimensions of IoT functionality mapped to the operational departments found within a typical city

Place – requiring knowledge of a space that is either build or natural environment; includes management of those places

Time – where timeliness matters within the system; near real-time processes are the ones of main interest in this IoT model

Domain Models – all areas have specific models that can be contributed by a department; example might be a traffic prediction model that would impact environment or planning functions

Physical – the need to measure a physical parameter; in health this might be patient vitals or treatment parameters, for housing it could be an energy sensor; usually will be provided by a sensor and might be controlled by an actuator

Person – in some cases the person is critical component to the system; the person may be acting as a sensor, making choices or influencing other parts of the system directly

4 Application Use Cases

This deliverable is an update to the Year 1 scenarios with six new use cases added. The new use cases are extensions to the physical systems that were described in Year 1.

4.1 Heating Control

Use case : Heating Control
ID: 5
Brief Description: The Energyhive system is measuring the temperature of the properties where it is installed and has the ability to control the delivery of heat through a valve. A new tablet has been deployed within the property that allows for a set point and schedule to be entered. Feedback from users has been that they would like the system to automatically help them set a programme and manage efficiencies on an ongoing basis, for instance detection of whether or not at home; using weather forecast to help with program and supply side management when solar thermal is available for use. The tablet is a COSMOS compatible device and it can act locally to run case based reasoning in an efficient manner.
Primary Actors: Resident
Secondary actors: Mechanical & Electrical Engineer, Sustainability Officer
Preconditions: EnergyHive system must be installed within a resident's premises
Main Flow: 1) Resident will select an autopilot function on their tablet 2) Autopilot will determine a recommended set point for the temperature in the house 3) Set point can be over ridden by the resident 4) The system will learn the patterns of occupation and adjust the run programme to turn off the system based on un-occupied property; the resident can over ride 5) Savings should be quantified over using a normal time based programmer
Postconditions: An improvement in the efficiency in the heating system should be reported

4.2 Building Performance Management

Use case : Building Performance Management
ID: 6
Brief Description: The boiler systems within buildings have master programmers and temperature settings that are controller by a Trend boiler control system. There are also verification instruments installed within buildings to measure the effects of the boiler control, they can provide feedback to inform the run time commands to the boiler control as well. A more granular view of the energy demand, including tradeoffs with electricity usage is desired so that individual residential premises are getting higher comfort while balancing the energy input.
Primary Actors: Mechanical & Electrical Engineer, Sustainability Officer
Secondary actors: Capital Planning Officer, Resident

Preconditions: EnergyHive system must be installed throughout each building in the estate, boiler controls and verification systems must be installed
Main Flow: 1) Temperature readings are collected at a room level within Camden hostel premises 2) Electricity metering will be logged as an input into the energy demand 3) The energy balance model will be run against the Trend readings and the temperature/electricity readings showing performance indicators (degree hour per kWh) against a network model for the delivery 4) Normalisation for seasons and weather conditions should be applied (subtract degree hours inside versus degree hours from weather)
Postconditions: Ranked performance of buildings and properties is reported

4.3 Capital Planning

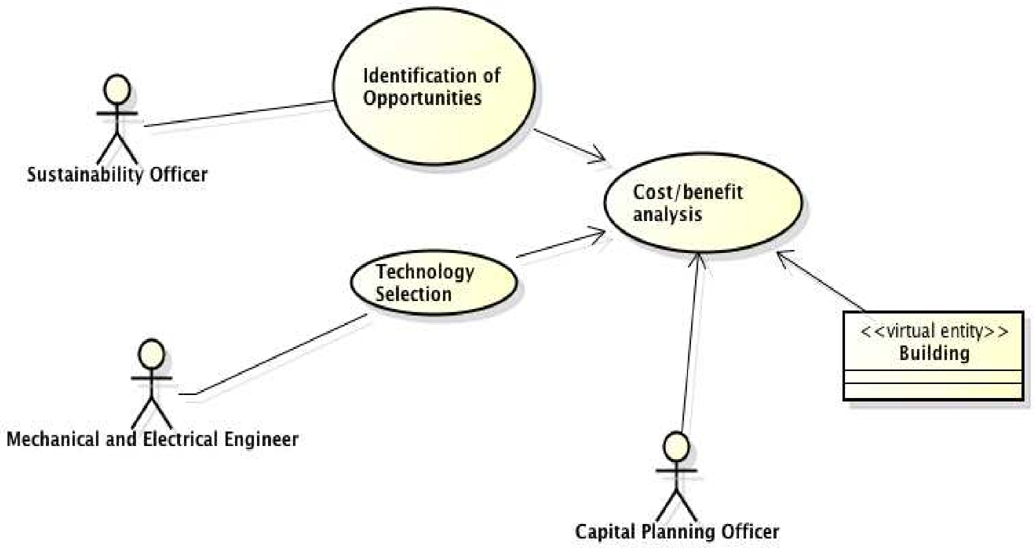

Figure 8 Use case diagram for Capital Planning

Use case : Capital Planning
ID: 1
Brief Description: The EnergyHive system in each building enables Capital Planning officers to perform a more rigorous cost/benefit analysis of suggested programs or technology installations. The system provides accurate information as to the carbon/monetary saving of an implementation.
Primary Actors: Capital Planning Officer
Secondary actors: Mechanical & Electrical Engineer, Sustainability Officer

Preconditions: EnergyHive system must be installed throughout each building in the estate
Main Flow: 5) Sustainability Officer identify an opportunity for environmental improvement of system 6) Engineer select appropriate technology for instalment 7) EnergyHive system provides detailed information as to the effect of the change in the system 8) Capital Planning officer uses EnergyHive information to assist in cost/benefit analysis
Postconditions: The Capital Planning officer decides whether to rollout the proposal

4.4 Identification of Opportunities

Using machine learning, identify where energy savings opportunities exist. This will help sustainability officers suggest projects that can then be put through the Capital Planning use case.

Use case : Identification of Opportunities
ID: 7
Brief Description: The EnergyHive system running in planning mode can use machine learning to suggest opportunities for efficiency. This is largely an unsupervised learning exercise where cause and effect models can be run with comparisons to other like buildings or similar conditions that have been observed
Primary Actors: Sustainability Officer
Secondary actors: NA
Preconditions: EnergyHive system must be installed throughout each building in the estate
Main Flow: 1) Sustainability Officer creates model constraints for parameters to optimise (i.e. cost or carbon savings desired with physical systems 2) Model runs within system bringing up bands of savings that can be made from changes in input parameters 3) System provides control ranges that would have to be implemented in order to make potential savings
Postconditions: A quantified opportunity for efficiency within the energy system is presented for evaluation

4.5 Minimising Carbon

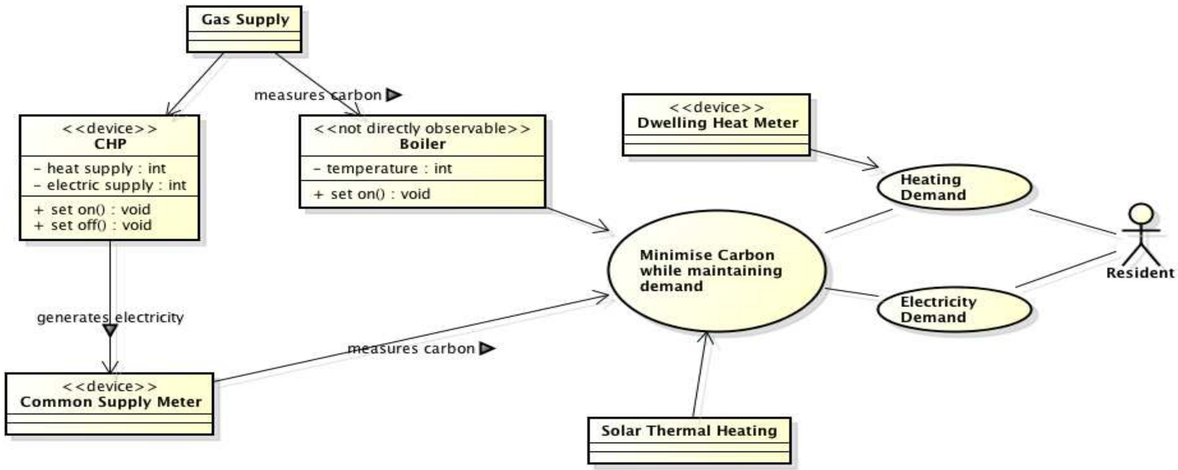

Figure 9 Use case diagram for Minimising Carbon

Use case : Minimising Carbon
ID: 2
Brief Description: An effective way to minimise carbon is to give more weighting to processes with lower carbon production levels, whilst maintaining the demand. The interconnected IoT-based system using an energy platform will make possible effective management of the energy supply in order to minimise carbon production. With minimal input by the resident or site staff, the system will predict the estate's heat and electricity consumption in half hourly intervals and manage the CHP and boiler accordingly.
Primary Actors: Resident
Preconditions: Specialised Instalments 1) Gas Flow meter to CHP from boiler to regulate the Gas supply 2) Control system with temperature sensor on boiler 3) Flow meter/temperature sensor on Solar Thermal 4) Heat meter in each dwelling 5) Communication infrastructure between sensors and hub
Main Flow: 1) System predicts the estate's heat and electricity demand for a half hour period 2) System calculates required gas supply and distributes to CHP and boiler accordingly 3) Carbon produced is measured 4) Individual resident heat consumption is monitored
Postconditions: 1) The resident is charged for their personal heat consumption 2) Prediction errors are logged to improve system on later iterations

4.6 Minimising Demand

Use case : Minimising Demand
ID: 3
Brief Description: Another method of reducing carbon production is to minimise the demand for Heat energy production. This is possible through the current IoT platform, namely EnergyHive (designed by Hildebrand). The EnergyHive system will use smart meters to report real-time energy consumption information automatically and remotely. The system assists the user in setting a heating schedule with accordance to their budget.
Primary Actors: Resident
Preconditions: 1) EnergyHive system implemented in each dwelling 2) Valve up/ down control system to the radiator
Main Flow: 1) Resident accesses their customer account to view balance 2) Resident can set a heating schedule 3) Resident is given tariff and projected balance for a given schedule
Postconditions: 1) User can optimise their schedule to minimise their consumption

