Energy Management for the Smart Home

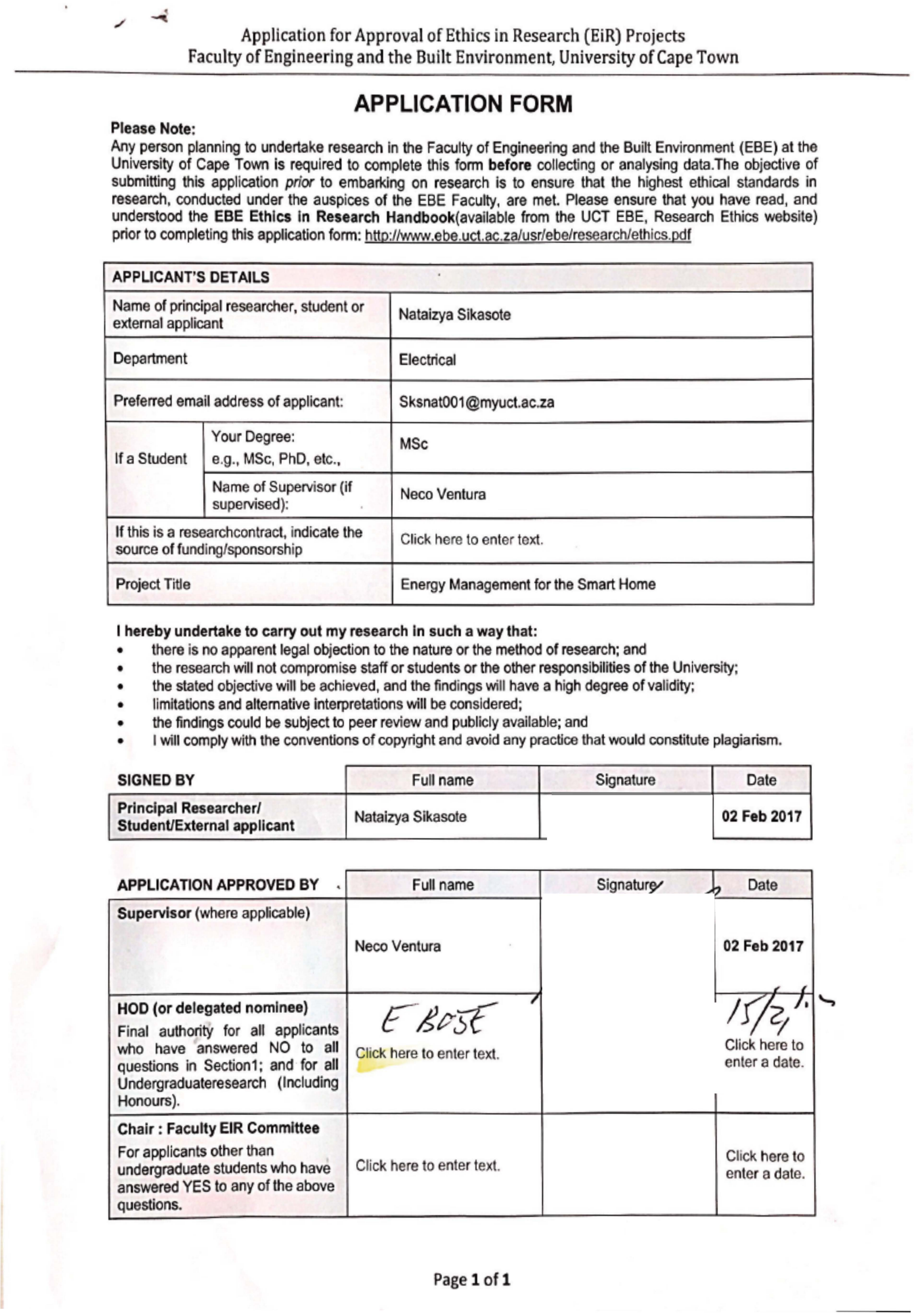
Nataizya Sikasote

Supervisor:

Neco Ventura

Dissertation submitted to the Department of Electrical Engineering in fulfilment of the requirements for the Master of Science in Electrical Engineering at the University of Cape Town

July 2017



Application for Approval of Ethics in Research (EiR) Projects
Faculty of Engineering and the Built Environment, University of Cape Town

APPLICATION FORM

Please Note:

Any person planning to undertake research in the Faculty of Engineering and the Built Environment (EBE) at the University of Cape Town is required to complete this form **before** collecting or analysing data.The objective of submitting this application *prior* to embarking on research is to ensure that the highest ethical standards in research, conducted under the auspices of the EBE Faculty, are met. Please ensure that you have read, and understood the **EBE Ethics in Research Handbook**(available from the UCT EBE, Research Ethics website) prior to completing this application form: http://www.ebe.uct.ac.za/usr/ebe/research/ethics.pdf

APPLICANT'S DETAILS		
Name of principal researcher, student or external applicant		Nataizya Sikasote
Department		Electrical
Preferred email address of applicant:		Sksnat001@myuct.ac.za
If a Student	Your Degree: e.g., MSc, PhD, etc.,	MSc
	Name of Supervisor (if supervised):	Neco Ventura
If this is a researchcontract, indicate the source of funding/sponsorship		Click here to enter text.
Project Title		Energy Management for the Smart Home

I hereby undertake to carry out my research in such a way that:

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

SIGNED BY	Full name	Signature	Date
Principal Researcher/ Student/External applicant	Nataizya Sikasote		**02 Feb 2017**

APPLICATION APPROVED BY	Full name	Signature	Date
Supervisor (where applicable)	Neco Ventura		**02 Feb 2017**
HOD (or delegated nominee) Final authority for all applicants who have answered NO to all questions in Section1; and for all Undergraduateresearch (Including Honours).	E BOSE Click here to enter text.		15/2/17 Click here to enter a date.
Chair : Faculty EIR Committee For applicants other than undergraduate students who have answered YES to any of the above questions.	Click here to enter text.		Click here to enter a date.

Page **1** of **1**

Declaration

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This dissertation has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Signed by candidate

Nataizya Sikasote

July 2017

Supervisor’s Declaration

As the candidate’s supervisor, I have approved this dissertation for submission.

Acknowledgements

I would like to express my sincere gratitude to the following people for their help, support and encouragement through this project.

My parents, brothers, extended family and friends for their continuous support and encouragement.

Neco Ventura, for his supervision and guidance.

Nyasha Mukudu, Kurai Gombiro, Fredrick Chisanga and Udit Paul for their technical and writing advice.

Finally, I would like to thank God for guiding me from the beginning of this project to its completion.

Abstract

Utility providers are under constant pressure to meet the ever-increasing demand for energy with a finite production capacity. Due to this, utility providers need to ensure that the demand does not exceed the supply. The use of demand response programs has been used as a solution to better match the available supply to the demand experienced on the grid.

While demand response programs may improve the relationship between the supply and the demand, increasing consumer participation can further improve the effectiveness of demand response programs. The level of consumer participation is highly dependent on the incentives offered and the convenience of participating in the demand response program. However, most of these programs require consumers to actively monitor the available services and take appropriate action on their appliances.

Home energy management systems attempt to provide convenience to consumers as well as increase their participation in demand response programs. They do this by enabling autonomous control and remote control of appliances. In addition, a home gateway makes it possible for the consumer and utility provider to interact with devices in the home remotely. Current solutions host the home energy management software on a home gateway or on a dedicated device in the home. Furthermore, these systems make use of proprietary software and devices to monitor and control the home environment.

However, proprietary systems are costly to implement and maintain due to their dependence on the developers and the varying needs of consumers. This dissertation proposes a home energy management system solution that makes use of a standardized distributed machine-to-machine (M2M) middleware framework to support heterogeneous devices, technologies and protocols. In addition, the proposed solution shifts the software logic of the home energy management system from the gateway to a private cloud. This allows the system to leverage the benefits of virtualization and cloud computing such as cost efficiency, faster deployment and easier maintenance of the system.

The proposed solution was implemented using a European Telecommunications Standards Institute (ETSI) compliant distributed M2M middleware framework (i.e. OpenMTC) and an open-source private cloud platform (i.e. OpenStack). A prototype was developed and tested with demand response programs that included a control demand response (DR) service, a consumption monitoring service and a time-of-use (ToU) service. In addition, the prototype made use of various third party applications, protocols and devices to support the aforementioned demand response services and provide automated and remote control of home appliances. Finally, an evaluation of the proposed system was conducted and concluded that the number of home energy management systems subscribed to the defined services influenced the effectiveness of these services because of the additional processing that the utility provider is required to perform.

Table of Contents

1. Introduction 1

1.1 Research Motivation 2

1.1.1 Problem Definition 3

1.1.2 Research Questions 4

1.2 Dissertation Objectives 5

1.3 Scope and Limitations 5

1.4 Dissertation Outline 6

2. Background and Related Work 8

2.1 Demand Response 8

2.2 Smart Homes 10

2.3 Current Solutions 13

2.3.1 Middleware 16

2.3.2 Cloud Computing 20

2.4 Chapter Discussion 24

3. Requirements & Functional Architecture 26

3.1 Stakeholders & their requirements 26

3.1.1 Utility Provider 27

3.1.2 End-Users 28

3.1.3 Home Area Network (HAN) 29

3.2 Technical Requirements 30

3.3 Energy Management Services 32

3.3.1 Control Demand Response 32

3.3.2 Time-of-Use 33

3.3.3 Consumption Monitoring 34

3.4 Functional System Architecture 34

3.5 Chapter Discussion 36

4. Implementation of the SHEMS and Evaluation Platform 38

4.1 SHEMS 38

4.1.1 Utility Provider Applications 40

4.1.2 End-User Applications 44

4.2 HAN ... 49

4.2.1 Home Gateway ... 50

4.2.2 Load Controllers ... 50

4.2.3 Transceivers ... 51

4.3 Performance Metrics and Evaluation Tools ... 51

4.3.1 Energy Management Services ... 52

4.3.2 Device Emulator ... 53

4.3.3 Experimental Setup ... 55

4.4 Chapter Discussions ... 56

5. Evaluation Results & Analysis ... 58

5.1 Experimental Scenarios ... 58

5.1.1 Delivery time of control DR ... 58

5.1.2 Response time to received control DR ... 59

5.1.3 Retrieval time ... 59

5.2 Control Demand Response ... 59

5.2.1 Effect of the Number of Energy Managers on the Delivery of a Signal ... 60

5.2.2 Effect of the Number of Energy Managers on the Control DR Response Time for a Single HAN ... 62

5.2.3 Effect of the Number of Energy Managers on the Time for the Total Consumption to Drop 64

5.3 Time-of-Use ... 65

5.4 Feedback ... 66

5.4.1 Notifications ... 67

5.4.2 End-User remote consumption monitoring and appliance control ... 68

5.5 Connectivity & Reliability ... 69

5.6 Heterogeneity & Flexibility ... 69

5.7 Chapter Discussion ... 69

6. Conclusions & Recommendations ... 71

6.1 Summary ... 71

6.2 Conclusions ... 72

6.3 Recommendations ... 73

References ... 75

Appendix A 80

OpenMTC Resource Structure 80

List of figures

Figure 1-1: Middleware Structure [7] 2

Figure 2-1: Demand Response Classifications [20] 9

Figure 2-2: DR Signal flow between Utility Provider and Residential Home in a Smart Grid Network [20] 9

Figure 2-3: Smart Home Energy Management System Overview 11

Figure 2-4: General Middleware Model [30] 16

Figure 2-5: Context-Aware Smart Home Middleware Architecture [33] 19

Figure 2-6: System Architecture of Community Based Management [39] 21

Figure 3-1: Overview of a SHEMS 27

Figure 3-2: Functional System Architecture 36

Figure 4-1: Architecture of the SHEMS 39

Figure 4-2: Modular diagram for utility provider application 40

Figure 4-3: Sequence diagram for grid_monitor 41

Figure 4-4: Activity diagram for grid_monitor script 42

Figure 4-5: Example of notification signalling 44

Figure 4-6: Modular diagram for end-user's applications 44

Figure 4-7: Sequence diagram for feedback script 45

Figure 4-8: Feedback script modules 46

Figure 4-9: Sequence diagram for home_server 47

Figure 4-10: Activity diagram for home_server script 48

Figure 4-11: IWP sequence diagram 49

Figure 4-12: Home Area Network 49

Figure 4-13: Connection arrangement of load controller 51

Figure 4-14: Control DR sequence diagram 52

Figure 4-15: ToU retrieval sequence diagram 53

Figure 4-16: Modular diagram for Device Emulator application 54

Figure 4-17: Experimental Testbed Setup 56

Figure 5-1: DR Delivery Latency for a one HAN 61

Figure 5-2: Time for an Energy Manager to Respond to control DR 63

Figure 5-3: Time for Total Consumption to Drop 65

Figure 5-4: ToU Latency with Increasing Homes 66

Figure 5-5: Email and SMS notification screenshots 67

Figure 5-6: Freeboard web dashboard 68

Figure 0-1: OpenMTC server resource tree 80

Figure 0-2: OpenMTC gateway resource tree 81

List of tables

Table 3-1: Energy Management Services 32

Table 3-2: Control DR Signal Content Requirements 33

Table 3-3: ToU Signal Content Requirements 34

Table 5-1: Effect of control DR on appliances 60

Table 5-2: Control DR Latencies 60

Table 5-3: Delivery times for notifications 68

Table 0-1: Examples of URLs used to get and retrieve data from OpenMTC server and OpenMTC gateway 82

List of examples

Example 4-1: HTTP Get to retrieve total data from server container 41

Example 4-2: ToU (1) and Control DR (2) 42

Example 4-3: HTTP Post to add ToU to server container 43

Abbreviations

API	Application Programming Interface
DR	Demand Response
DSN	Data Streaming Network
EMC	Energy Management Controller
EMS	Energy Management System
ETSI	European Telecommunications Standards Institute
ETSI TC	European Telecommunications Standards Institute Technical Committee
FHEM	Freundliche Hausautomantion und Energie-Messung
GB	Gigabyte
HAN	Home Area Network
HEM	Home Energy Management
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure-as-a-Service
ICT	Information and Communication Technology
IoT	Internet-of-Things
IP	Internet Protocol
ISP	Internet Service Provider
IWP	Interworking Proxy
JSON	JavaScript Object Notation
R/kWh	Rands per kilowatt-hour
M2M	Machine-to-machine Communication
PaaS	Platform-as-a-Service
PLC	Power-line Communication
QoS	Quality-of-service

RAM	Random-Access Memory
REST	Representational State Transfer
RF	Radio Frequency
SHEMS	Smart Home Energy Management System
SLA	Service Level Agreement
SMS	Short Message Service
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TOT	The Open Transporter
TOT_IWP	The Open Transporter Interworking Proxy
ToU	Time-of-use
URL	Uniform Resource Locator
USB	Universal Serial Bus
VM	Virtual Machine
W	Watts
WAN	Wide Area Network

1. Introduction

The ever-increasing demand for electricity makes it imperative for utility providers to generate more energy and improve the efficiency of current electrical systems. Failing to meet this increasing energy demand will run the risk of a potential energy crisis that will eventually lead to grid blackouts. For example, Southern African countries like Zambia, Zimbabwe and Botswana have an energy deficit as their current infrastructure cannot support the high demand for electricity [1], forcing utility providers in these countries to resort to load shedding in an attempt to minimize the stress on the grid.

Furthermore, the demand on the grid varies between periods of low demand and periods of high demand commonly referred to as off-peak and peak periods respectively. The peak load requires utility providers to operate their equipment at a much higher capacity than usual to meet the increased demand. This increase in demand puts significant stress on the grid and it is because of this that peak-load reduction has been seen as a key driver in the management of energy in the grid [2].

Consequently, demand response (DR) programs have been used to influence and reshape the demand profile to reduce peak load. These programs make use of various schemes, services and incentives to help peak-load reduction. For example, the utility provider may implement a dynamic pricing scheme that influences the demand by having higher electricity rates during peak periods. By doing this, the utility provider incentivises consumers to shift the majority of their consumption to off-peak periods.

DR programs are often classified as rate-based, event-based or demand reduction bid programs [3]. Utility providers use rate-based programs to influence demand by varying the price of electricity over time in an effort to motivate consumers to modify their consumption habits. Event-based programs provide consumers with rewards for complying with demand reduction requests. In these programs, the utility provider is often given some form of control over the consumer's electric loads and in exchange, the consumer is rewarded with monetary payments. Demand reduction bid programs require consumers to send the utility provider bids on the load reduction they would be willing to participate in at a given rate that they are willing to pay.

Despite offering more influence over the demand and helping utility providers better match the demand to the available supply, increased consumer participation in these programs can further improve grid efficiency [4]. One of the main challenges to increasing consumer participation is the inconvenience of actively participating in the demand response programs. Most DR programs require direct physical control of home appliances for consumers to gain any benefit from them.

As a result, smart home energy management systems have been a growing topic of interest in research [5]. These systems provide monitoring and control services to the residents of a home to assist them in managing different aspects of their home environment like energy consumption, temperature and humidity. In addition, smart home energy management systems have been suggested to leverage DR programs to offer more efficient management of a home's energy consumption [6].

1.1 Research Motivation

Despite the various benefits that smart home energy management systems offer consumers and utility providers, current solutions are often restricted due to their proprietary nature, high implementation and maintenance costs. Consequently, this has hindered the acceptance and growth of such home management systems. Therefore, it is necessary to provide solutions that are capable of adapting to the various needs of utility providers and consumers by maintaining flexibility in the devices, technology and protocols that can be integrated into the system.

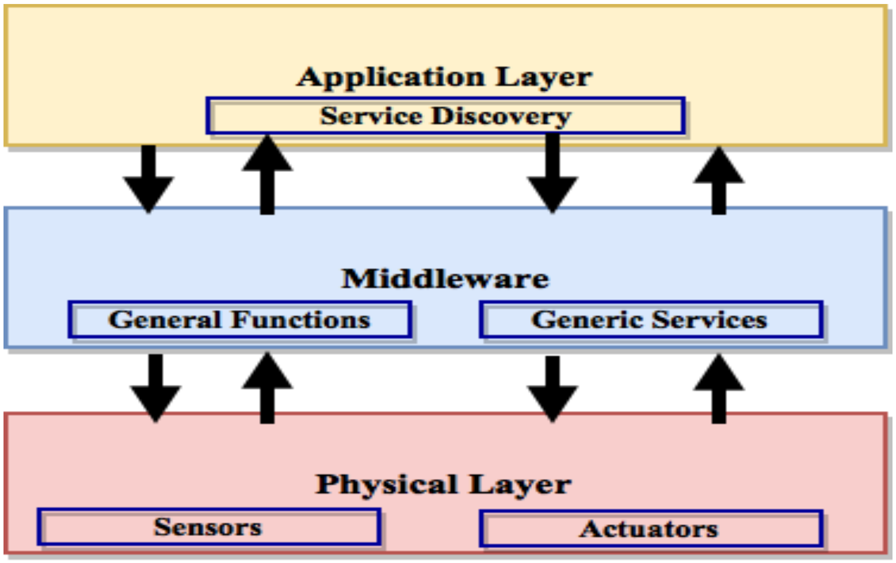


Figure 1-1: Middleware Structure [7]

The use of middleware platforms to integrate various elements of a smart home energy management system has been seen as key in supporting heterogeneous devices, protocols and technologies [7]. Figure 1-1 shows the structure of a typical middleware and illustrates how it can act as an abstraction layer between the application and the physical layers of a system. By doing this, the middleware platform enables applications to be developed across various domains without having to know the specifics of the underlying physical hardware. Furthermore, institutes such as the European Telecommunications Standards Institute (ETSI) define standards for information and communication technology (ICT) applications that address technical issues for fixed, mobile, radio and Internet technologies [8]. In addition, utility providers service consumers over a large geographical area and a management system that is implemented over such an area would have to be distributed over the different elements of the system. Therefore, making use of a standard compliant distributed middleware platform would ensure that the proposed home energy management system is able to support such a widely implemented architecture.

Virtualisation and cloud computing have been a growing topic of interest across various applications. Making use of cloud computing provides the benefit of reduction in capital and operational expenditure by reducing the need to invest in dedicated infrastructure. Furthermore, technical benefits would include energy efficiency, hardware and software optimisation, and flexibility [9]. Virtualisation provides separation between the hardware and the software and therefore makes elements such as servers, workstations, storage and other such systems independent of the physical hardware. Cloud computing, on the other hand, uses virtualisation to offer on demand services to end-users and applications. By using both virtualisation and cloud computing, the proposed system can leverage additional abstraction from hardware offered through virtualisation while benefiting from automated management and scalability of resources and services offered through cloud computing [10].

1.1.1 Problem Definition

The use of smart home energy management systems to increase consumer participation in demand response programs will require optimal management of computing resources. By leveraging cloud computing and virtualisation, smart home energy management systems can adapt to the varying demands of utility providers and consumers. In addition, the heterogeneity of such systems needs to be maintained to ensure their success and acceptance.

The energy management services to be considered in this dissertation are:

- Control Demand Response – with this service, the utility provider generates a control signal to control specific high consumption appliances in homes with energy management systems. The main aim of this service is to offer utility providers the ability to control the demand without completely disconnecting consumers from the grid. This service allows the utility provider to directly influence the demand on the grid and make load reductions without completely disconnecting homes from the grid.

- Time-of-Use (ToU) – similar to the control demand response service, the utility provider generates the ToU signal and transmits it to the home energy management system. Thereafter, the system can either make autonomous appliance operation decisions based on the received signal or simply inform the consumer of the received signal.