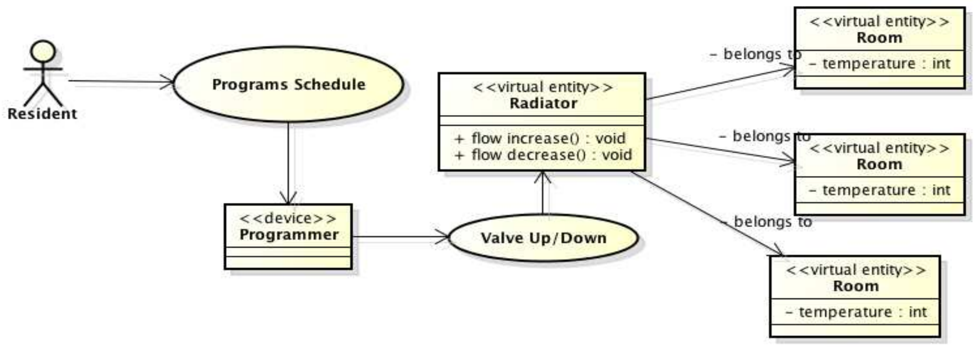

Figure 10 Use case diagram for Minimising Demand

4.7 Special Needs Passenger with Carer Assistance

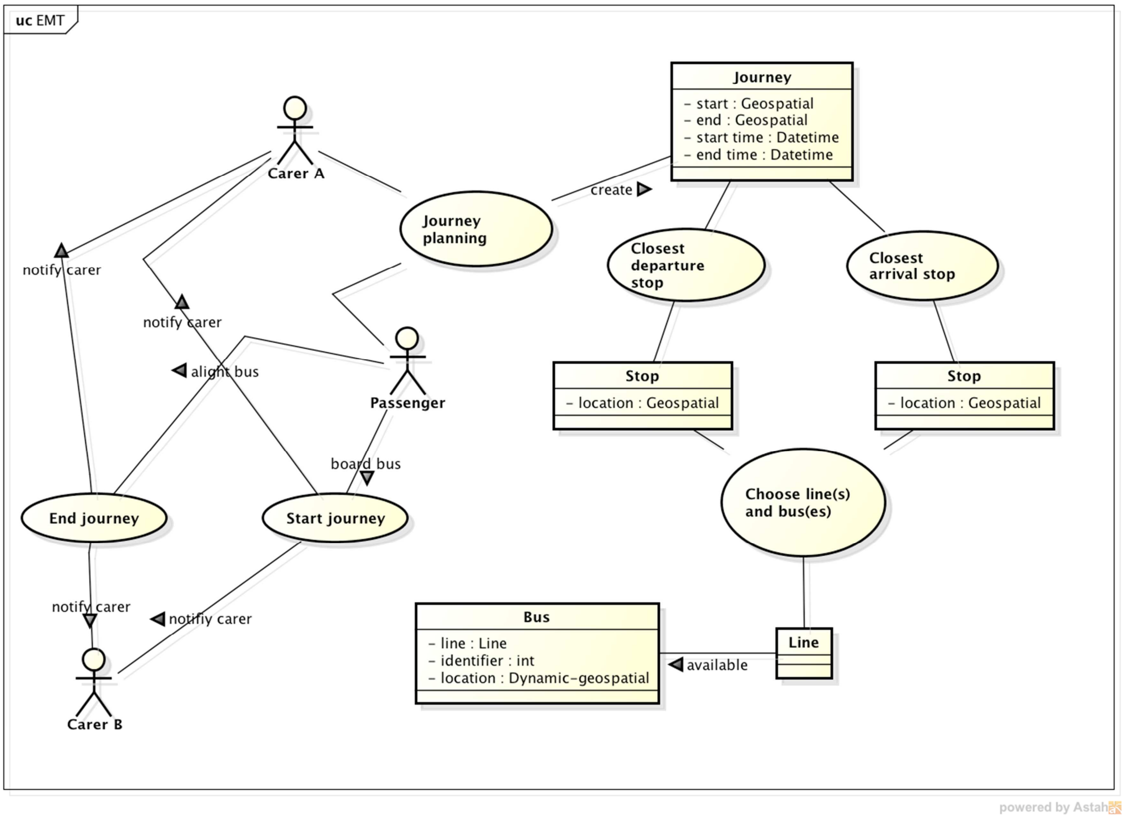

Figure 11 Special needs passenger traveling on the bus system with assistance from two Carers

Use case : Special needs passenger journey

ID: 4

Brief Description: Passengers that have special needs such as children, elderly, disabled and the like, may choose to use the bus system if they can get assistance on the beginning and end of their journey. Assistance would come in the form of a carer who might help the passenger plan their journey, track the passenger's progress and then hand off to a new carer that would be waiting at the destination.

Primary Actors: Passenger, Carer A and Carer B

Preconditions: Journey requirement

1) Passenger, Carer A and Carer B know of each other
2) Carer A is with the passenger at the beginning of the journey

Main Flow:

5) Carer A and the Passenger configures a new journey
6) System finds the closest departure stop
7) System finds the closest arrival stop
8) Lines are selected such that the Passenger boards a bus that will travel from departure to arrival
9) Carer A and B are notified that the Passenger boarded a specific bus
10) Carer A and B are updated with progress of the bus including estimated time of arrival
11) Carer A and B are updated when the Passenger has finished the journey

Postconditions:

12) The Passenger has arrived at the destination
13) Carer A knows that the Passenger arrived safely

14) Carer B knows if the Passenger has not arrived when expected

5 IoTA Model

5.1 IoTA Model for the Energy Scenarios

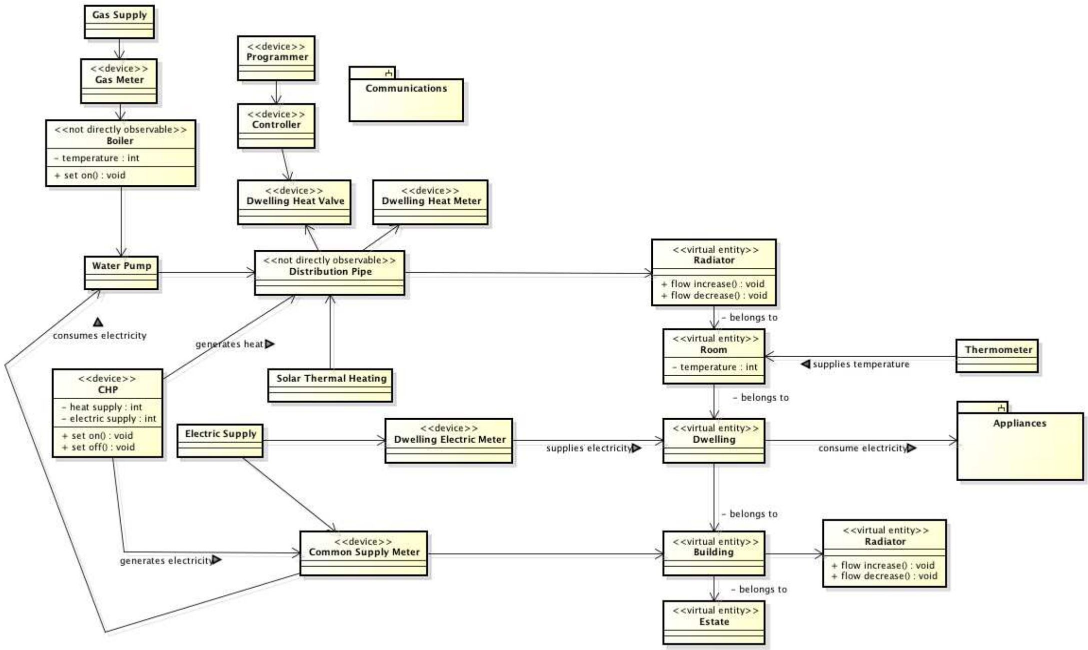

Figure 12 Model for the IoT system with adjoining description of components for the energy scenarios

Component	Description
Gas Supply	Natural Gas supplied to the Ampthill Estate. Consumed by the boiler and the CHP
Electric Supply (National Grid)	Electricity supplied to the Ampthill Estate from the National Grid. Electricity consumption from individual residents is taken from this supply. The National Grid also contributes to the common supply for building and estate utilities.
Boiler	Natural Gas fuelled boiler. Contributes hot water on resident, building and estate level. The boiler is not directly observable, therefore it is monitored and controlled by the Gas meter
Solar Thermal Heating	Multiple solar thermal panels installed on the roof of each building. Contributes hot water on resident, building and estate level.
Combined Heat & Power (CHP) Engine	Natural Gas fired Gas turbine engine. Contributes hot water on resident, building and estate level. Also contributes electricity to the common supply for building and estate utilities.
Water Pump / Distribution Pipe	A system of water pumps transport hot water from each source through the distribution pipe. Consumes electricity from the National Grid and the CHP. The distribution pipe supplies water to each individual resident, along with utilities in the building and on the estate.
Gas Meter	Monitors the flow and temperature of gas through the boiler to indirectly regulate and control boiler. Also monitors flow to the CHP, to indirectly measure electricity and heat energy from the CHP.
Common Supply Meter	Measures the total electricity supply on a building and estate level (excluding individual resident supply).
Dwelling Electric Meter	Measures electricity to individual resident dwellings. Electricity is only supplied by the National Grid.

Dwelling Heat Meter & Valve	Measures heat energy consumption for each resident dwelling. The valve is controlled by the programmed schedule
Programmer/Controller	Receives/sets schedules for heating and operates the hot water valve accordingly

5.2 IoTA Model for Bus Scenarios

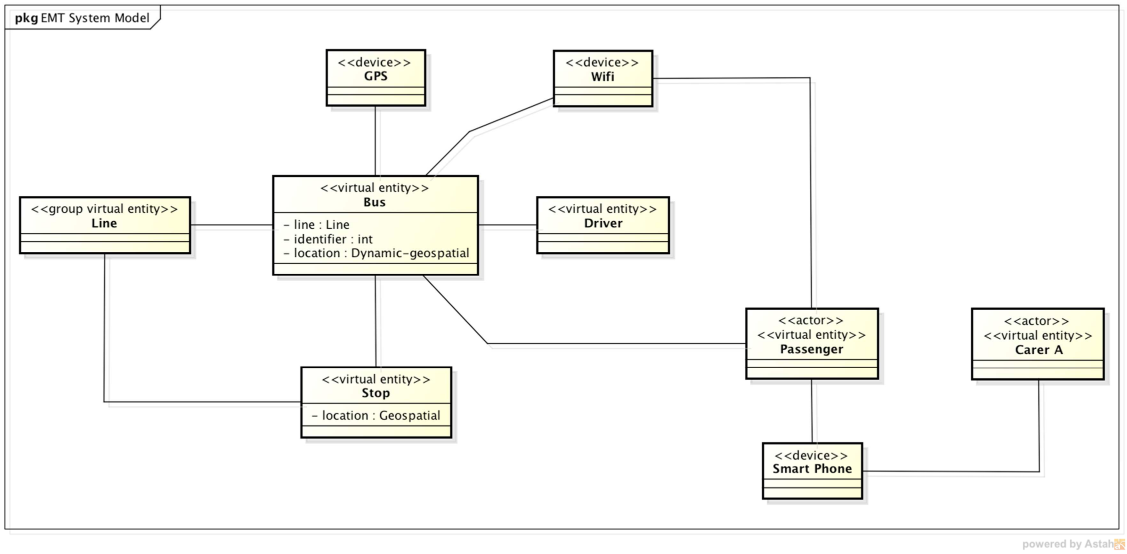

Figure 13 Model for the IoT system within the EMT transport scenarios

Component	Description
Line	A line is a group virtual entity that is composed of the buses and stops that operate on a given line. In real terms the line is uniquely identified to bus users with a line name and color.
Bus	The bus is the vehicle that operates on a line. It is flexible in that one day a particular bus may be operating on one line and another day it has been redeployed. The identifier that goes along with the bus will uniquely identify the bus for its lifetime. The bus is a virtual entity as it describes the moving space tracked by the devices on board.
GPS	GPS is a device that tracks the location of a bus and maintains the bus virtual entity position sensing over time. It is a sensor type device.
Wifi	The Wifi is a device that is a resource to passengers for gaining access to the Internet. It also acts as a sensor in that if someone is within radio contact to the Wifi then they are within proximity of the bus. There is an implication that if the duration of radio contact is long enough, then the smart phone giving off a Wifi signal is onboard the bus. Otherwise if the duration is small, the bus may have driven past the smart phone. Note, the smart phone is also a proxy for a person which is carrying the phone.
Driver	The driver is in control of the bus and can indicate various statuses of the bus. The driver takes on the position of the bus and could be used as an extra "sensor" in future scenarios.
Stop	A stop is a place that has a position in geospatial terms. It will have a unique identifier and be a part of one or more lines. Stops that are a part of multiple lines are of special interest as passengers may change at these locations to get on to different parts of the bus network.
Passenger	A passenger is a virtual entity that will assume characteristics of the bus once they are on board. It is also assumed that a passenger will have a mobile phone that will act as a sensor as well. Typically a passenger is a primary actor in a bus scenario and may need various foreign identifiers to be useful for an applications.
Smart Phone	A smart phone is available to human actors in the system. The smart phone can be considered a sensor as well as a actuator within an IoT context. The actuator functionality is usually informational.

Carer	A Carer is a special actor that is observing an IoT process. They can contribute information to the system via a smart phone or web application.

6 Data Sources and Structures

6.1 Heating System Data Feeds

The complete set of data feeds required for the IoT system is described in Figure 14. The data feeds that are necessary for prediction will need to be logged over a trial period, to provide training data for the system.