With the use of the aforementioned services and smart home energy management systems, utility providers are able to more efficiently affect the demand experienced on the grid. Furthermore, the ToU service encourages the consumer to move their high consumption activities to off-peak periods and the control demand response service eliminates the need for the consumer to physically take action on appliances. In addition, using these services, utility providers are able to directly influence the demand and gain the required load reduction more effectively.

With home energy management systems implemented over large geographical areas, the network requirements need to be clearly defined to ensure latencies are kept to a minimum. Current solutions are often focused on consumption management and the delivery of services to a single home [11], [12], [13], [14]. A growth in the number of homes subscribed to the utility provider's energy management service may cause inefficiencies in service delivery. However, it is important to guarantee consumers efficient service delivery regardless of the number of subscribed consumers [15], [3]. In addition, the heterogeneous nature of these systems should guarantee flexibility in the type of devices, technology and protocols that can be integrated into the system. Furthermore, the optimal use of available resources can be achieved with the use of virtualisation techniques and cloud computing.

1.1.2 Research Questions

The main research questions investigated can be summarised as follows:

- With the diversity in devices, technologies and protocols, are current smart home energy management solutions sufficient for meeting the varying needs of consumers?
- Can a smart home energy management framework be developed to support various demand response programs and their services by leveraging a

standardised distributed middleware platform, cloud computing and virtualisation?

- With utility providers making use of demand response programs and smart home energy management systems, how will an increase in subscribed consumers affect the effectiveness of the utility provider's implemented services?

1.2 Dissertation Objectives

The success of home energy management systems is dependent on its ability to adapt to the various needs of consumers and accommodate various devices and technologies. Furthermore, the utility providers' services must be delivered in a timely manner to ensure consumer satisfaction, increase consumer participation and achieve better synchronisation between the supply and the demand.

The first objective of this dissertation is to present a background on demand response programs and the various services that can be implemented with these programs. In addition, available literature on related works and current solutions are discussed to form a basis for the proposed smart home energy management system. The implementation strategies of current solutions are investigated with a focus on their heterogeneity and scalability to meet varying consumer requirements.

Secondly, based on the reviewed solutions and their implementation strategies, a smart home energy management system that makes use of cloud computing, virtualisation and a distributed standardised machine-to-machine (M2M) middleware to deliver energy management services to consumers will be presented. It will be designed and implemented to address heterogeneity and scalability of the service delivery applications to cope with the varying requirements of consumers.

Finally, the dissertation presents an evaluation on the performance of the implemented energy management services and smart home energy management system. The results of the evaluation will be presented and discussed to understand the proposed architecture's effectiveness in supporting the defined services.

1.3 Scope and Limitations

The scope of this study is limited to the control demand response service and the ToU service. Due to privacy concerns [16], this work assumes that there is a service-level agreement between the consumer and the utility provider that allows the utility provider to remotely control and monitor certain appliances in the consumer's home. Furthermore, the security concerns that arise from highly interconnected infrastructures such as smart homes, smart grids and smart cities are briefly discussed but are not the main focus of the implemented system. The security and privacy of personal information, such as location and consumption information, are issues that are key in smart grids and smart cities.

The scope of this dissertation also does not include the integration of technologies such as smart meters and renewable energy into an existing grid system. However, the proposed design will accommodate various devices, protocols and technologies to meet the set requirements. These devices and technologies will be selected based on the requirements set, their availability and their cost.

To provide connectivity between the utility provider, the consumers and the home energy management system, it is assumed that a network provider provides the available infrastructure that meets the requirements of the system. Furthermore, the network provider and the technology they use to implement their infrastructure are not a focus of this work.

In reality, a utility provider services thousands of consumers. This would mean that a single utility provider must communicate with thousands of home energy management systems and deliver services effectively to each one. However, the number of homes that could be represented in the implementation of the proposed system was limited by the available hardware. As a result, a minimal number of homes and appliances are considered to demonstrate a proof of concept.

1.4 Dissertation Outline

The rest of this document is structured as follows.

Chapter 2 discusses the relevant literature and related work. The motivation behind the interest in peak-load reduction and demand response programs is described. Consequently, the smart grid and the requirements for its applications are presented as a possible solution for peak-load reduction and more efficient grid management. The chapter introduces smart home energy management systems proposed by other researchers and how they can leverage virtualisation and cloud computing.

Chapter 3 focuses on the design requirements of the energy management services and the home energy management system. The chapter defines the utility providers service requirements and sets out communication network requirements. Subsequently, the chapter describes various elements of the proposed system and concludes with a chapter summary.

Chapter 4 presents the design and implementation of the proposed energy management system and the evaluation framework. The functional system diagram is presented and the various elements of the system are described in detail. In addition, signal flow diagrams are presented to better illustrate the interaction between the different elements of the system. Furthermore, the implemented services and the prototype are clearly described to ensure that the work can be extended.

Chapter 5 presents the detailed results of the proposed smart home energy management system, focusing on the delivery of the defined services. Furthermore, the effectiveness of the services was analysed for both the utility provider and the consumer.

Chapter 6 presents the conclusions from the work, highlighting the key aspects of this project and reviews whether or not the purpose of the study was achieved based on the set objectives and requirements. The chapter then concludes by presenting recommendations for any future work that could further improve the delivery of services to consumers using home energy management systems.

2. Background and Related Work

The previous chapter discussed the challenge that utility providers face in matching the energy demand to their generation capacity and introduced demand response programs used with smart home energy management systems as a possible solution. This chapter begins by providing more detail on the services provided through demand response programs and then moves on to discuss a general smart home architecture and its main components.

Current smart home energy management system solutions are also discussed to provide the necessary background for the proposed solution. Furthermore, the solutions presented are discussed with a focus on the services they provide to improve consumer participation in demand response programs. Finally, the use of middleware and cloud computing to further improve the heterogeneity and adaptability of smart home systems is discussed.

2.1 Demand Response

As previously highlighted, utility providers can make use of demand response programs, or services, to influence the demand being experienced on the grid. These programs have been defined as methods used by utility providers to dynamically adjust the demand on the grid to match the utility provider's generation capacity [17]. According to Chai et. al. [18], the management of the grid with the use of demand response is key to increasing grid efficiency and reducing generation costs. Furthermore, Nolan et. al. [19] highlight the convenience of acting on demand response programs as a feature that would determine the success of these programs. For example, if a consumer is inconvenienced in any way due to demand response signals, it is highly detrimental to its acceptance and would lead to a poor relationship between the utility provider and the consumer.

Demand response programs are often defined in two ways, namely incentive-based programs and time-based programs [20], [21]. Figure 2-1 illustrates the breakdown of demand response programs and the common ways in which utility providers implement them. Incentive-based programs are defined as those that involve voluntary participation of consumers by allowing the utility provider, or grid aggregator, direct control of certain high consumption appliances.

This allows the utility provider to selectively turn off appliances in the home during peak times or in cases of emergency. In exchange, consumers are often offered some form of incentive or rebate for participating in the incentive-based program. Time-based programs implement different electricity rates for different periods of the day. For example, the state of California in the United States implements a demand response program that implements an off-peak rate of \$0.05/kWh and a peak rate of \$0.099/kWh for large commercial customers [15]. As a result, the consumers are encouraged to move their consumption to off-peak periods to take advantage of the economic benefits.

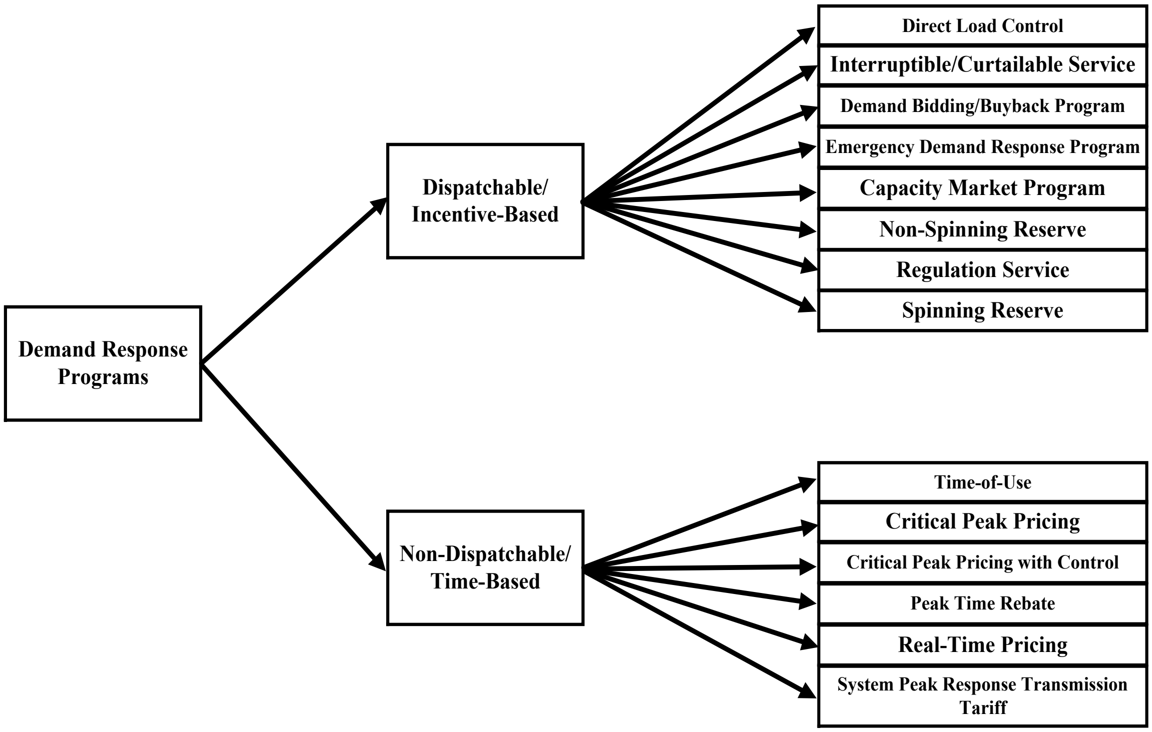


Figure 2-1: Demand Response Classifications [20]

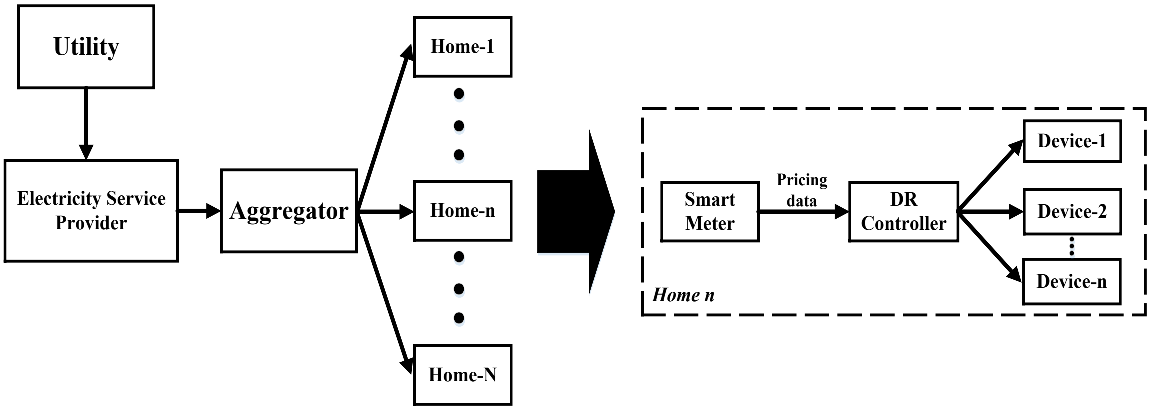


Figure 2-2: DR Signal flow between Utility Provider and Residential Home in a Smart Grid Network [20]

Figure 2-2 illustrates the flow of demand response signals from the utility provider to the residential consumer's smart home energy management system. The utility provider generates the demand response signal and forwards it to the network aggregator. It must be noted that the aggregator may be owned by the utility provider or may be a third party service provider that is responsible for the management of implemented DR programs. Thereafter, the aggregator transmits the DR signal to each home subscribed to the utility provider's services [22].

In the home, a smart meter acts as an interface and receives the demand response signal and forwards it to the home's demand response controller. Thereafter, the controller makes appliance operation decisions based on the received signal. With the described scheme, the utility provider is able to control the consumption in homes with higher accuracy and efficiency. In addition, the end-user, or consumer, can configure their preferences on the home demand response controller to ensure that they are not inconvenienced by the actions taken by their home controllers.

2.2 Smart Homes

With autonomous and remote control of appliances offered through smart home energy management systems, utility providers are able to directly influence the consumption of their customers more effectively. In addition, these home systems may have the ability to make intelligent decisions on appliance operation to achieve electricity and cost saving based on the end-users preferences [11].

A smart home can be defined as a composition of computer technology, control elements and sensing elements used within a household for various applications [23]. For example, applications such as health, security and energy management are services that can be offered in a smart home environment. In a smart grid infrastructure, demand response programs can be used together with smart home energy management systems to increase consumer participation in grid management. Furthermore, home energy management systems can offer the utility providers more real-time insight into the consumption patterns of residential households and give them the opportunity to improve their management strategies.

Demand response programs play a key role in home energy management systems [4]. Therefore, it is important to understand the manner in which DR programs leverage current home systems to increase efficiency and provide various services to consumers [6]. To make any influence on home consumption, the demand response signal must be received as input into the home energy management system, processed and then actions must be taken to achieve the required objectives. Based on this flow of events, the fundamental elements of an energy management system is discussed in this section paying close attention to the main components that have been suggested by other researchers [24], [25], [26].

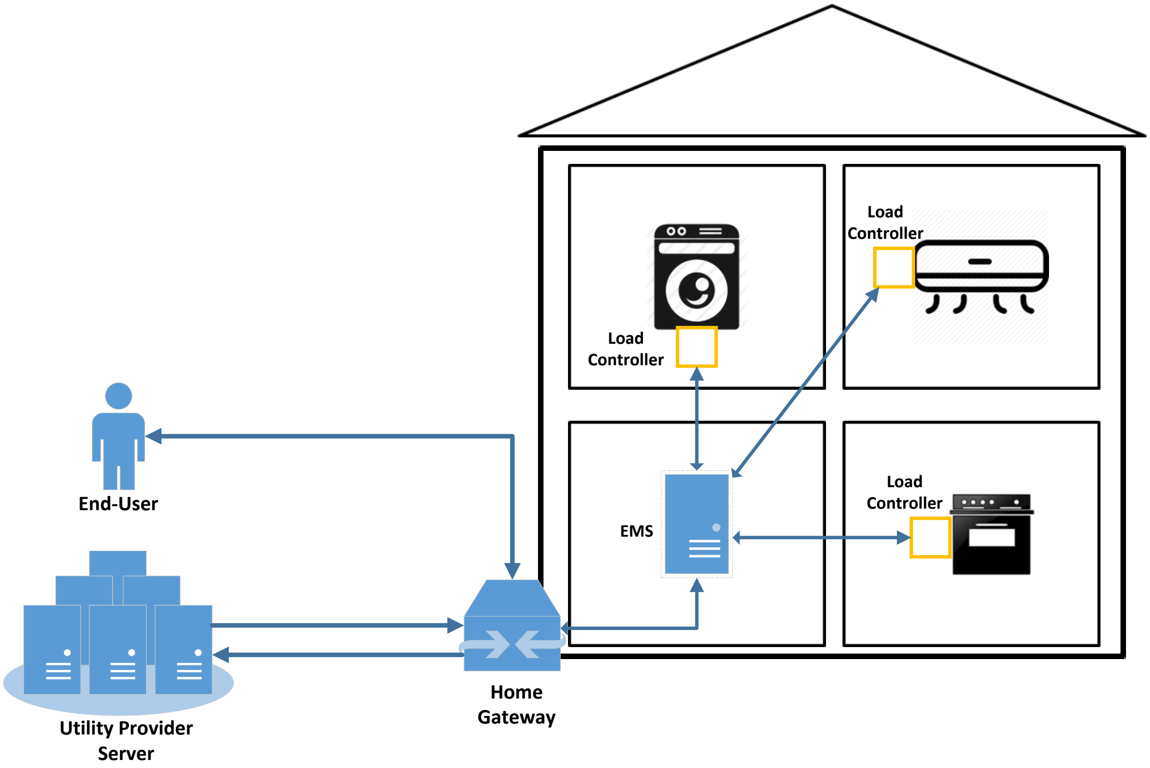


Figure 2-3: Smart Home Energy Management System Overview

Figure 2-3 illustrates the general overview of a smart home energy management system. The home gateway enables communication between the utility provider and the home energy management system. In addition, smart meters have been suggested as possible interfaces for home gateway devices as they would make implementation costs of energy management systems lower for consumers that already have the meters installed [27].

When the demand response signal is received at the home gateway, it should then be forwarded to the central processing unit, or energy management system (EMS). This EMS unit is also referred to as a Home Energy Management (HEM) unit [27] or Energy Management Controller (EMC) and processes the received signal by making appliance operation decisions based on the received signal and the preferences of the consumer.

To control and monitor appliances, the EMS makes use of sensors and actuators that either have a wireless or wired connection to the central control system. In addition, the energy management system must be able to assist consumers in achieving their consumption targets using actuators to control appliances. For example, the energy management system may need to turn off a geyser to reduce the total consumption of the home based on a demand response signal.

In this case, the home energy management unit would send the appropriate control signal to the actuator to perform the required command operation and turn the geyser off. Actuators may include smart plugs and other similar devices that enable the control of appliances by an energy management unit. Furthermore, there is a need for the system to be able to monitor the consumption of each appliance in the home. Each appliance must have a sensor that monitors the consumption and state of the given appliance and transmits the collected data to the energy management unit. Some smart plugs have the ability to both control the appliance and measure the consumption of the appliance e.g. the Zuli smart plug [28] and the TP-Link HS110 smart plug [29].

To enable communication between the EMS, actuators and sensors, various communication technologies can be used. For example, power-line communication (PLC), Ethernet, Wi-Fi, ZigBee and Bluetooth are possible technologies that could be used to interconnect the different elements of the smart home [30]. With each technology, there are advantages and disadvantages that must be considered in the implementation of the system in a residential household.

Rahman et. al. [30] review a set of wireless communication technologies that could be used in the implementation of a home energy management system and discuss the benefits and flaws of three prominent communication technologies used in home energy management systems. They concluded that ZigBee is the more suitable technology for home energy management systems due to its high power efficiency, high security and low processing requirements. However, it must be noted that the communication technology to be used in any implementation of such systems is dependent on the application requirements, the limitations of the environment and budget constraints. Therefore, a home energy management system must be able to support various communication technologies to ensure that it is adaptable and flexible enough to support various environments to meet the requirements.

Home energy management systems should be easy to install with the lowest implementation cost possible to ensure their acceptance and success. Therefore, if a home has an existing PLC network, it would be a waste of resources and money not to use it. On the other hand, if the proposed technology were different from what is implemented in the home but offers greater advantages, e.g. security and higher power efficiency, then it would be beneficial to use the former.

2.3 Current Solutions

The use of information technology in the home has shown benefits for the utility provider and consumers [31], [6]. It is for this reason that researchers have designed and implemented various energy management systems to investigate ways in which society can benefit from smarter management of energy. Saha et. al. [32] present a home energy management (HEM) system that controls a set of appliances classified as high consumption appliances and tries to keep the overall consumption of the home below a required limit.

The HEM system receives external signals from the utility provider via a home gateway. These signals consist of a demand response limit on the total consumption, in kWh, for the household and the duration of the given limit in minutes. In addition, to ensure that comfort of the consumers is preserved, the HEM system also accepts input from the residents of the household on load priorities and comfort settings for selected appliances. Based on the information collected from the utility provider and the resident, the HEM system makes appliance operation decisions to meet the desired consumption requirements.

The automatic control of appliances is exclusive to the load intensive appliances such as the air conditioning unit, water heater, dryer and electric vehicle. The algorithm employed by the HEM unit ensures that load intensive appliances are controlled in the order of their particular priorities that are set by the resident. When a decision has been made by the HEM unit to control a particular appliance, a control signal is sent to the load controller, which is connected to the particular appliance. This approach is similar to the approach taken by the proposed solution in later chapters to ensure that end-users are able to influence how their home system manages appliances to preserve comfort.

Using the control demand response service with the HEM unit, the utility provider could send the control signal to the HEM unit via home's gateway. Upon reception of the signal, the HEM unit could then decide which appliances to control to meet the required consumption reduction. This ensures that the decision as to which appliances are to be controlled is under the control of the resident and their preferences. After the HEM unit makes the decision on which appliances are to be controlled, a signal containing the control command is sent to the relevant load controllers.

The load controller in turn directly turns off, or on, the appliance. In addition, the system keeps track of comfort and demand response violations. When a demand response violation takes place, i.e. the total consumption of the household exceeds the given limit, the HEM unit cuts off power to lower priority load intensive appliances. Furthermore, when a comfort violation takes place, the HEM unit turns on or off an appliance that rectifies the comfort violation. Using this approach to appliance control ensures that an end-user's comfort is maintained and inconveniences are minimal.

Regardless, taking violations into consideration ensures that both the consumption targets of the resident and their comfort preferences are maintained. In addition, the utility provider has the ability to influence consumption reduction through the automatic control of power-intensive loads with the use of demand response signals.

Despite the benefits of this solution, Saha et. al. do not address issues such as scalability, flexibility and heterogeneity of the home energy management system. The work is focused on a single service offered to the residents through the home gateway but the work makes no mention of whether the system can be adapted to accommodate different load controllers. Furthermore, the HEM system does not provide a framework to extend the functionality to include other services that the utility provider may look to implement. Therefore, the accommodation for various devices, protocols and services will be addressed by the proposed smart home energy management system. The proposed energy management system will allow for various devices, protocols and services by using a distributed middleware platform, i.e. OpenMTC. The platform supports a RESTful architecture, APIs and standardized data structures to enable fast application development and the integration of various devices and protocols.