Data Feed	Source	Purpose
Luminosity	EnergyHive multi-sensor	Thermal Load Prediction. Solar Thermal Supply Prediction
Weather (Historic/Real-time /Forecast)	MetOffice	Thermal Load Prediction Electricity Demand Prediction
Temperature (indoor/outdoor)	EnergyHive multi-sensor	Thermal Load Prediction
Electricity Consumption data	Common/Dwelling Electric Meter (Hildebrand)	Total Electricity Supply Calculation Electricity Demand Prediction Post-Prediction Error Calculation
Thermal Load data	Common/Dwelling Heat Meters (Hildebrand)	Total Thermal Energy Supply Calculation Thermal Load Prediction Post-Prediction Error calculation
Solar Thermal Energy supply	Solar Heat Meter (Hildebrand)	Total Thermal Energy Supply Calculation Post-Prediction Error calculation
Gas flow/temperature	Heat Meter	Boiler/CHP Control System

Figure 14 Data feeds described by their source and purpose in the system

6.2 Heating System Data Flow Model

An example model of the IoT system, built in SciLab, is shown in figure 10.

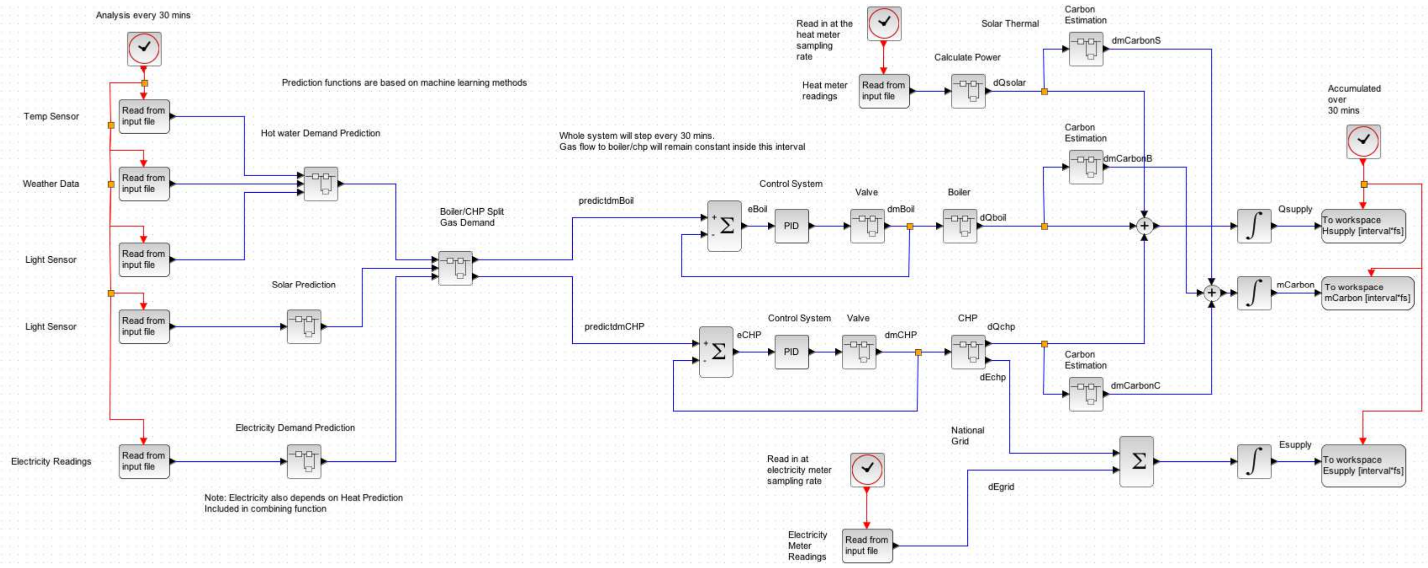

Figure 10: SciLab model of the system

The diagram provides an overview of the processing of data from the input feeds. These input feeds are processed using a statistical prediction model, which outputs the expected thermal load, electricity usage and solar thermal input for the next half hour period. Another calculation step is necessary to determine the optimum gas supply allocation to the CHP and boiler for the predicted conditions.

The SciLab model represents the gas supply to the Boiler and CHP through a control system with a PID controller, which controls a valve to reach the desired mass flow rate of gas. The expected output from the Boiler and CHP is calculated using a set of thermodynamic equations. The final electricity and thermal energy supply is stored in (kWh), along with an estimation for the total mass of carbon produced in the half hour period.

6.3 EMT Bus System Data

Data from the buses is available in a variety of formats via the EMT Open Data portal (http://opendata.emtmadrid.es/) , but specifically we will use the Google Transit formats for cycle 1 as they serve as a defacto standard to integrate other cities within the COSMOS framework. The following table describes those sources and how they will be used within the project.

The data sources can be found at: http://servicios.emtmadrid.es:8080/GTFS/transitEMT.zip and a more detailed description at https://developers.google.com/transit/gtfs/ Within the table, the source column describes the file within the zip archive that will supply the data.

Data Feed	Source	Purpose
Lines	routes.txt	Lines and identifiers that are used for keys into other data sources

		Links to status URLs that show the line
Weather	Norwegian Meteorological Institute (http://www.yr.no/)	Predicting disruption Predicting demand
Stops	stops.txt	Position of the stops within the bus network, including a unique key and lat/lng coordinates
Incidents	http://servicios.emtmadrid.es:8080/rss/emtrss.xml	RSS feed of disruptions, both planned and unplanned
Stop Times	stop_times.txt	The planned schedule for buses to arrive and depart at particular stops Initialising the model and tracking performance against this schedule
GetEstimatesIncident	https://openbus.emtmadrid.es:9443/emt-proxy-server/last/media/GetEstimatesIncident.php	Latest information for a stop/line combination. Includes GPS data of the buses and estimated time of arrival to the stop.

Real time return from the EMT GetEstimatesIncident API:


```
{
  "errorCode": "000",
  "description": "La recuperación de ServiceMedia ha sido correcta.",
  "stop": {
    "label": "1",
    "description": "AV.VALDEMARIN-ALTAIR",
    "direction": "Av. de Valdemarin, 88",
    "stopLines": {
      "data": {
        "label": "161",
        "description": "MONCLOA - ESTACION ARAVACA"
      }
    }
  },
  "arrives": {
    "arriveEstimationList": {
      "arrive": [
        {
          "stopId": 1,
          "lineId": "161",
          "isHead": "False",
          "destination": "ESTACION ARAVACA",
          "busId": "4650",
          "busTimeLeft": 711,
          "busDistance": 7129,
```

```
          "longitude": -3.7398198974758,
          "latitude": 40.43554589658,
          "busPositionType": 1
        },
        {
          "stopId": 1,
          "lineId": "161",
          "isHead": "False",
          "destination": "ESTACION ARAVACA",
          "busId": "4638",
          "busTimeLeft": 999999,
          "busDistance": 9568,
          "longitude": -3.730715139771,
          "latitude": 40.434996699893,
          "busPositionType": 1
        }
      ]
    }
  },
  "incident": {
    "lastBuildDate": "24 Nov 2014 16:16:05 GMT"
  }
}
```


6.4 Bus System Data Flow Model

The data flow model for the bus system has two main models that need to execute within COSMOS for IoT data, firstly presence of a passenger and secondly prediction/estimation of arrival of the bus.

For the presence detection, a MAC address will be used as the method to identify a passenger via Wifi sensing. Only MAC address that are registered will be considered.

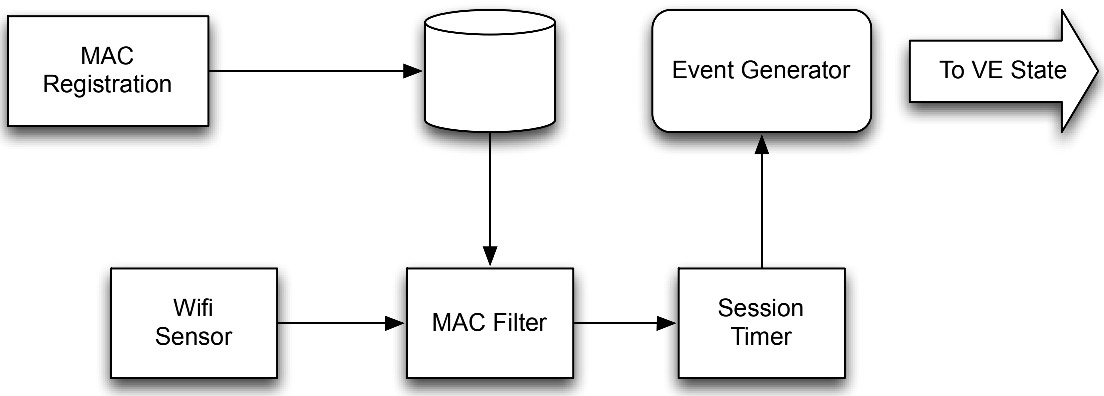

Figure 15 Presence detection using a MAC address registered to a virtual entity

The update to predict a bus arrival will be done on a cascading estimator that will be recomputed at every cycle. A cache will be maintained to reduce load on the system.

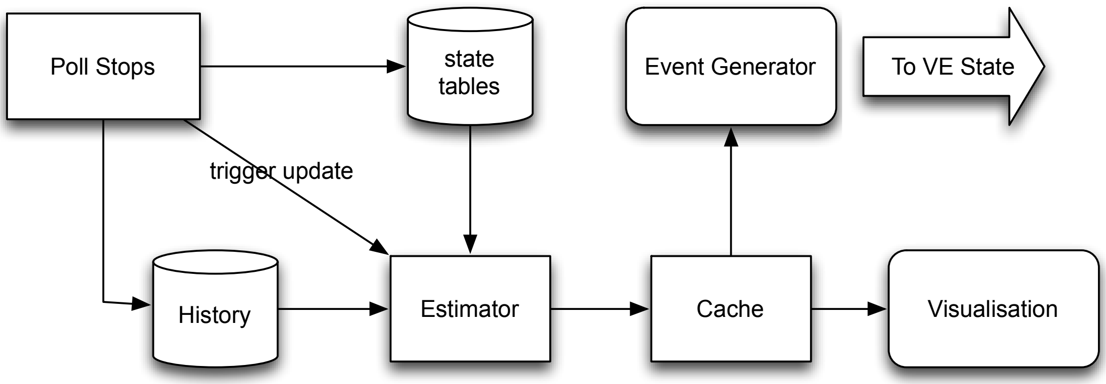

Figure 16 Estimation of position and arrival time of buses, will also map to virtual entities that are on board a bus

The estimator will look at the previous journey times on the segments of interest and apply those to near future predictions. In the case of predictions further out, history of journeys based on average travel time and variance will be used alongside time of day (within a 30 min window) as a classifier. A naïve Bayes classifier will use that history as trained input and it will be run for day of the week, holiday days and 30 minute time window on each route.

6.5 Taipei Smart Home Management System Data

In-Snergy API uses the HTTP protocol to transfer data, including information queries, device control and other functions. In-Snergy receives HTTP Request command, and will return HTTP Response. For the Content-Type, HTTP Response supports JSON and XML format. If return format not specified, the default format returned will be JSON data. All API URL begins with http://api.insnergy.com/api/.

For example, the query API for the basic user data is user / info.

The complete URL is http://api.insnergy.com/api/user/info.

6.5.1. Request Format

Request format uses the following format:

<api_method>[.format][?format=format_value][&param_name=param_value]

.format can be .json or.xml, format_value can be json or xml, returned in JSON format and XML format.Ex:

* user/info

* user/info?user_id=myuserid

* user/info.json?user_id=myuserid

* user/info.xml?user_id=myuserid

* user/info?user_id=myuserid&format=json

* user/info?user_id=myuserid&format=xml

6.5.1 API directory

The following lists the detailed usage of API.

Function	Method name	Authentication	Parameters
auth/login	Authenticate the user account password	Permissions verification must be done	* auth (required) Verification code: connect user_id and user_pass colon with ":", and then perform Base64 to encode string together.
device/control	The specified device control	Permissions verification must be done	dev_type (required) Device Type * 02 = Socket-110V * 06 = Multicircuit Socket * 07 = Socket-220V *dev_id (required) Device ID * action (required) Device Action * on = Device ON * off = Device OFF
appliance/user	Query according to the user account for appliances account information	Permissions verification must be done	cust_id (required) User Account

* Example requests

http://api.insnergy.com/api/appliance/cust_id?myuserid

6.6 Taipei Smart home management System Data Flow Model

The following diagram describes the Taipei Smart home management System's data flow used for Minsheng Community household trial. The flow is, the sensors uploads data to the management system, and through the server, the information will distribute data to Rule engine and database, Rule engine performs database auto services including firing alarm and auto control of the on/off switches, and VEE will base on database setting performing the data checking and adjustments.