Similar to the work done by Saha et. al., Sianaki et. al. [26] present an energy management agent that uses demand response signals from the utility provider to control appliances in the home. The energy management agent takes the cost of electricity and the resident's energy budget requirements into consideration when scheduling and operating appliances. In addition, the home energy management system proposed in this work is focused on using a scheduler to provide energy savings by scheduling appliances to operate during low-cost periods. This implementation strategy is similar to the ToU service previously described and makes use of the HEM system to make autonomous appliance operation decisions on behalf of the consumer to further increase participation in the demand response program.

In addition to the homes electrical circuit, the home energy management system includes a communication network that allows for the exchange of data between the different elements of the system. The energy management agent receives pricing information from the utility provider via the home gateway and based on the received information, the resident's preferences and energy budget, appliances are scheduled to operate at the most cost effective time.

The energy management agent consists of multiple functional blocks: the predictor system, the monitor and allocator system, the identifier system and the optimizer system. Together, these functional blocks or sub-systems work together to provide the resident with a versatile energy scheduling system [26]. In addition, the energy management system uses these sub-systems to adapt to the resident's behavioural patterns to ensure that their comfort is preserved.

The work by Sianaki et. al. is focused on the use of algorithms to schedule appliance operation but doesn't pay particular attention to the manner in which the actual system interacts with the utility provider and the consumer. Furthermore, the specific hardware and communication technology used are not discussed. Similar to the work presented by Saha et. al., Sianaki et. al. also do not address issues relating to the scalability, flexibility and heterogeneity of the system. Their main focus is on offering a single appliance control service to the consumer with no mention of accommodating expandability for additional services.

The systems presented can be used to deliver the ToU and control demand response services to consumers. The utility provider can generate signals from their control centre and send them to the smart home energy management systems. When the home's central control unit receives the signal via the home gateway, it can then decode the signal and take appropriate action according to the received service signal. In the case of the ToU service, the received signal would allow the control unit to schedule selected appliances to operate during off-peak periods based on the preferences of the consumer. In addition, the control demand response service would involve the control unit turning off a set of appliances to meet a consumption reduction limit set by the utility provider.

The discussed solutions implement strategies to deliver energy management services to smart homes but do not take into consideration the scalability of the proposed systems and neither do they discuss the ability of the solutions to accommodate other technologies and devices. Therefore, this dissertation will address the extendibility of a smart home energy management system to enable heterogeneity and will investigate the effectiveness of the implemented demand response services used in the system with an increase in the number of end-users.

2.3.1 Middleware

The heterogeneity of smart grids, smart homes and their applications makes it important to be able to integrate various protocols, communication devices and computing technology. To enable interoperability, middleware has been suggested to act as an interface between end-users, devices and also between the various devices in the home. In a distributed environment such as an electrical grid system, middleware would have to be implemented on the various elements of the system ranging from the utility provider to the consumers and their homes.

Figure 2-4 describes the general middleware model that contains an application layer, middleware layer and physical layer. The application layer is made up of the applications and end-users that provide and receive services respectively.

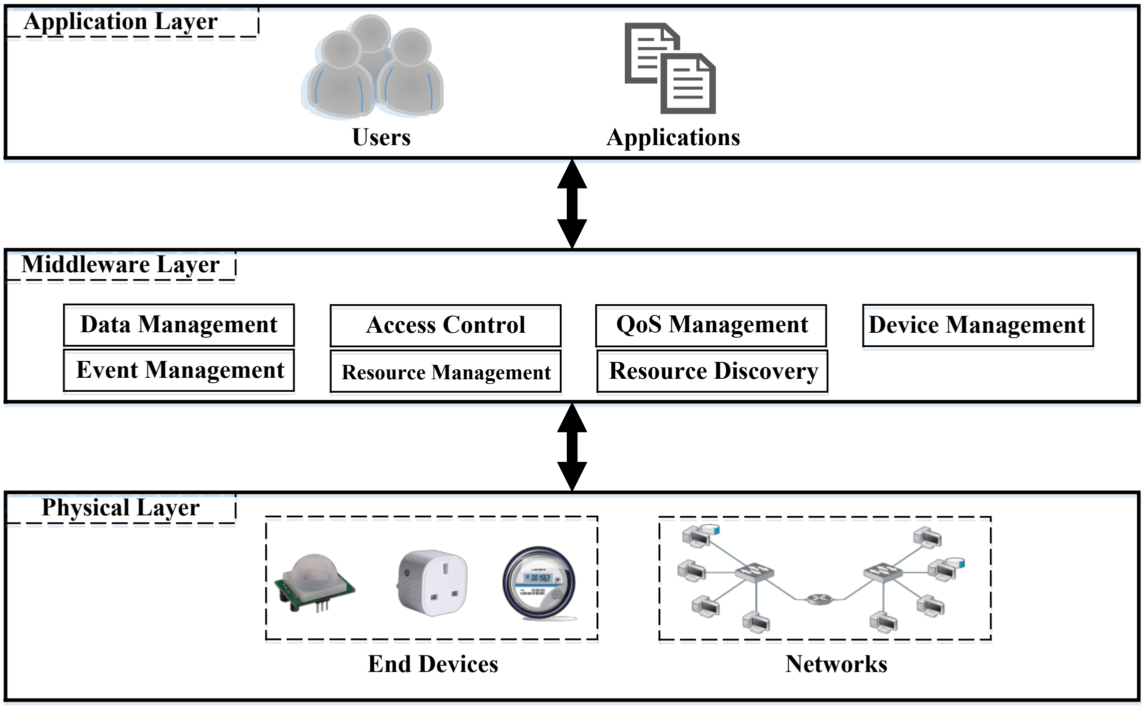


Figure 2-4: General Middleware Model [33]

The middleware layer abstracts the complexities that make up the system to enable application developers to focus on achieving their requirements and provides various services to the applications and end-users. Furthermore, the middleware provides a software layer between applications, the operating system, the network communication layers and the hardware of the system [33]. The physical layer is made up of end devices of the system that may include sensors, actuators, smart meters and network devices such as routers, switches and transmitters. Furthermore, the physical and application layers can be spread out over large geographical areas and it is for this reason that the middleware layer must be distributed across them to meet the desired requirements.

Middleware solutions can be classified differently based on their design approaches. The event-based middleware approach makes use of events as the basis of interaction between the components, applications and other smart home participants. Each event has a type and parameters associated with it. The events are generated by sending applications (producers) and transmitted to receiving applications (consumers). The event-based middleware makes use of a publish/subscribe model where subscribers are offered access to publishers' data through a database. In addition, the subscribers register for events and receive notifications when certain events occur. Event-based middleware's are appropriate for systems that involve mobility because it is possible to strongly distinguish between the subscribers and the publishers. However, the middleware does not adequately address interoperability, adaptability, timeliness and context-awareness [33]. Furthermore, low mobility and low failure applications do not benefit from event-based middleware because such applications would not generate events frequently enough to leverage the middleware functionality.

Service-oriented design approaches provide applications in the form of services. The middleware provides data management services, service management services, service discovery and quality-of-service (QoS) management [34]. Using these services, the middleware is capable of offering technology neutrality, service reusability and enable service discovery. However, the resource discovery and management do not scale due to their predefined and deterministic characteristics [33]. Therefore, service-oriented middleware is not suited for the large-scale implementation of distributed systems.

Agent-based middleware divides applications into modules to enable distribution across the network using mobile agents. The agents maintain their execution state when moving from node to node and allow for the development and deployment of decentralised systems such as smart cities and smart home energy management systems. However, the use of mobile agents can lead to unpredictability because they operate autonomously. In addition, mobile agents require significant resources to be effective and can lead to message loss if implemented in resource-constrained systems [33].

Madhusudanan et. al. [35] present a generic middleware model for a smart home that is made up of four layers: the user layer, device layer, location layer and environment layer. Firstly, the user layer consists of all the users associated with the smart home environment like the resident, utility provider, housekeepers, etc. Secondly, the device layer consists of the various devices that are part of the smart home like the appliances, sensors and actuators. Thirdly, the location layer describes the various locations in the smart home like the living room and bedroom. Lastly, the environment layer describes the various smart home environmental conditions that may include the temperature, humidity and utility rates retrieved from power and water utilities. Furthermore, these environmental conditions govern the manner in which appliances and devices in the smart home are operated in various contexts to achieve the end-users objectives. The work concludes by highlighting that with the use of middleware, end-users, devices, locations and services can be added to the smart home environment ensuring scalability and heterogeneity.

Despite taking into consideration scalability and heterogeneity of smart homes and energy management systems, the work presented by Madhusudanan et. al. only focus on the middleware aspect of the smart home. However, the work does not describe how utility providers can use the proposed system to deliver demand response services to consumers. The proposed system is focused on the architecture of the smart homes middleware and its implementation within the home.

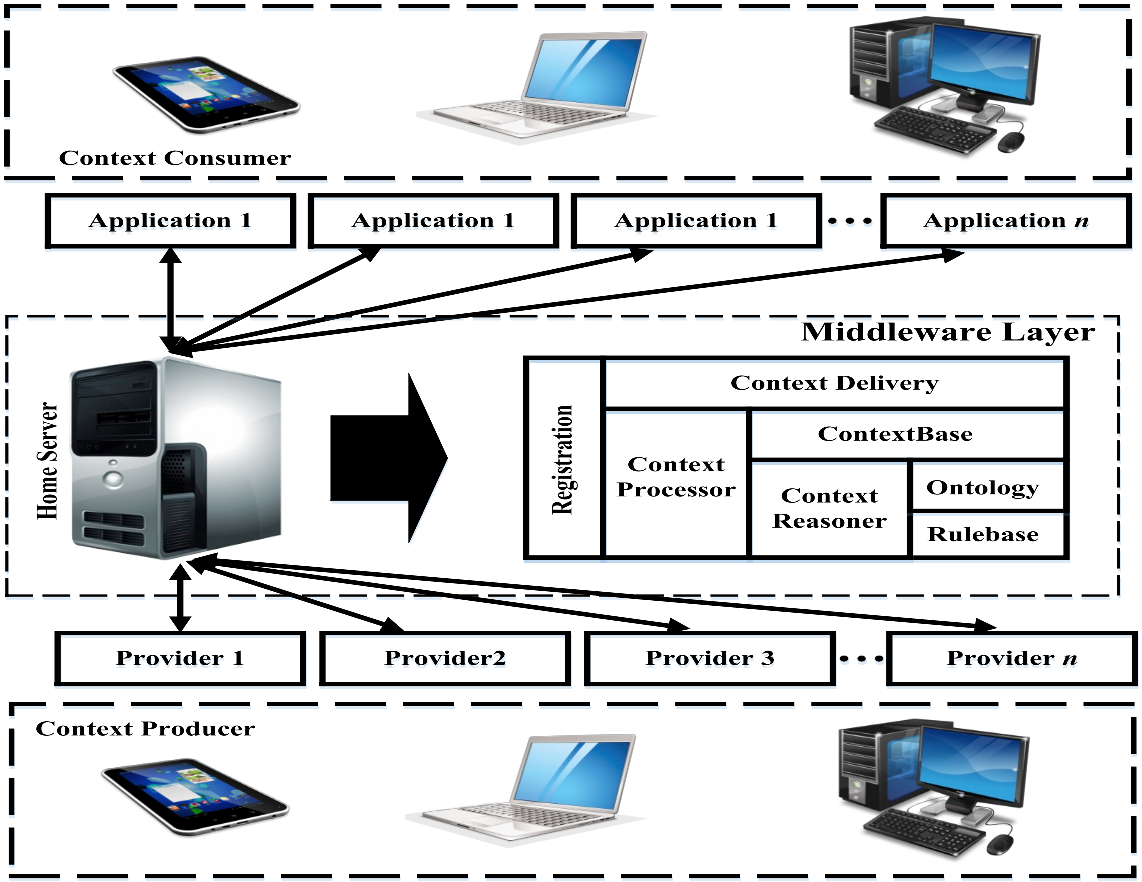


Figure 2-5: Context-Aware Smart Home Middleware Architecture [36]