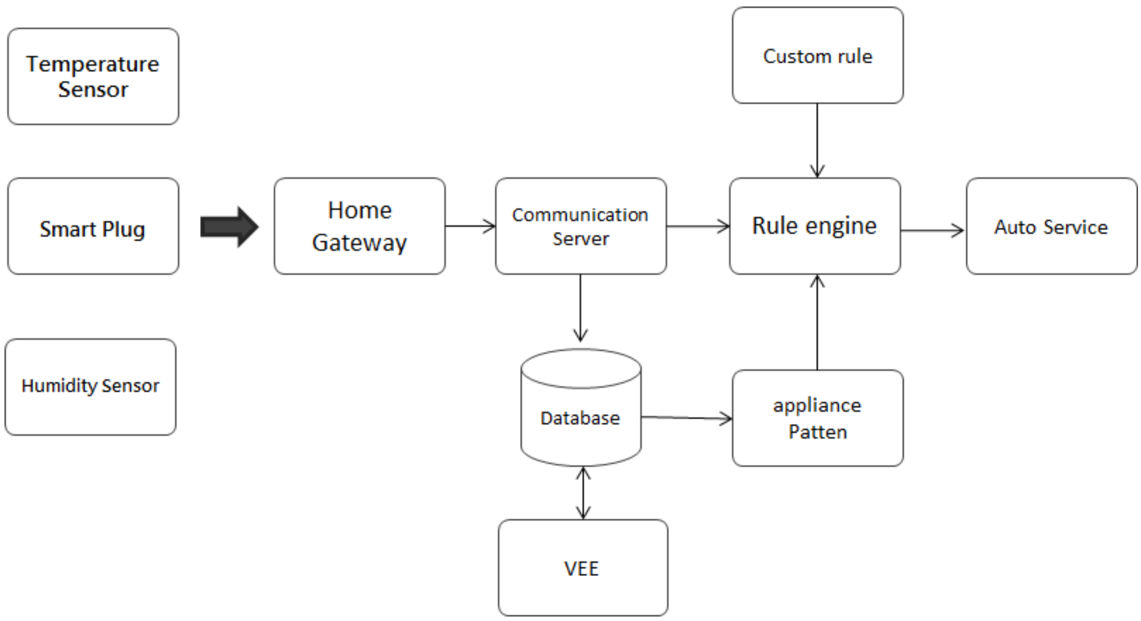

Figure 17 In-Synergy data flow

7 Scenario Implementations

Each of the Use Cases above will be implemented through the realisation of an application. The application is where specific end user requirements are addressed, rather than in the core COSMOS system. Therefore the application approach for the various Use Cases will be shown in this section.

The application developer requirements and requirements for the city to support COSMOS has been the main focus of Year 1 and will span into Year 2 with more depth and breadth.

7.1 Camden Heat Metering Application

7.1.1. Integration with COSMOS

The heat metering application implements the Camden Heat Metering System with integration to COSMOS technologies to supply intelligence to the system and end user.

This section provides a high level design of the Heat Meter Application which forms the core of functionality of an end to end system to measure, bill, collect revenue and manage Camden Heat Meters. Because the overall Camden Heat Meter Programme has many aspects, the delivery is broken down into four packages that are structured in a way that simplifies the communication, functionality and management.

1. Physical Heat Meters and Communications equipment – incorporating all of the physical assets and software found within the residential premises.
2. Heat Metering System – the system that manages the metering and communication assets, manages data from the meters, enables the reporting of the consumption information back to end users and back office staff. Used principally by Hildebrand for the operation of the Heat Metering System, all functionality that is exposed to COSMOS via virtual entity APIs.
3. Back Office System – the system that reports data back to back office staff through web views and exports, allows for different roles within Camden to perform their tasks (such as Finance, M&E, IT, etc.)
4. Payment Interface – this is the interface such that payments collected via other systems can be posted as credits into the Heat Metering System.
5. Resident Interfaces – this is the user interface for the resident to manage their interactions with the Heat Metering System, including a tablet and web site used for support.

This section will show how the Heat Meter System is intended to be used at a process level from the Back Office Portal for management/operation of the:

- physical assets – heat supply, metering, communications and feedback technology
- energy performance management – building and household energy consumption and efficiency through reporting and consumption data export
- accounting and financial elements – prepaid billing system for heat, payment into a prepayment account, management of prepayment accounts, rate setting and tariff management

Also this section will show how Payments will be received from the Capita payments system and be credited to prepayment accounts held inside the Heat Metering System. This is done through an application-programming interface (API) that allows the Capita Payments System to securely communicate with the Heat Metering System. The specification of the Payment interfaces are in that project's detailed documentation, however this document acts as the high level design document to show system level interactions.

This section will show the interfaces that will be exposed to the Camden Customer Access Programme such that dashboards can be presented to Camden residents. These are system level API interfaces and interactions, not graphical user interfaces. Those GUI specifications will be covered in a yet to be decided project in cooperation with the Camden Customer Access Programme.

In addition to the Camden Customer Access integration, the tablet interface will be shown. The requirements have been specified Camden Council on behalf of the residential end users as there are specific regulatory and operational process requirements that need to be met in the first instance of this application.

Finally, this section will show the system level interactions with the Physical Metering and Communications devices and systems. The detailed internal workings of the meters and communications devices will not be covered within this document.

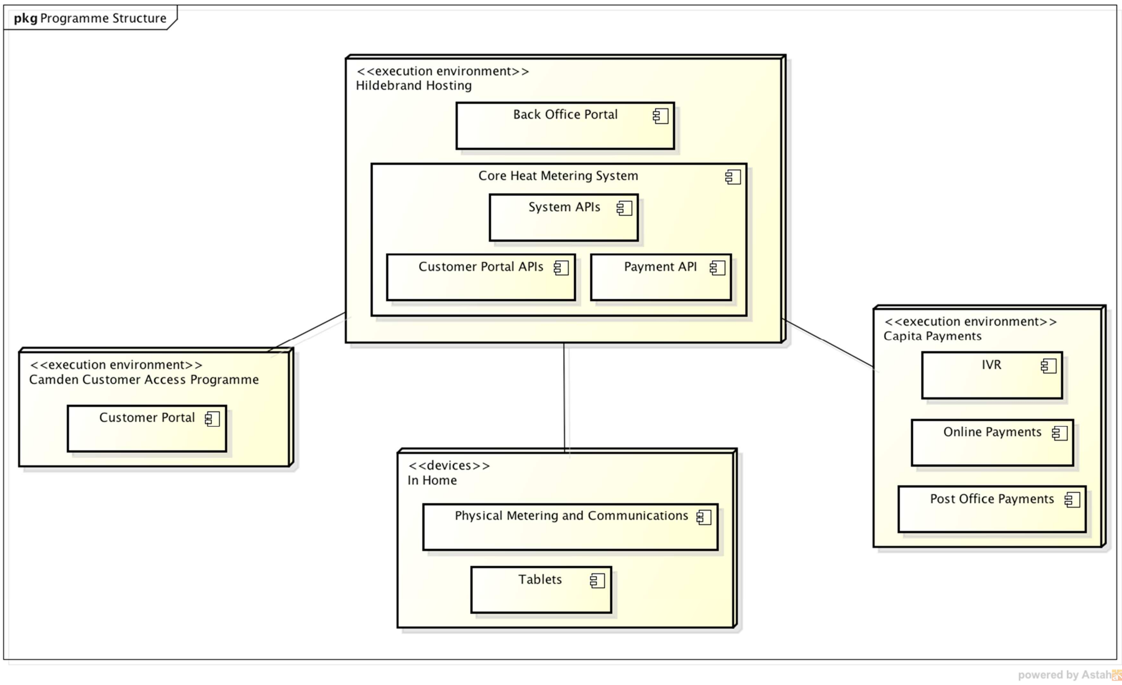

Figure 18 Component and deployment diagram showing all of the physical locations, scope of responsibility and systems associated with the Camden Heat Metering Programme. Note the Core Heat Metering System is at the centre of the Hildebrand responsibility.

7.1.2. Heat Metering System Core

Taking into consideration the very specific area of the Core Heat Metering System as show above in Figure 1, it has a few high level objectives:

- safe and robust operation of the assets in the field, including the ability to bring new properties onto the scheme, maintain the working condition of those assets and signal faults that may have wider impact

- measurement of energy (specifically heat in this case) consumption per household with the generation of a real-time bill and account balances that can be used in a “pay as you go” scheme
- a system for controlling and managing the feedback to householders and energy consumers to include them in consumption, economic, environmental and financial planning decisions
- robust operation of payments and balances, including management of exceptions, overrides and credits

COSMOS interfaces with the Core Heat Metering System and works with the local tablets to enact offline services and social sharing.

Requirements

7.1.3. In-home Components

The in-home components are largely managed by the Tablet that is found in the home. The Tablet is set up to work with a meter and in turn the meter is assigned to a property. Therefore the assignment of the Tablet to a Property is through this chain of associations.

A Customer Account that holds prepaid balance is assigned to a Property. This allows for meter assets to be replaced without disrupting the account balance for a Customer Account. Transactions (credit/debit/adjustments) contain a log entry of the Property that they were made against. Transactions are created with the information that is the current association of Customer Account to Property. If that assignment is changed, history will not be changed, but any future transactions will have the new association of Customer Account to Property.

Section details the design of the In-home components and the tablet, including the User Interface elements where appropriate. The following table is the high level description of the interactions that are required.

ID	Name	Description
SI-1	Alert balance	A visual alert on the tablet which is triggered by the CHMS indicating there is problem with the account balance linked to the meter that the tablet is paired with.
SI-2	View message	Messages can be sent to a tablet with the view message function being able to display a message to the resident. Messages originate from the CHMS or can be put into the CHMS by Backoffice Portal functionality.
SI-3	View tariff	The tariff that is assigned to the Customer Account
SI-4	View weather	Current and forecast weather for the nearest relevant location that the Property that the meter is assigned to.
SI-5	View projected balance	Based on weather forecast and historical usage there will be several views into a projected balance. The first is in terms of projecting a projected “Day to Top Up” and the second as a forecast of spend today, this week and this month.
SI-6	View comparisons	Using usage across similar households, display how your household compares in terms of energy use

SI-7	View instant values	Display of the power (in kilowatts) being used at current time. This is a dynamic reading based on how hot water and heating are being used and will have some lag while the system transmits that reading, however it will be considered "real-time"
SI-8	View historical consumption	Display of energy (in kilowatt hours) that has been used over previous days, weeks, months as a summary and as navigable timeseries graphs
SI-9	View account	Ability to display the Customer Account number on screen with near real-time Balance and the last 5 credits and last 5 debits, including any adjustments that have been made, also the status of the account and the meter/property information that the tablet is configured against.
SI-10	Pair tablet with meter	In order for the tablet to be able to display the above information, it needs to be able to be paired with a meter. The meter, property and customer account associations will then link back through a chain of associations to the tablet. This process will enable the association of the tablet with the meter.

7.1.4. Customer Access Portal System Interactions

The Customer Access Portal is provided by Camden with a link to the information stored within the Core Heat Metering System. The link must be set up by securely associating the CAP identifier with the Property identifier in the CHMS.

If the resident moves or ceases tenancy, the CAP must direct the unlinking and linking processes provided by the CHMS. The CAP may need to validate tenancy agreements, address confirmation, proof of residency to make those links.

The following requirements have been identified for the initial customer access portal:

Unique ID	Requirement
CHM.CAP.101.	CAP must have the ability for a user to change their password
CHM.CAP.102.	CAP will assign or let the user choose a user name
CHM.CAP.103.	User must have the ability to securely associate their CAP account with their CHM Customer Account
CHM.CAP.104.	User must be able to view the status of their account
CHM.CAP.105.	User should be able to turn off their heat and ensure that no money will be taken from their Customer Account (suspend account) if they are gone for a long period
CHM.CAP.106.	User should be able to see their instant power usage in Watts
CHM.CAP.107.	User should be able to see their rate of use in monetary terms, £ per hour usage calculated over a rolling 15 minute window
CHM.CAP.108.	User should be able to see the total amount of energy in kWh used by periods
CHM.CAP.109.	Periods will be defined as day (last 24 hours), week (last 7 days), month (last 30 days) and last year (last 365 days)
CHM.CAP.110.	Historical usage should be shown graphically as a bar chart showing the daily usage for a selectable periods

CHM.CAP.111.	User should be able to see the total amount of usage (cummulative in kWh) from 1 April 2013
CHM.CAP.112.	User should be able to view comparisons with flats of the same size (number of bedrooms) within the Block that it is located in for Periods as defined. This is shown in kWh
CHM.CAP.113.	User should be able to view energy use for the previous year for the same Periods as defined. For example, same week last year or same month last year.
CHM.CAP.114.	User should be able to see the account number and barcode of their Customer Account. There is no difference in Prepaid to Credit customers in this case
CHM.CAP.115.	User should be able to see their Balance that is on the Customer Account
CHM.CAP.116.	User should be able to see the cost of energy for the same Periods as defined
CHM.CAP.117.	The User should be shown an estimated top up date based on a forecast of energy usage
CHM.CAP.118.	User should be able to see their tariff with unit costs displayed
CHM.CAP.119.	User should get a strong indicator on the User Interface that they are running low on credit
CHM.CAP.120.	User should get a very strong indicator when they credit has run out and they are to be or have been disconnected from supply
CHM.CAP.121.	User should have access to the transaction log of financial transactions against their account, this will include a rolling balance on a daily basis, i.e. at least one transaction per day showing the consumption and balance at the end of that day
CHM.CAP.122.	User should have access to the history log of state changes against their account
CHM.CAP.123.	Messages and message history must be able to be delivered to the user, they will be limited to 100 characters
CHM.CAP.124.	Communication outages should be handled
CHM.CAP.125.	Estimated performance and loads should be handled
CHM.CAP.126.	The solution should be able to look up all information based on the Customer Account number
CHM.CAP.127.	The solution will use secure HTTPS as the application data protocol with a Restful design
CHM.CAP.128.	Data encoding will be via JSON notation
CHM.CAP.129.	The solution will only accept calls from known IP addresses
CHM.CAP.130.	The solution will use HMAC tokens to secure the integrity of API requests
CHM.CAP.131.	The solution should achieve an up time of better than 99.8% excluding scheduled outages
CHM.CAP.132.	User should have the ability to change their billing preference from Paypaid to Credit if they are a vulerable tenant

Following the requirements that have been captured, a set of system interactions have been designed to meet those requirements.

ID	Name	Description

SI-11	Alert balance	A visual alert on the tablet that is triggered by the CHMS indicating there is problem with the account balance linked to the meter that the tablet is paired with.
SI-12	View messages	A list of all of the messages that have been sent to the user account.
SI-13	View tariff	The tariff that is assigned to the Customer Account showing the structure and values assigned.
SI-14	View weather	Current and forecast weather for the nearest relevant location that the Property that the meter is assigned to.
SI-15	View projected balance	Based on weather forecast and historical usage there will be several views into a projected balance. The first is in terms of projecting a projected “Day to Top Up” and the second as a forecast of spend today, this week and this month.
SI-16	View comparisons	Using usage across similar households, display how your household compares in terms of energy use
SI-17	View instant values	Display of the power (in kilowatts) being used at current time. This is a dynamic reading based on how hot water and heating are being used and will have some lag while the system transmits that reading, however it will be considered “real-time”
SI-18	View historical consumption	Display of energy (in kilowatt hours) that has been used over previous days, weeks, months as a summary and as navigable timeseries graphs
SI-19	View account	Ability to display the Customer Account number on screen with near real-time Balance and all transactions including credits, debits and any adjustments that have been made, also the status of the account and the meter/property information that the CAP account is configured against.
SI-20	Create Customer Account	The ability for a CAP user to create a heat metering Customer Account, this will then provide the CAP user with the ability to make payments against that Account, view the data from the linked meter, etc.

7.1.5. Backoffice Portal

The Backoffice Portal has systems interactions that are used by staff to manage various aspects of the overall system. This includes physical assets, residential end users and financial accounts belonging to residents.

The following requirements for the Backoffice were identified by Camden staff:

Unique ID	Requirement
CHM.BO.101.	Account information must be able to be found through a search by block and flat information, showing the PRN that the account is currently assigned to
CHM.BO.102.	The Customer Account can be of either type prepayment or payment on account (extending credit)

CHM.BO.103.	Within the solution when showing details of the Property and Account, the Customer Account number should be shown
CHM.BO.104.	Within the solution when showing details of the Account, the full bar code number that also contains the Customer Account number should be shown
CHM.BO.105.	The solution should show the account status of which there is Active/Ceased/Suspended - there are other inferred states on the Account as well, Disconnected and Unlinked/Linked
CHM.BO.106.	The system should record the time and date that the Customer Account was created and by what method
CHM.BO.107.	The system should record the time and date that the Customer Account was ceased and by what method
CHM.BO.108.	The system should record the time and date that the Customer Account was suspended and by what method
CHM.BO.109.	The system should record the time and date that the Customer Account was ceased and by what method
CHM.BO.110.	The system should record the time and date that the Customer Account was Linked and by what method
CHM.BO.111.	The system should record the time and date that the Customer Account was Unlinked and by what method
CHM.BO.112.	The system should have a time and date that the suspension should resume to Normal state for the Customer Account
CHM.BO.113.	A Customer Account should contain a vulnerability indicator (true/false) to be used for do not disconnect
CHM.BO.114.	The Customer Account must contain an indicator to show that it is either a Tenant or a Leaseholder account, an additional type should be added which is internal
CHM.BO.115.	The system should provide a listing of the Account details for a specified period per Property, Block or Type (status, do not disconnect, tenant/leaseholder)
CHM.BO.116.	The system will be able to show the energy consumption at a Property level, organised by Block and also aggregated by Block (total consumption and average consumption)
CHM.BO.117.	The system will be able to show the consumption by Block recorded by the Bulk meter if one has been installed
CHM.BO.118.	The system will be able to show the comparison for the total consumption as the sum of all of the Blocks to the Bulk meter
CHM.BO.119.	The system will be able to show the average consumption of the Block and the averages for Properties within that Block that are of the same number of bedrooms (as provided in a reference). The comparison will be shown between the individual flat and the average for the group that it belongs to (all and number of bedrooms)
CHM.BO.120.	The system will be able to show the energy consumption per property for a date range and the reads (actual meter readings). If the Property has 2 meters, consumption will be summed and reads will be indicated as to which meter they came from
CHM.BO.121.	Bulk meter reads will be presented as consumption (for the Periods) and the reads (actual meter readings) for a date range
CHM.BO.122.	Properties with two meters will have an extended report showing the two meters separated out from the total. This includes both consumption and reads
CHM.BO.123.	The system will offer a prediction/forecast of consumption for a 12 month period

CHM.BO.124.	The solution will be able to show ranked order of energy consumers (based on meter and links back to Property/Customer Account) and be filtered based on Status (Active/Ceased/Suspended, Type (Tenant, Leaseholder, Prepaid, Postpay),
CHM.BO.125.	The solution should be able to produce an energy report which is a CSV file downloadable with Property identifiers and reads on a daily basis.
CHM.BO.126.	The solution will be allow for a financial manager to be able to create new rates for charging consumption. The rate in pence per kilowatt hour can be combined with a standing charge to form a Tariff. Tariffs will be able to be managed at an individual or block level
CHM.BO.127.	A standing charge can be a part of the Tariff. It will be expressed as pence per day and when used in the balance calculation will be charged by the minute. The standing charge will work within a Tariff structure and be configured along with a rate.
CHM.BO.128.	Payments can be manually entered into the system with credit going to an account. The source of that payment will be from the Back Office system
CHM.BO.129.	Adjustments (credit/debit) will be able to be made against an account at anytime.
CHM.BO.130.	Disconnection must be able to be restricted based on time and date. This will mean that disconnections are not allowed at those times by the system.
CHM.BO.131.	A vulnerability indicator will be recorded against an account that indicates this Customer Account will not be disconnected.
CHM.BO.132.	The system will be able to override the disconnection and turn the heat back on. This will be an administrative function performed by a sustainability officer
CHM.BO.133.	The system should show the Customer Account balance
CHM.BO.134.	The system will show the rolling balance of the Customer Account on a daily transaction log whereby for any given day the transactions for credits and debits with the corresponding resulting balance will be visible
CHM.BO.135.	The system will show the current Customer Account balances by Block and Property
CHM.BO.136.	While looking at a Customer Account, a historical log of activity should show disconnection and reconnection events with a time and date
CHM.BO.137.	Should be able to list closed accounts and unlinked accounts and the current Balance of those Accounts. Rejected income will be reported by the Payment system supplied outside of this system
CHM.BO.138.	System will send an alert for vulnerable Accounts going into emergency credit
CHM.BO.139.	System will send an alert for Account that is has been disconnected
CHM.BO.140.	System will produce a CSV export on a per Account basis that shows their transaction history. This will include credits/debits/adjustments with daily debits reflecting the consumption costs for that day. A date range can be selected for the export (to/from date)
CHM.BO.141.	System will produce a CSV export of the financial activity for a Block with individual Property and Accounts listed for a given date range.
CHM.BO.142.	System will show a list of meters that are not making contact with the server infrastructure. This will be shown as an ordered list by age of last contact
CHM.BO.143.	System will show a list of broadband lines that are down
CHM.BO.144.	System will show a list of screens that have not made contact with the server. This will be shown as an ordered list by age of last contact
CHM.BO.145.	The system will have a facility to generate energy statements on a Property by Property basis

As a part of the design process the system interactions have been designed to meet the requirements above.

ID	Name	Description
SI-21	Property Group creation	The ability to create property groups based on Estate and Block hierarchies. The Groups will be used for filtering, sorting and aggregating Properties.
SI-22	Property creation	The ability to create Properties that contain a unique reference number and are used in assignment to Customer Accounts, Meters and other in-home equipment. They should also be associated with any Groups that are relevant. This is considered a key link to Camden systems for reporting and assignment.
SI-23	Equipment sourcing	This is the process of batch numbering equipment that will be placed in the residential system so that any pre-configured associations between Tablet, Meter, PCB and Router can be made prior to installation. Intended to be used by installers and assembly partners. Should also contain serial numbers or unique identifiers and asset tags of interest to the process.
SI-24	Equipment installation	This is the process of assigning a batch of equipment to a Property at the time of installation and carrying out any quality checks on the installation, validating the associations of equipment and leaving the equipment in proper working order. Should also record the date and time of installation for reference.
SI-25	Equipment monitoring	The process that actively checks the equipment in the home and its functional parameters. It will send those parameters to the Fault Detection system to either trigger an alert, escalate, ignore or take some control of the equipment.
SI-26	Status report	A report that shows the status of the equipment, what its current readings the ability to sort and filter.
SI-27	Communications network install	The process of installing the broadband and mesh network elements, including any security settings and placement within the management tool.
SI-28	Communications network monitoring	The process that actively monitors the communications network operational performance including data connectivity, throughput, mesh routes, clients attached, etc.
SI-29	Fault detection	The central process the view monitoring data and generates a fault based applying rules on the data that is coming in to the Fault Detection system. Some basic alerting functionality will be provided, but alerts that are generated may need to be tracked and escalated within other operational systems outside of the CHMS.
SI-30	Equipment change	Equipment will be maintained and replaced over time and this process allows for the management of those changes. It will support the creation of new Equipment elements and the assignment of those elements to the various Property references and communications network elements.