Hoque et. al. [36] present a context-aware middleware that provides the necessary functionality and services to smart home applications as illustrated in Figure 2-5. The middleware presented is focused on sharing contextual information using a Java framework that is used for developing modular software solutions. Each component in the middleware is represented as a service and the overall middleware architecture is described using three layers, which are the context-providing layer, the middleware layer and the application layer. Furthermore, the middleware layer is hosted only on the smart home server, while the smart home applications can reside on any device in the smart home.

This layer is made up of a context processor, a context delivery module, a context base module, a context reasoner, an ontology module and a rule base. The providers in the architecture perform the task of collecting raw data from devices and send it to the middleware layer. Thereafter, the middleware layer processes the raw data and provides a context. The registration module provides a service that declares the capabilities of the various devices in the system. In addition, the applications access the context provider through context delivery to generate the required smart home services. With the use of this kind of middleware, it is possible to extend the functionality of the smart home by enabling the support for various devices, applications and contextual information. However, like the work presented by Madhusudanan et. al., the service provider is not taken into consideration in the implementation of the proposed system. The system is focused on providing contextual services to end-users using the middleware.

2.3.2 Cloud Computing

Due to the large number of consumers that energy utility providers have, the implementation of an energy management system must be scalable to ensure that services maintain a quality of service. The systems presented in the previous section have a physical device that hosts the energy management unit or energy management applications. However, there are limitations that come with hosting applications on a physical device in the home. For example, if the applications on the physical device outgrow the resources of the physical device, the applications need to be moved onto a device with more resources. With consumers ranging in the hundreds, and possibly thousands, it is not feasible for utility providers to redesign applications for individual end-users.

In addition to the host device resource capacity, the large-scale implementation of energy management systems requires high connectivity and availability. This is similar to the requirements of the smart grid, where there is a need of a scalable implementation model that can support the high connectivity and high volumes of data.

To meet scalability requirements, it has been suggested to leverage virtualisation and cloud computing for smart grid applications such as smart home energy management systems [37], [38], [39], [40]. Virtualisation is the separation of the software from the hardware and works together with cloud computing to provide flexible and adaptable computing resources [10]. Cloud computing is defined as a computing model that offers universal, convenient and on-demand access to computing resources e.g. networks, storage, processing power, etc. Furthermore, on-demand self-service, broad network access, resource pooling and rapid elasticity are some of the features that are offered by cloud computing. These features ensure that services offered via cloud computing are ubiquitous, scalable, secure, private and economic [37], [41].

In the general description of a smart home energy management system, devices or appliances are monitored and controlled by a central entity hosted on a physical device. The applications make use of the physical resources of the device they are hosted on. However, using cloud-computing applications can leverage the virtual resources available on the cloud platform to meet the desired requirements.

Home energy management systems make use of a physical gateway or device to act as an interface between the utility provider and the systems central management unit. In addition, the home gateway may host and run the necessary energy management applications. Virtualizing the home gateway can be considered as moving some of the functionality or applications away from the physical gateway into a virtual environment i.e. the cloud. For example, applications that make appliance operational decisions can be moved to the utility provider's cloud server and will enable the simplification of the home gateway.

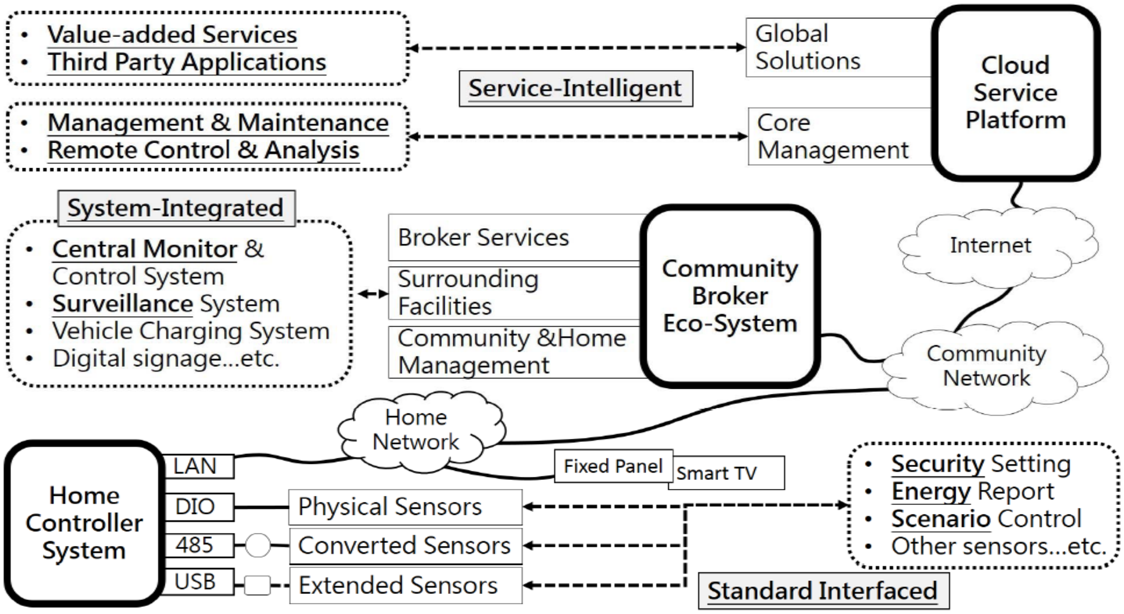


Figure 2-6: System Architecture of Community Based Management [42]

Lee et. al. [42] present a three-layer hierarchical architecture consisting of a smart home, community management system and a cloud service platform. Figure 2-6 illustrates this proposed architecture, which consists of the home domain, the community domain and the cloud domain. First, the home controller, in the home domain, is in direct control of devices in the household via various interfaces. Furthermore, the home controller connects to the community network via its own home network. Second, the community broker eco-system or community management system, in the community domain, is responsible for integrating each smart home with community-based services such as the surveillance system and vehicle charging system.

Lastly, the community management system is connected to the cloud service platform, in the cloud domain, through the Internet. On the cloud platform, global solutions and core management services are offered. The core management services comprise of a community or home monitor configuration module to enable multiple display functionality and a remote control and analysis module that enables the use of a smartphone to control and monitor devices in the home.

In addition to the control and monitoring of devices, the core management module implements HTTP-based authentication and authorization to ensure security and privacy. On the other hand, the global solutions service comprises of value-added services and third party applications. The value-added services include family applications (e.g. music and photo-based applications) and location-based applications (e.g. News and Weather applications). Third party applications may include healthcare, transport, mapping and government information applications. Unlike the other proposed systems, this is one of the few proposed systems that consider the relationship of each smart home with other homes in the system. The system does provide the ability to extend the available services through the use of value-added services and third party applications in the global solutions module.

Furthermore, the use of standard interface devices ensured that the system could include and integrate various devices to provide diverse services. However, this system does not investigate or consider the effect that other smart home's have on the quality-of-service experienced by each home. It is assumed that with the use of cloud computing, resources can be dynamically adjusted on-demand to meet the requirements of growing consumers. Nevertheless, the effectiveness of service delivery may be affected by an increase in the number of smart homes in the system. Therefore, it is important to identify the bottlenecks to ensure that they can be resolved with cloud computing. If they cannot, it is important to identify what strategies must be implemented to ensure that each home receives effective and efficient services regardless of the number of homes in the system.

Soliman et. al. [43] present a ZigBee-based smart home architecture that uses ZigBee devices to monitor and control appliances in the home. The system consists of two microcontroller boards i.e. the central transmitter and the central receiver. The central receiver is connected to the actuators in the home network and is also connected to the database server through an Internet connection. The central transmitter is connected to all the sensors in the network and is responsible for collecting information from these sensors.

This architecture uses the cloud-computing model to implement web applications that read sensor data, store the readings and allow for the remote monitoring of appliances. Furthermore, the web applications are separated into the front-end applications and the back-end applications. The front-end applications are responsible for end-user interactions while the back-end applications are responsible for the processing and storage of data. In addition, Soliman et. al. implement the use of JavaScript Object Notation (JSON) for data exchange between elements of the system. Due to its lightweight nature, JSON is ideal for applications such as the smart home because of the low processing and low power requirements of devices in the internet-of-things (IoT) and the smart home [43].

Similar to the work presented by Lee et. al., this work does not make use of an IoT-specific data management tool that applications such as smart homes can significantly benefit from. Furthermore, the design of the system is based on the use of a ZigBee network for the communication between devices in the home and the central controller and receiver. There is no mention on whether or not the system can accommodate other communication technologies. Therefore, the heterogeneity of the system is not adequately addressed as it is highly dependent on a single technology i.e. ZigBee.

Ye et. al. [44] present a cloud-based framework for a smart home. The framework consists of a smart home cloud, a smart home network, smart homes and consumers. The smart home cloud is used by a service provider to perform the necessary data storage and processing tasks that are assigned to the management of appliances in individual smart homes. Furthermore, a smart home network is defined as a collection of smart homes while a smart home itself is represented as a single-family household containing a central control system and multiple devices. The central control system is responsible for the exchange of data between devices in the home and the processing applications in the cloud.

In addition, users use web-based tools or applications on their personal devices to access services provided by the cloud-based smart home applications. However, despite taking into consideration the security of the system, this solution also does not define the data management tools used. In addition, the communication technology used and the systems ability to accommodate other technologies is not discussed. This limits the ability of the system to adapt to the varying requirements of smart homes and restricts the growth and acceptance of smart home energy management systems. The use of the OpenMTC distributed middleware platform will allow the proposed energy management system to be able to support various services, adapt to varying requirements and enable fast deployment of applications. Furthermore, the OpenMTC platform provides standardized data formats and management tools that enables applications and users to interact with the system effectively.