SI-31	Infrastructure view	This is a general export and reporting area that provides access to the equipment that is providing CHM services. That includes the assignments and relationships between equipment and properties.
SI-32	User management	The Backoffice Portal will have users that will be given credentials to log in and use functionality. A system administrator will be allowed access to manage users, such as creation, deactivation, password changes and permissions/role assignment. Individual users will have some user management functionality that will allow them to change their password and contact details.
SI-33	Role management	The Backoffice Portal will restrict functionality based on Roles. Role management is the ability to create those Roles and enable the permissions on to functionality. For instance a Role may be Sustainability Manager that only has access to Energy related, not financial data on Properties.
SI-34	Heat distribution model management	This is the functionality that allows for central boilers to be created with size, type and other descriptive information can be held for analysis purposes. Also Properties will be able to be modelled against which generation and distribution system they are a part of.
SI-35	Gas reads	This is the process that generates Gas readings. This the input gas that is metered for some of the Boiler inputs. It may come from a file or be entered by hand from gas meter readings. This represents the raw energy input into the entire heating system. The reads will be in the same units across the entire system (kWh) and need caloric values to be applied before they are put into the system.
SI-36	Bulk meter reads	This is the process that generates Bulk meter readings. Bulk meters are reading the hot water energy that has been generated within the Boiler plant. The difference between the Gas input energy and the Bulk meter read is the boiler's efficiency in converting gas into hot water. The Bulk meter read may have energy, power, temperature, flow rate and other parameters that must be captured.
SI-37	Meter readings	This is the process that generates meter readings. The meter readings are at an individual residential level with the difference between the Bulk reads and the Meter reads being the efficiency of the distribution system (i.e. losses in the pipework). The Customer Account will be billed against the kWh energy of the Meter readings. The other readings such as power, temperature and flow will be transmitted and used as input into analysis and presentation for Customer and Backoffice facing functionality.
SI-38	Tariff management	This is the process of maintaining the tariffs or unit rates that will be applied to the Customer Account and/or the consumption of energy. The tariff is the price to the Resident. This may be a combination of standing charge and rate per kWh consumed. The Balance of the Customer Account will be debited at the rate specified by the Tariff that is assigned to their account. Note: there can be multiple Tariffs in the system however only one Tariff can be assigned to a Customer Account at any one time.
SI-39	Event management	Events are a general term used to capture something that has happened at a particular time. This functionality is a general facility for capturing and managing Events that might be used for future

		analysis. This might be a record of a system outage or fault that needs to be recorded.
SI-40	Time aggregate calculations	This is the process that calculates energy and usage statistics over various time periods. These are the aggregates that are used for reporting, billing and general analysis. They are at the heart of the Core Heat Metering System. An example would be the number of kilowatt hours consumed by a Customer Account from the 1 Feb to the 28 Feb or the day by day amount consumed across that period. More advanced to that, would be the average amount of energy consumed by a Block, Estate or other group for a time period. This function makes extensive use of the data that has been recorded in the CHMS database.
SI-41	Cost calculation	This is the process that takes the Tariff and Customer Account (including consumption from the Account) and calculates the cost. This is a real-time running process that will debit the Customer Account Balance.
SI-42	Financial view	This is a general process description of the export and reporting functionality that is required for the financial aspects of the Core Heat Metering System. This includes reports/exports of Balances by Property, spend per Property for various time periods, etc. Also there is the ability to view transactions by Customer Account, credits, debits and adjustments.
SI-43	Financial adjustments	Financial adjustments are administrative credits and debits that are entered to Payments API. For instance a credit may be given.
SI-44	Customer Account Edit	This is the ability to edit the Customer Account details and change the status or assignments of the Customer Account. This is meant to be a super user type function that allows deep edits within the CHMS data structures. In particular attributes like, Do Not Disconnect (DND) can be toggled within this area. Customer Account Creation has been captured in the CAP components description in **¡Error! No se encuentra el origen de la referencia.**.
SI-45	Energy model management	Although this may not be readily clear, it is meant to reflect the capture of an energy model. For example, you may want to actively record the Bulk meter reads subtracting all of the Meter reads that are downstream from it. This model might write data into a new timeseries that can be displayed as an ongoing efficiency measure. Also, internal to the Core Heat Metering System, things like comparisons and benchmarks are stored as models. This functionality may be reserved for internal Hildebrand staff.
SI-46	Energy measures	This is the process of reading and naming the output from the Energy Models. It is a report or export of energy measurements derived from the readings and data/algorithms from the Model.
SI-47	Energy view	This is the overall reporting/export function of all of the energy Measures, including the meter reads.
SI-48	Household energy feedback	This is the area that allows for the communication of messages and the assignment of a Customer Account into a comparative group. The Feedback are the measures (statistical or direct readings)

		provided to the Tablets and the CAP, plus any messages that might get sent to the Customer Account. The messages are managed and originated from this functionality.
SI-49	Tablet view management	This is the functionality that manages the view of the Tablets.

From the system interactions, a navigation framework has been constructed.

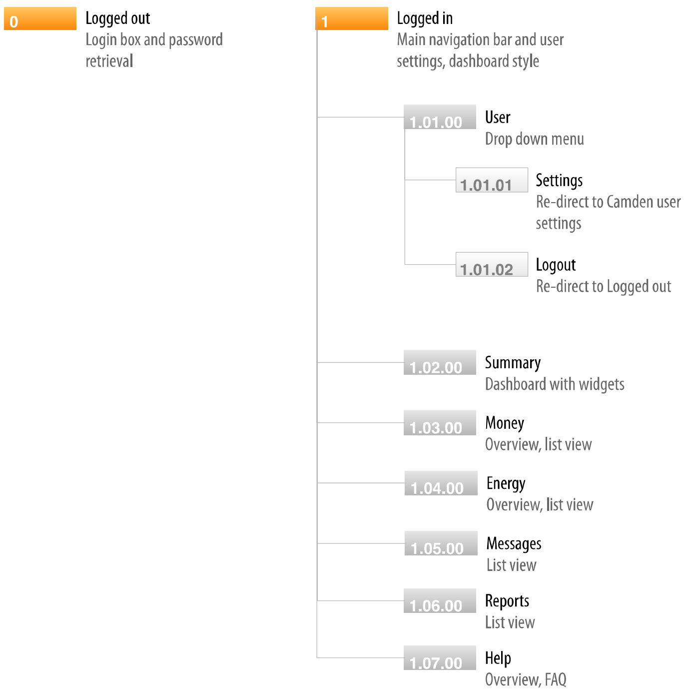

Figure 19 Backoffice navigation structure

The following is a short synopsis of the intention of each of the areas that corresponds to the navigation element.

- **Login** (logged out Home) – this represents that page that will ask for a username and password in order to authenticate users. There is a function to send a password to recall the password. This is the default view when navigating to the Backoffice portal URL.
- **Home** – this the page just after login, which will be like a dashboard showing the main navigation that is allowed for the role that is logged in. Messages and alerts should be shown on this page.
- **Property** – most of the management of the system will be from an index of a property. This allows for users of the Backoffice portal to find information about the state of the heating system at a particular property and the Customer Account information linked that that Property. There is cross navigation to Accounts such that Account functions are performed at that detailed level.
- **Account** – this give the BO portal user the ability to find information about the Customer Account regardless of what Property it is linked to (or in some cases the

Account might not yet be linked), transactions against the Account and the ability to make adjustments to the Account will be in this area.

- **Energy** – group level (estate and block) energy consumption, plus the ability to compare groups of flats (for instance of similar size) will be the primary view of this area. Individual Property consumption information can be viewed and the generation of things like Energy reports will be provided in this area.
- **Finance** – accounting functions to view transactions across the whole heat payment scheme will be the focus of this area. Financial controllers should be able to see the total credit and debit accounting logs. This is also the area that tariffs will be managed, however they can be assigned At the Account level as well.
- **Operations** – mostly pertaining to the safe and efficient operation of the metering and in-home assets, plus the communications network that brings data back to the main cloud service.
- **Reports** – a log of the reports that have been run in various export functions, plus any system generated reports that are scheduled or triggered from the backend.
- **User Settings** – individual management of passwords and any self service user settings. This is not to be confused with the accounts that are for residents which are held in the CAP. This is only the users that have access to the Backoffice portal.
- **Role Management** – roles define the entitlement/rights to perform functions within the Backoffice portal, for instance a generic Manager role can create new Officers.
- **User Management** – ability to see all of the users of the Backoffice system, it will be restricted to Superuser and Manager level permissions.
- **Signout** – this will redirect to the Login page

7.1.6. Payments System

The payment system provides the interface to credit funds on to the Customer Account Balance. It also provides some management interfaces to the Customer Account, notably creation of the Customer Account and the ability to change the status of Customer Accounts. By design, this system is quite independent from the CHMS and has the ability to de-couple from the Property and Meter aspects of the system.

The motivation for this separation is for ease of implementation of security, adaptability to other payment mechanisms and portability of Account Balances between Properties.

ID	Name	Description
SI-50	Cease account	A Customer Account may be ceased if the Customer Account is no longer required. This means the Account no longer accepts payments, but the identifier of the Account is not reused and historical transactions can still be reported. A ceased state is the only one that Balances can be zeroed and cash returned to the Customer Account.
SI-51	Un-cease account	This is a change of Account status to take it back into normal operation.

SI-52	Freeze account	A Customer Account might need to be frozen and reject credits and debits, however this may be while the Customer is changing Properties or has an administrative issue. For this reason adjustments can be posted to a frozen account.
SI-53	Unfreeze account	Ability to reverse the state of frozen and take an account back to normal operation.
SI-54	Disconnect account	This turns on and off the flag that allows for the heat to be disconnected as a result of payment rules. If the disconnect flag is false, then the heat can not be disconnected. This function will set the disconnect flag to true, which is the default state.
SI-55	Un-disconnect account	This reverses the disconnect flag and sets it to false.
SI-56	Unlink Property	This unlinks the Property from a Customer Account. Credits and adjustments can still be made, by there is no consumption recorded against the Customer Account, so no debits will practically be transacted.
SI-57	Link Property	Provides a function for the linkage of a Customer Account to a Property. No validation is done within this system, so any validation/authorisation must be implemented before this function is called.
SI-58	Create Account	An account must be created and as a result an Account number will be generated and returned. The Customer Account can then be used to credit and link to a Property.
SI-59	View Account	A Customer Account has various attributes, states, associations and transaction records against it. The view Account functionality provides a view/export of that information.

7.2 Madrid Application

7.2.1. Integration with COSMOS

The Madrid EMT application is at a conceptual level for Year 1 and would be implemented as a mobile native application in order to make use of mobile operating system's communication and location services. Figure 20 shows the local data structures that will be relied on for application development with the Application Data being used for virtual entity data to power the application.

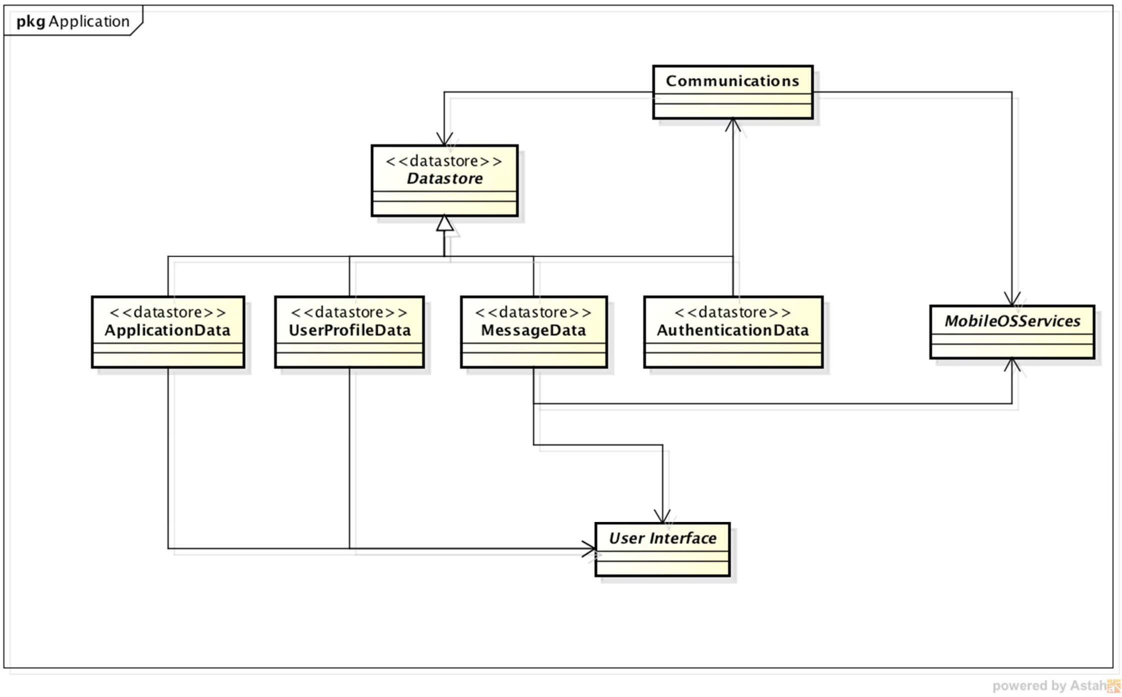

Figure 20 Mobile application architecture implemented on a handset through native OS services. This is for both Android and IOS

User Profile and Message Data are the responsibility of the application developer, including any support for creation of users and addressability of the messages.