2.4 Chapter Discussion

This chapter defines how utility providers implement demand response programs to influence consumption and how these programs integrate into smart homes. The general smart home was presented consisting of an end-user, a utility provider, a home gateway, load controllers and appliances. Specific examples were then discussed from previous research, middleware based smart homes were defined and the ability of smart homes to leverage the benefits of cloud computing was discussed.

Secondly, middleware based smart homes were defined and presented. These solutions make use of a middleware framework to ensure heterogeneity and adaptability to the various requirements of smart homes and their residents. Furthermore, the smart homes presented that make use of middleware do not focus on specific services that can be implemented. The presented solutions provide a general framework to allow for service providers to deliver various services and accommodate various technologies and devices within the home.

Thereafter, smart home energy management systems leveraging cloud-computing technology are discussed. It is highlighted that cloud computing can provide scalable services to end-users through the use of on-demand resource management. In addition, the smart home energy management systems can benefit from the security and flexibility of cloud computing to ensure that end-users and utility providers have their data and services guarded against unauthorised and malicious access.

With the use of cloud computing and distributed middleware, energy management systems can allow utility providers to ensure that their services can be delivered effectively to all end-users. In addition, systems making use of the aforementioned combination can ensure that each end-user is able to implement their preferred or available communication technologies and devices. This ensures that implementation costs of the home energy management systems is kept to a minimum and helps ensure the success and acceptance of such systems.

Therefore, this dissertation will investigate the effectiveness of a set of demand response services implemented with a smart home energy management system. Furthermore, the proposed solution will enable the inclusion of various technologies and devices by making use of middleware. In addition, the proposed smart home energy management system will make use of cloud computing and virtualisation tools to enable the simplification of the home gateway and leverage the various benefits of cloud computing.

The following chapter will introduce requirements for the demand response services and the smart home energy management system. The chapter will also present the design of the proposed solution and discuss the various elements of the system.

3. Requirements & Functional Architecture

Chapter 2 presented the current work on smart home energy management systems (SHEMSs). In addition, the chapter presented middleware and cloud computing as technologies that smart homes could leverage to improve their flexibility to the varying requirements of utility providers and end-users.

The main aim of this dissertation is to provide an energy management solution that provides scalable and adaptable services to utility providers and end-users using SHEMSs. Therefore, it is important to consider the requirements of these stakeholders in the design of the system.

This chapter begins by defining the stakeholders of the proposed SHEMS, i.e. a utility provider, end-users and home area networks (HANs), and their roles in the proposed system. Thereafter, the stakeholder's requirements are defined to meet the required energy management objectives. Furthermore, the chapter presents the technical requirements for the SHEMS to ensure that the system is capable of providing reliable and effective services.

Due to the fact that a SHEMS would involve thousands of end-users, smart homes and devices that have various requirements and capabilities, the use of cloud computing and a distributed middleware is proposed. This will allow the system to meet the defined requirements and ensure scalability and flexibility of the SHEMS. Lastly, the energy management services are defined and the functional system architecture to meet the defined stakeholder and technical requirements is presented and discussed.

3.1 Stakeholders & their requirements

This section defines the various stakeholders that are involved in the implementation of a SHEMS. The stakeholders of the system are defined as a utility provider that provides energy management services, end-users that are recipients of these services and a smart home represented by a home area network. Furthermore, the SHEMS consists of power lines that provide electricity to each home and a wide area network (WAN) that supports communication between the stakeholders of the system. Figure 3-1 illustrates a general overview of a SHEMS based on systems presented by other researchers [45], [46].

3.1.1 *Utility Provider*

The utility provider is responsible for providing electricity and energy management services to its consumers. In addition, using SHEMS, utility providers are able to monitor and influence the energy demand on the grid more accurately. Using the WAN the utility provider can remotely monitor and control each of the smart home's consumption. Therefore, the utility provider can monitor the energy demand on its grid in near real-time and more accurately influence it to match their generation capacity.

To influence the energy demand, the utility provider may implement various DR programs that allow them to provide various energy management services to their end-users. This dissertation considers two of such programs, i.e. time-of-use (ToU) and control DR energy management services.

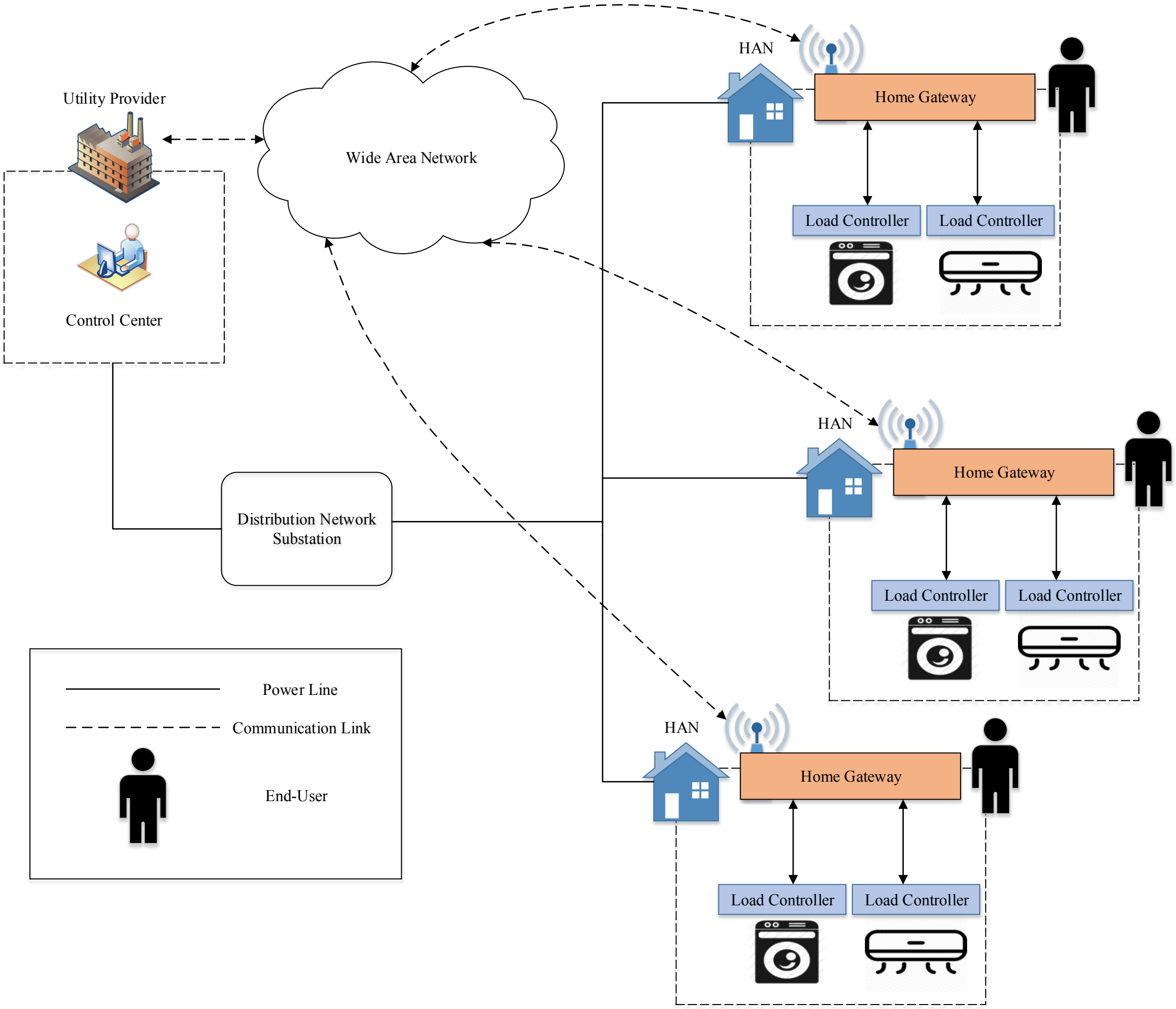


Figure 3-1: Overview of a SHEMS

The ToU service involves varying the electricity rate to influence a change in energy demand. In this service, each smart home receives the electricity rate via the home gateway and makes appliance operation decisions using the SHEMS hosted on the home gateway. Using this service, end-users can configure their smart homes to automatically control appliances when the rate falls within a given range. Therefore, using such a service would enable the end-user to benefit financially from the service and allow the utility provider to achieve the desired reduction in the energy demand.

The control DR service provides the utility provider with the ability to directly control appliances in each of the homes connected to their network. Using this service, the utility provider generates a control signal from their control centre that is then sent to a given set of smart homes to control certain high consumption appliances. The control signal stipulates a set of appliances that should be turned off based on service level agreements (SLA) with end-users. The smart homes receive the signal via the home gateway and control the required appliances using the load controllers connected to the appliances.

With the aforementioned energy management services, the utility provider's requirements can be summarised as follows:

- The utility provider must be able to control appliances in the home, either directly or indirectly, using the home energy manager. The control DR service describes the direct control of appliances based on a SLA and has the SHEMS responsible for the control of appliances. In addition, the ToU service describes the indirect control of appliances using a varying electricity rate and the SHEMS using the rate to operate appliances.
- The use of SHEMSs offers higher accuracy information on the energy demand. Therefore, the utility provider must be able to monitor the effects of their implemented services on the energy demand in near-real time to determine their effectiveness.

3.1.2 End-Users

Like the utility provider, the end-user (or consumer) should be able to directly control and monitor their home environment. However, the end-user has more control over the way the SHEMS is configured to operate their appliances. It is assumed that every HAN is owned by a single end-user that can monitor appliance-specific consumption and their total consumption via a web dashboard or in-home display that the SHEMS transmits to. In addition, the end-user receives notifications from their SHEMS on actions taken based on the reception of a ToU and control DR signal. This allows the end-user to receive specific information such as when a signal was received, when an appliance was turned off and the cost savings due to the varying ToU rate.

Furthermore, the end-user should be able to control appliances remotely and override control DR actions taken on appliances in their home. This ensures that end-users are ultimately in control of their home environment and if need be, can still operate their appliances regardless of the implemented ToU and control DR services.

To ensure that the system delivers the required services appropriately, the end-user's requirements can be summarised as follows:

- The end-user must be able to remotely control their appliances to more effectively manage their energy consumption. This enhances the relevance of SHEMS because it adds convenience to appliance operation.
- Feedback on actions taken must be offered to the end-user to ensure that they are always informed of the operations of the SHEMS. In addition, the current and historic energy consumption of each appliance must be available to the end-user. This allows the end-user to have a better understanding of how different usage patterns of their appliances can affect their energy consumption and costs. Therefore, the end-user must have on-demand access to their total and appliance-specific energy consumption data from their home gateway.