The Application Data representing the virtual entity will be delivered through the Reactive Box framework which will be brokered via the COSMOS platform. Reactive Box will be described in more detail in Year 2.

8 COSMOS Devices – Open Things

OpenThings is a lightweight messaging protocol for sending reports (e.g. temperature measurement, power reading) and commands (e.g. turn on socket, set dimmer level) between small sensors and connected devices. The protocol is intended for use in simple applications with point-to-point or star network topologies. In both cases one device is a master and remaining devices are slaves.

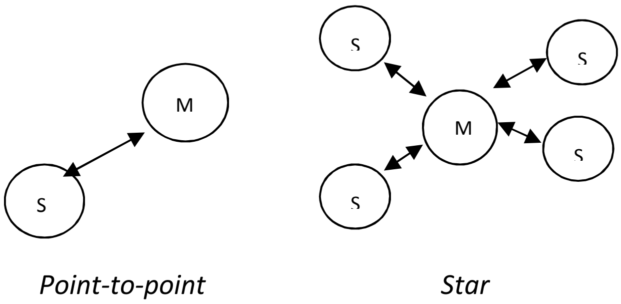

Figure 21 Openthings network topologies

The protocol provides basic message structuring and data validation (CRC). Messages are transmitted in one direction at a time without acknowledgement. The protocol does not contain any encryption, quality of service, anti-collision or network routing elements. These can be implemented in additional network layers if required; the goal of this protocol is only to define a universal message structure and framework for how data is described, represented and sent. The protocol is hardware agnostic, it can be used with any suitable transport means; for example a 433MHz radio or serial bus connection.

8.1 Transmissions

Each transmission has a defined messaging structure containing one or more records. Each record contains a parameter identifier (e.g. temperature, light level, voltage) and type description (e.g. integer, float, character) of the data value to be sent. In this way the protocol is extensible and allows new categories of data to be described and sent without any changes to the underlying protocol.

Figure 22 Openthings example transmissions

8.2 Addressing

In OpenThings network topologies only slave devices have addresses. An address is the combination of the Manufacturer ID, Product ID and Sensor ID and must be unique. Slave devices will only act on messages sent containing their address and only originate messages with their address. Master devices listen to all messages sent on the network (other than messages they have sent themselves) and originate messages with the address of the slave they want to send the message to. The protocol does not allow broadcasting; all messages must be individually addressed.

This example shows the exchange of messages and addressing when a slave device reports a parameter:

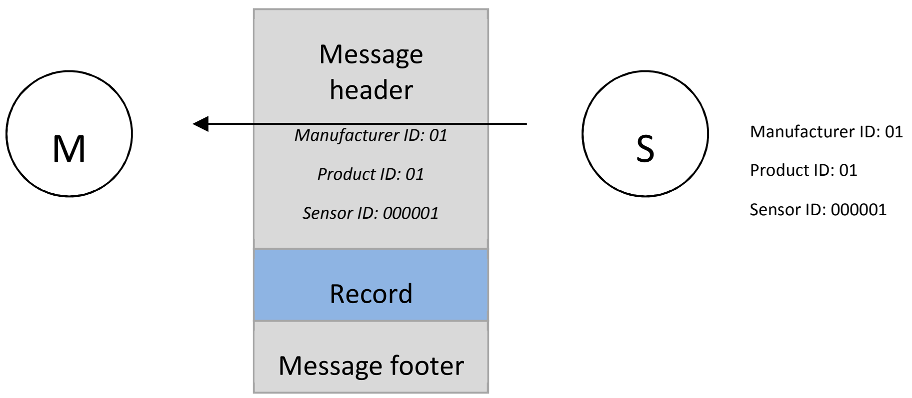

Figure 23 Report parameter example

This example shows the exchange of messages and addressing when a master commands a parameter on a slave device and the slave device is expected to respond.

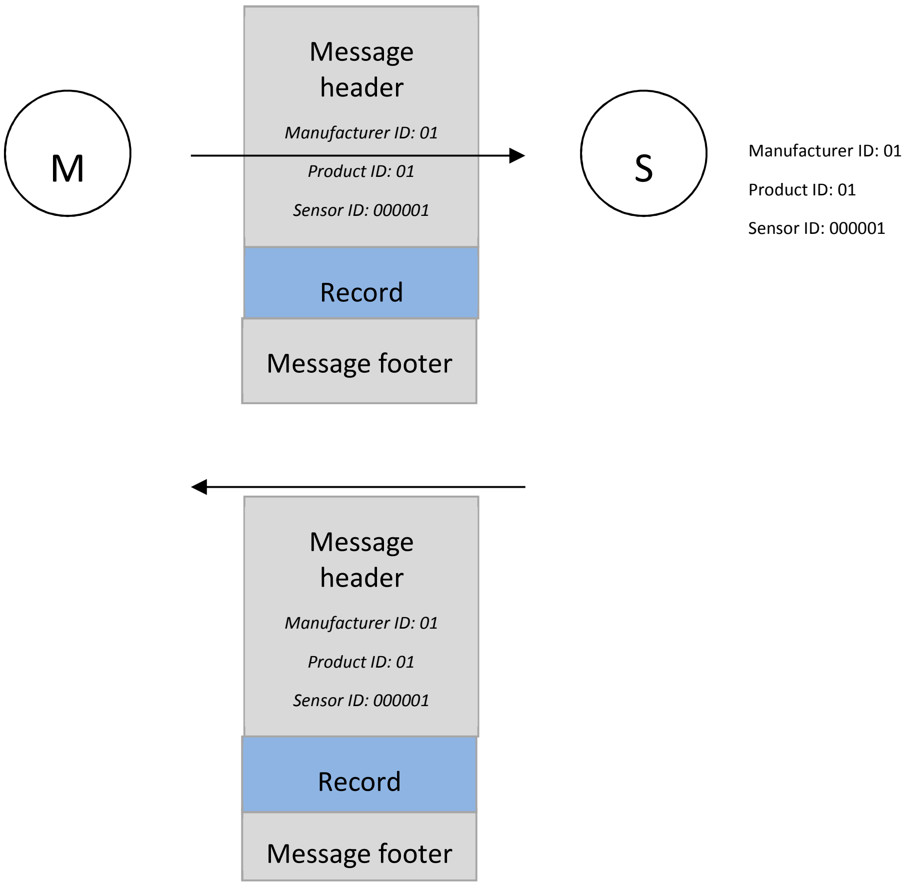

Figure 24 Command parameter example

8.3 Default dictionary of parameter identifiers

Table 1 below lists the parameters that can be reported and commanded as part of the default parameter identifier dictionary. Manufacturers may extend (but not modify) this to meet their requirements.

Parameter	Char	Hex (Report)	Hex (Command)	Units
Alarm	!	0x21	0xA1	*See "Alarms"*
Debug Output	-	0x2D		*Reserved for debug*
Identify	?	0x3F	0xBF	*See "Identify"*
Source Selector	@		0xC0	*See "Multiple Commands"*
Water (Flood) Detector	A	0x41	0xC1	1 = flooded, 0 = dry. *See "Binary Parameters"*
Glass Breakage	B	0x42	0xC2	1 = broken, 0 = intact. *See "Binary Parameters"*
Closures	C	0x43	0xC3	Curtains/blinds; 1 = open, 0 = closed. See *"Binary Parameters"*
Door Bell	D	0x44	0xC4	1 = pressed, 0 = released. See *"Binary Parameters"*
Energy	E	0x45	0xC5	kWh
Fall Sensor	F	0x46	0xC6	1 = fall, 0 = no fall. *See "Binary Parameters"*

Gas Volume	G	0x47	0xC7	m^3
Air Pressure	H	0x48	0xC8	mbar (millibar)
Illuminance	I	0x49	0xC9	Lux. *See "Illuminance/Light Level"*
Level	L	0x4C	0xCC	*See "Generic Level"*
Rainfall	M	0x4D	0xCD	mm
Apparent power	P	0x50	0xD0	VA
Power factor	Q	0x51	0xD1	
Report Period	R	0x52	0xD2	s (seconds). See *"Sensor Report Frequency"*
Smoke Detector	S	0x53	0xD3	1 = smoke detected, 0 = no smoke. *See "Binary Parameters"*
Time and Date	T	0x54	0xD4	Seconds since Epoch (1 Jan 1970)
Vibration	V	0x56	0xD6	1 = vibration detected, 0 = no vibration. *See "Binary Parameters"*
Water Volume	W	0x57	0xD7	l (litres)
Wind Speed	X	0x58	0xD8	m/s
Gas Pressure	a	0x61	0xE1	Pa
Battery Level	b	0x62	0xE2	V
CO Detector	c	0x63	0xE3	1 = gas detected, 0 = no gas detected. *See "Binary Parameters"*
Door Sensor	d	0x64	0xE4	1 = open, 0 = closed. *See "Binary Parameters"*
Emergency (Panic Button)	e	0x65	0xE5	1 = emergency, 0 = no emergency. *See "Binary Parameters"*
Frequency	f	0x66	0xE6	Hz
Gas Flow Rate	g	0x67	0xE7	m^3/hr
Relative Humidity	h	0x68	0xE8	%
Current	i	0x69	0xE9	A
Join	j	0x6A	0xEA	No data. See *"Joining"*
Light Level	l	0x6C	0xEC	Unsigned int, 0 = off, max = full on. *See "Illuminance/Light Level"*
Motion Detector	m	0x6D	0xED	1 = motion, 0 = no motion. *See "Binary Parameters"*
Occupancy	o	0x6F	0xEF	1 = room occupied, 0 = room not occupied. See "*Binary Parameters"*
Real Power	p	0x70	0xF0	W
Reactive Power	q	0x71	0xF1	VAR
Rotation Speed	r	0x72	0xF2	RPM
Switch State	s	0x73	0xF3	1 = on, 0 = off. *See "Binary Parameters"*
Temperature	t	0x74	0xF4	Celsius
Voltage	v	0x76	0xF6	V
Water Flow Rate	w	0x77	0xF7	l/hr (litres/hour)
Water Pressure	x	0x78	0xF8	Pa

Table 1: Default dictionary

Devices are not required to support all parameters and the dictionary does not in general define the type of parameters (See Table 7); whether Energy is reported as a 32-bit unsigned integer, 48.16 bit fixed point or even as an unhelpful character string is up to the manufacturer, as long as it is reported in kWh.

Some parameters can also be set as a command by a master device or very occasionally by a slave; this is signalled by setting the top bit of the parameter identifier (i.e. adding 0x80 to the hex number). Thus a smart plug might report that its switch was off with parameter identifier report “0x73”, and a gateway might command it to turn the switch on with the parameter identifier command “0xF3”.

8.4 Joining

The default joining protocol is simply a device announcement. The slave device sends a Join command (parameter identifier “j” command, 0xEA) and waits for a listening gateway to send it a Join report (parameter identifier ‘j’ report, 0x6A). Neither join record carries any data, so should have a type of 0x00 (See Table 7).

8.5 Keep-Alive

The default parameter identifier dictionary does not define any heartbeat or keep-alive record identifier. In the interests of brevity, devices wishing to send a keep-alive message that have nothing else to send can simply send a message with no records in it; in other words, they just send bytes 0 to 7 and x to x+2 from Table 5.

8.6 Identify

The “Identify” parameter (parameter identifier “?” report, 0x3F) may be used to report version numbers or model names. More usefully, as a command (0xBF) it requests the receiver to identify itself, for example by flashing an LED or beeping for a few seconds.

8.7 Sensor Report Frequency

Some sensors make periodic reports of their data. The “Report Period” parameter (parameter identifier “R”) allows the device to announce how often these reports will be made, and may allow a master device to command the time between reports if the slave device allows. By convention, a report period of zero indicates that a sensor will or should report parameter values when they change rather than at fixed intervals.

8.8 Multiple Reports

Some devices may have multiple sources of information. Consider monitoring three-phase mains, for example; the device would need to report the voltage, current and power for each phase. This is done by sending multiple records with the same identifier in the same message. So to report three-phase real power, for example, a device would send a message containing a ‘p’ record containing the power on phase 1, another ‘p’ record containing the power on phase 2, and a final ‘p’ record containing the power on phase 3.

8.9 Multiple Commands

If a slave device has more than one commandable parameter, a master device may need to command only one parameter out of a set; for example to set the light level of one dimmer switch of a pair, or turn on one switch out of a multiswitch gang. To avoid any race conditions it must send a record containing the source selection parameter (parameter identifier "@" command) with a bit field of the indices of the parameter to command immediately before the part of the record setting the parameter.

For example to turn on the first (bit 0) and third (bit 2) switch of a multiswitch gang the master should send the following record (see Record Structure section for details of the elements):

Byte sequence	Value	Note
1	0xC0	Parameter identifier "@" command
2	0x01	Unsigned normal integer, data length 1
3	0x05	Bitfield, bits 0 and 2 set
4	0xF3	Parameter identifier "s" command
5	0x01	Unsigned normal integer, data length 1
6	0x01	Command switch on

Table 2: Example record to turn on first and third switch of a multiswitch gang

However to turn on the first switch and turn off the third switch a single source selection parameter cannot be used. Instead two source selection parameters commands must be sent in the same record:

Byte sequence	Value	Note
1	0xC0	Parameter identifier "@" command
2	0x01	Unsigned normal integer, data length 1
3	0x01	Bitfield, bit 0 set
4	0xF3	Parameter identifier "s" command
5	0x01	Unsigned normal integer, data length 1
6	0x01	Command switch on
7	0xC0	Parameter identifier "@" command
8	0x01	Unsigned normal integer, data length 1
9	0x04	Bitfield, bit 2 set

10	0xF3	Parameter identifier “s” command
11	0x01	Unsigned normal integer, data length 1
12	0x00	Command switch off

Table 3: Example record to turn on the first switch and turn off the third switch of a multiswitch gang

8.10 Alarms

The “Alarm” parameter (parameter identifier “!” report) is intended for conditions internal to the device that may need to be reported urgently. There is no obligation on a device to report these conditions. The parameter value is one or more characters defining the alarm conditions, as follows:

Condition	Char (set)	Hex (set)	Char (clear)	Hex (clear)	Comments
Low Battery	B	0x42	b	0x62	The device’s battery is failing and must be replaced soon.
Power Fail	P	0x50	p	0x70	A mains-powered device is currently running off its battery backup.
Over Temperature	T	0x54	t	0x74	The device is running unexpectedly hot and needs attention.
Tamper	Z	0x5A	z	0x7A	The device’s anti-tamper mechanism has been triggered.

Table 4: Default Alarm Values

The device may report that an alarm condition no longer applies by sending an alarm with the lower case version of the condition character in its parameter value. For example, a device whose ventilation has been fixed might send the record “0x21 0x71 0x74” to indicate that its temperature is back within normal operating parameters.

Alarms can be cancelled from a master device by sending an alarm parameter identifier “!” command.

8.11 Generic Level

“Generic Level” is intended for reporting and controlling generic devices; its value is an unsigned integer where zero indicates fully off, and the maximum value as defined by the type indicates fully on.

8.12 Illuminance/Light Level

“Illuminance” is intended for reports from light sensors, and gives the amount of illumination in Lux. “Light Level” is intended for reporting and controlling dimmer switch settings; its value is an unsigned integer where zero indicates that the switch is fully off, and the maximum value as defined by the type indicates that the switch is fully on.

8.13 Binary Parameters

A number of the parameters describe binary states, such as a switch being on or off. All these parameters are "active high", i.e. they report their active state ("the switch is on") with a value of 1 and their inactive condition with a value of 0.

Some slave devices may have multiple sources for the same binary parameter, such as individually controllable switches on a multiswitch gang. A slave may compress the different instances of the binary parameter into a single bitfield, with each bit reporting a different instance. So for example a strip of six switches could report that switches 0 and 3 were on and all the others off with a data value of 0x09 (binary 00001001). Binary parameters cannot be commanded using this shorthand; to avoid race conditions, master devices must use the "Source selection" parameter as described above.

Message structure

Each transmission is sent according to the following message structure:

	Byte Num.	Data Name	Remark
Message header	0	Remaining length [7:0]	Number of bytes in whole message excluding this byte(x+1) – *see Note 1*
	1	Reserved bit [7]	Must be zero – see *Note 2*
		Manufacturer ID [6:0]	Manufacturer identifier – see *Note 3*
	2	Product ID [7:0]	Product identifier – see *Note 4*
	3	Reserved [15:8]	Reserved, default 0x0000 – see *Note 5*
	4	Reserved [7:0]	
	5	Sensor ID [23:16]	Unique Sensor ID to allow differentiation between end devices – see *Note 6*
	6	Sensor ID [15:8]	
	7	Sensor ID [7:0]	
Records	ˆ		
	Bytes 8 to x-1 contain one or more Records as shown in Table 6		
	ˇ		
Message footer	x	End of data [7:0]	NULL (0x00) to indicate end of data
	x+1	CRC [15:8]	CRC-16-CCITT – see *Note 7*
	x+2	CRC [7:0]	

Table 5: Message structure

The message header and footer are mandatory for each message transmission. Each message may contain one or more records, up to a maximum message length of 256 bytes. Bit numbers are shown in square brackets “[ ]”.

Note 1: This byte can be used by the receiver to determine the length of the message to be received. Some radio chipsets with a packet handler can use this byte to automatically receive variable length messages.

Note 2: This reserved bit will always be zero in this version of the protocol.

Note 3: The Manufacturer ID is a globally unique number from 0 to 127 inclusive. It is allocated on request and published on the OpenThing’s website.

Note 4: The Product ID should be used to identify the type/model of the product (e.g. smart plug, temperature sensor, PIR sensor etc). There is no reservation on Product IDs and manufacturers are expected to allocate their own; however, devices with Product IDs from 0 to 127 inclusive are expected to use a subset of the default parameter dictionary detailed above.

Note 5: These bytes are reserved for application/manufacturer specific functions. For example they could be used for an encryption seed. If not used the default is NULL (0x0000).

Note 6: The Sensor ID must be unique to each end device of a given Product ID within the same radio network. To ensure this is the case it is recommended that the Sensor ID should be incremented (+1) sequentially in production for a given Product ID by the manufacturer.

Note 7: See Appendix A for more details on the cyclic redundancy check ("CRC") validation. The CRC is calculated from byte number 5 to byte number x inclusive, i.e. over the Sensor ID, the message body and the end of data marker. If the message is encrypted, the CRC should be performed over the unencrypted data.

8.14 Record structure

Each record has the following structure:

Byte Num.	Data Name	Remark
1	Parameter identifier [7:0]	The identifier of the data being sent – see *Note 8*
2	Type description [7:0]	The type of the data being sent including its length z – see *Note 9*
ˆ		
Bytes 3 to 3+(z-1) contain the data value for the parameter		
ˇ		

Table 6: Record structure

Note 8: The parameter identifier indicates the physical measurement or control being communicated (e.g. power, frequency, switch state). Devices with Product IDs between 0 and 127 inclusive are expected to use parameter identifiers from the default parameter identifier dictionary, otherwise manufacturers are expected to define their own dictionary of parameter identifiers for each product.

The parameter identifier "j" (0xEA as a command, 0x6A as a report) has been reserved as a special parameter to allow slave devices to request to join/pair to a master device (typically a display or gateway) and must be included in each dictionary. If manufacturer products have a joining/pairing process it should be implemented around this command and response pair.

The parameter identifier of value 0x00 is forbidden.

Note 9: The first 4 bits of a record type description byte define the type of the data value being sent (e.g. integer, float). The last 4 bits define the length of the data value being sent (in bytes):

7	6	5	4	3	2	1	0
Type				Length			

Table 7: Type description byte

7	6	5	4	Type	Decimal point
0	0	0	0	Unsigned integer	x.0 normal integer
0	0	0	1		x.4 fixed point integer
0	0	1	0		x.8
0	0	1	1		x.12
0	1	0	0		x.16
0	1	0	1		x.20
0	1	1	0		x.24
0	1	1	1	Characters	
1	0	0	0	Signed integer	x.0 normal integer

1	0	0	1		x.8 fixed point integer
1	0	1	0		x.16
1	0	1	1		x.24
1	1	0	0	Reserved	
1	1	0	1	Reserved	
1	1	1	0	Reserved	
1	1	1	1	Floating point	IEEE 754-2008

Table 8: Type definition within type description byte

It is entirely legal for any record to have a zero length data value, like the example "join" record below. Any type may be used, but a type description of 0x00 (unsigned unshifted integer) is conventional. Multi-byte numerical data values are sent big-endian, i.e. with the most significant byte first; see the example "power" record below.

8.15 Example records

Byte num.	7	6	5	4	3	2	1	0	Note
1	1	1	1	0	1	0	1	0	Parameter identifier "j" command
2	0	0	0	0	0	0	0	0	Unsigned normal integer, data length 0

Table 9: Example "join" command record

Byte num.	7	6	5	4	3	2	1	0	Note
1	0	1	1	1	0	0	0	0	Parameter identifier "p" report
2	1	0	0	0	0	0	1	0	Signed normal integer, data length 2 bytes
3	x	x	x	x	x	x	x	x	Power [15:8]
4	x	x	x	x	x	x	x	x	Power [7:0]

Table 10: Example "power" report record

Byte num.	7	6	5	4	3	2	1	0	Note
1	0	1	0	1	0	0	1	1	Parameter identifier "S" report
2	0	0	0	0	0	0	0	1	Unsigned normal integer, data length 1
3	0	0	0	0	0	0	0	x	State [0]: 1 = smoke detected, 0 = no smoke.

Table 11: Example "smoke detector" report record

8.16 CRC validation

The CRC can be used by the application layer to check the validity of a message. The CRC-16-CCITT checksum is implemented using the divisor polynomial of 0x1021 or (x16 + x12 + x5 + 1). The `crc` function will calculate the 16 bit CRC for an array of message bytes using a code space efficient algorithm. Other algorithms are available optimised for speed but considering the low data rate, this algorithm is preferable.

```
int16_t crc(uint8_t const mes[], size_t siz)
{
	uint16_t rem = 0;
	size_t byte, bit;

	for (byte = 0; byte < siz; ++byte)
	{
		rem ^= (mes[byte] << 8);
		for (bit = 8; bit > 0; --bit)
		{
			rem = ((rem & (1 << 15)) ? ((rem << 1) ^ 0x1021) : (rem << 
1));
		}
	}
	return rem;
}
```

Figure 25: CRC validation code extract

8.17 Example radio settings

These are the radio settings used as default by Hildebrand's tablet reference designs.

Transmission frequency	434.300 MHz
Deviation	± 30kHz
Radio FSK data rate	2400 baud Manchester (encoded at 4800 bits/s)
Message transmission interval	10s
Preamble	3 bytes (0xAA , 0xAA, 0xAA)
Sync bytes	2 bytes (0x2D , 0xD4)

Table 12: Radio physical settings

For radio applications it is it is recommended that the total message length is less than 64 bytes.

9 Conclusions

This document has shown the Use Cases, domain specific background information and application design for the three test bed cities for COSMOS. The physical systems and input data represented as virtual entities (VEs) will be used in conjunction with COSMOS to deliver applications that are socially aware.

The implementation strategy is to make the data sources available to the other work packages while trying to think through the practical limitations of the city as a target user of the COSMOS system. Many of the advanced technologies and concepts in COSMOS will be difficult for city ICT and city business unit managers to understand, therefore WP7 must also provide a user experience that works to deliver services.

New details of communications protocols and device specifications have been added such that COMOS developers can work within the platforms that will be deployed within test beds.

A platform called UrbisAPIs will be the implementation during year 2 of a fully running instance of COSMOS that will be available to the general public for use. Public data source will be made available within the UrbisAPIs platform from the test beds of Camden, Madrid and in Year 3 Taipei.