3.1.3 Home Area Network (HAN)

The home area network (HAN) consists of a home gateway, load controllers, an end-user and appliances. The home gateway is a device that serves as the first point of contact between the home and the utility provider's communication network. Furthermore, the home gateway hosts the SHEMS and directly communicates with the appliances in the home via their load controllers.

Load controllers are devices that directly connect to appliances to enable control and monitoring of the appliances they are connected to. These devices communicate with the SHEMS on the home gateway using either a wireless or wired link. In addition, the load controllers collect consumption information from the appliances they are connected to and transmit the collected information to the SHEMS.

As illustrated in Figure 3-1, the HAN consists of a home gateway, load controllers and appliances. Each load controller monitors an appliance and is required to transmit the measured data to the SHEMS at regular intervals. The interval time is dependent on the configuration of the load controllers. Therefore, the SHEMS on the home gateway must receive appliance-specific updates from the load controllers at configured intervals that ensure that the information transmitted reflects the current consumption of the appliance. This allows for the end-user and utility provider to have near real-time information on the electricity usage of the home.

3.2 Technical Requirements

Based on the SHEMS architecture presented and the stakeholder's requirements, the following technical requirements are defined in this section. Utility providers using a SHEMS to provide energy management services to end-users can leverage the benefits that cloud computing and middleware provide to enhance the sustainability and robustness of their services.

Cloud computing offers dynamic management of computing resources that would allow the utility provider to efficiently manage the resources allocated to end-users and their smart homes [37]. In addition, using cloud computing, utility providers and third party service providers can reduce the time that it takes to develop and deploy solutions and applications. This is because cloud computing separates the underlying infrastructure from the applications. Therefore, service providers are able to focus on the development of their applications, reducing the time that it takes for the applications to get to the market. In addition, smart homes could generate a significant amount of data and need the efficient storage strategies that cloud computing can provide [37].

Utility providers provide services to end-users over large geographical areas and therefore, the SHEMS would have to be distributed over this large area. Furthermore, the variety in devices, technologies and protocols must be accounted for in the implementation of the SHEMS to ensure that it is heterogeneous, flexible and cost effective. Therefore, the use of a standard-compliant distributed middleware would provide communication and data management services on a common platform for the various stakeholders, devices and applications across the entire system. Furthermore, middleware ensures that different devices and applications can exchange information to provide the required services despite not using the same protocols or technology.

Therefore, the technical requirements can be summarised as follows:

- **Heterogeneity and Interoperability** will allow for various technologies, protocols and devices to be used with a single implementation. As has been mentioned, middleware can support this and allow for each SHEMS to adapt to the various needs and requirements of the end-users. Therefore, the home gateway will have to be expandable to support a diverse range of devices. To achieve this, middleware should be used to enable the integration of various technologies and devices into a single system implementation by providing an abstraction layer between the applications and the implemented hardware. Therefore, applications and devices that use different technologies and protocols can be integrated into a single SHEMS.

- **Scalability** ensures that the utility provider is able to deliver services to each SHEMS effectively regardless of the number and the amount of data generated. This is key as utility providers will generally manage a large number of consumers and an increase in the number of consumers using management systems should not affect the effectiveness of the DR services. In addition, the SHEMS must be able to support an increased number of devices and homes. Using cloud computing can ensure that computing resources can be scaled accordingly to achieve the desired performance. As has been mentioned, with utility providers servicing thousands of homes, cloud computing can be used to support the efficient delivery of services to large numbers of end-users by providing scalable computing resources to applications such as storage and processing power.
- **Reliability** requires the SHEMS to always be able to provide services to the end-user. In addition, the information provided by the home's energy manager must be accurate and up-to-date. Furthermore, the end-user's preferences must also be preserved with every operation that requires appliance control.
- **Connectivity** allows for the HAN, the utility provider and the end-user to exchange information. Using an Internet service provider (ISP) or a dedicated network owned by the utility provider, the various stakeholders must be able to communicate effectively.
- **Flexibility** allows for the expansion of the home gateway and SHEMS to support customisation. This will ensure that applications and elements of the system such as the home gateway can be configured or adapted to the requirements of the end-user or utility provider. Using cloud computing and middleware can ensure that the system is also flexible enough to provide quick deployment of applications and support the customisation of these applications with the use of a wide variety of devices and technologies.

3.3 Energy Management Services

The proposed SHEMS will be focused on the support of three services, namely control DR, ToU and consumption monitoring. This section gives a brief discussion of these services. In addition, these services take into consideration the control and monitoring functions that must be achieved to ensure that the utility provider is able to influence the energy demand on their grid appropriately. In addition, the services discussed in this section make it possible for each home's consumption to be monitored remotely by the utility provider and the end-user.

Table 3-1 lists the defined energy management services for the proposed solution. The DR service makes use of a control DR signal that is generated by a utility provider and sent to a consumer's home energy management system to directly control appliances.

The ToU service, similar to the DR service, has a signal generated by the utility provider and transmitted to the consumer. This service implements a varying electricity rate that is transmitted through the ToU signal to the end-user and their smart home.

Table 3-1: Energy Management Services

Service	Description
Control Demand Response	Uses a control signal to influence residential consumption by remotely controlling appliances by the utility provider.
ToU	Provides a varying electricity rate to influence the demand.
Consumption Monitoring	Allows the utility provider to monitor the demand on the electricity grid in near real-time and allows the end-user to monitor their home consumption remotely.

Lastly, the consumption monitoring service enables the utility provider to monitor the demand being experienced on their electricity grid in near real-time. In addition, this service allows end-users to monitor their appliances and home consumption remotely using a mobile device or personal computer.

3.3.1 Control Demand Response

The control DR service makes use of a command signal that is sent to each home's energy management system to control individual appliances. The control signal defines which appliances will be switched off to achieve the required energy reduction.

To illustrate how a control DR signal relates to the control of appliances, Table 3-2 describes the required content of a control signal. The command field contains either an "ON" or "OFF" command for a given set of appliances that the utility provider is authorised to control. The Appliance_ID field contains a list of appliances, each described by a unique identification value, which will be controlled when the DR signal is received.

Table 3-2: Control DR Signal Content Requirements

Value	Description	Example
Command	The action to be taken on the listed appliances "on" or "off"	"Off"
Appliance_ID	The unique identification for the appliances to be affected by the demand response signal.	WM_01, DR_01

The control DR service requires action to be taken on appliances in the home when the utility provider needs a reduction in energy demand on their grid. Utility providers usually have a good understanding of the energy demand and can forecast it [47]. Therefore, the utility provider can generate a control DR signal to influence an energy demand reduction an hour before the forecasted peak. Hence, the utility provider can expect the desired reduction an hour after the transmission of the control DR signal. This means that the SHEMS must be able to receive the control signal, operate the required appliances and provide feedback to the utility provider within an hour after the transmission of the signal. The one-hour requirement is an arbitrary value that was chosen for experimental purposes.

3.3.2 Time-of-Use

As has been mentioned, the ToU service influences the consumption of each home by varying the cost of electricity. Therefore, the end-user can use their SHEMS to schedule appliances to operate when the electricity rate is below certain thresholds. For example, the end-user could configure their energy manager to switch on the washing machine when the cost of electricity falls below 0.80 R/kWh.

Table 3-3 provides details of the contents of the ToU signal. The expiration time governs the validity of each signal and ensures that home energy management systems can determine how long an end-user can benefit from a given rate.

The latency requirements of the ToU service will depend on how often a signal containing the rate of electricity will be generated by the utility provider. It will be assumed that a new ToU rate will be generated not more than once every 30 minutes. It is because of this that the SHEMS must provide the signal to the end-user at least 5 minutes after its transmission from the utility provider to ensure that the end-user can significantly benefit from every rate.

Table 3-3: ToU Signal Content Requirements

Value	Description	Example
Rate	The current rate of electricity set by the utility provider in cents/kilowatt-hour.	15c/kWh
Expiration	The time at which the transmitted rate will be invalid.	12-02-2017-12:00:00

3.3.3 Consumption Monitoring

With the ToU and control DR service offering more effective influence on the grid, it is essential that the utility provider is able to observe the effects of these implemented services. Therefore, the consumption monitoring service enables the utility provider to monitor the consumption of each household before and after the execution of a given service. In addition, the service also provides feedback on consumption details of each smart home to the utility provider and the end-user.

Due to various privacy concerns [16], the information transmitted to the utility provider must be governed by a SLA that clearly indicates what information can be shared. Therefore, it is the responsibility of the SHEMS to collect appliance-specific information from the load controllers, process it and make the appropriate information available to the utility provider. However, the end-user can have access to all the information generated within their household.

3.4 Functional System Architecture

With the stakeholder and technical requirements set out, it is now possible to describe the rest of the system architecture used to implement the proposed SHEMS. The system architecture consists of a cloud platform and a HAN as is illustrated in Figure 3-2.

Firstly, the cloud platform provides cloud-computing services to the utility provider and end-user energy management applications using the infrastructure as a service (IaaS) cloud model.

IaaS provides users of the cloud platform virtualised computing resources as a service over the Internet. Therefore, in the proposed SHEMS, the cloud platform will provide virtualised-computing resources to the utility provider and end-user's energy management applications. Furthermore, hosting energy management applications on the cloud platform allows the utility provider and third party developers to easily manage the applications and leverage the various benefits that cloud-computing provides [40]. For example, cloud computing can provide scalable processing power and storage that the applications can leverage to provide the required services efficiently.

The end-user receives the energy management services through their end-user applications on the cloud platform. These applications are responsible for making appliance operation decisions using the end-user's preferences and the DR signals received from the utility provider. In addition, these applications are also responsible for providing feedback to the end-user using the relevant feedback channels, e.g. web-based dashboard, and facilitate the control and monitoring of appliances by the end-user.

Like the end-user, the utility provider uses a set of applications using the cloud platform to provide the required energy management services. The utility provider's applications are used to generate the required DR signals and provide the utility provider with feedback on the energy demand on the grid and the effect of their implemented energy management services.

Secondly, to integrate the energy management applications, the HAN, the utility provider and the end-user, a distributed standards-compliant middleware framework, OpenMTC [48], will be used. OpenMTC is middleware framework that is aligned to the European Telecommunications Standards Institute Technical Committee (ETSI TC) specifications and provides a convergence layer for various application domains such as transport, eHealth, utilities, etc. [48]. The middleware will provide heterogeneity and interoperability by enabling the different devices and technologies to be integrated into a single SHEMS by providing an abstraction layer between the devices and the energy management applications.

Lastly, to enable the control and monitoring of appliances in the home by the end-user's energy management applications hosted on the cloud platform, the home gateway provides an interface between the applications and the devices and appliances in the HAN. The devices in the home provide appliance-specific information to the energy management applications via the home gateway and the middleware. Devices such as load controllers will be used to collect appliance-specific information and transmit it to the middleware for storage via the home gateway. In addition, control signals from the energy management applications are transmitted to the load controller devices via the home gateway.

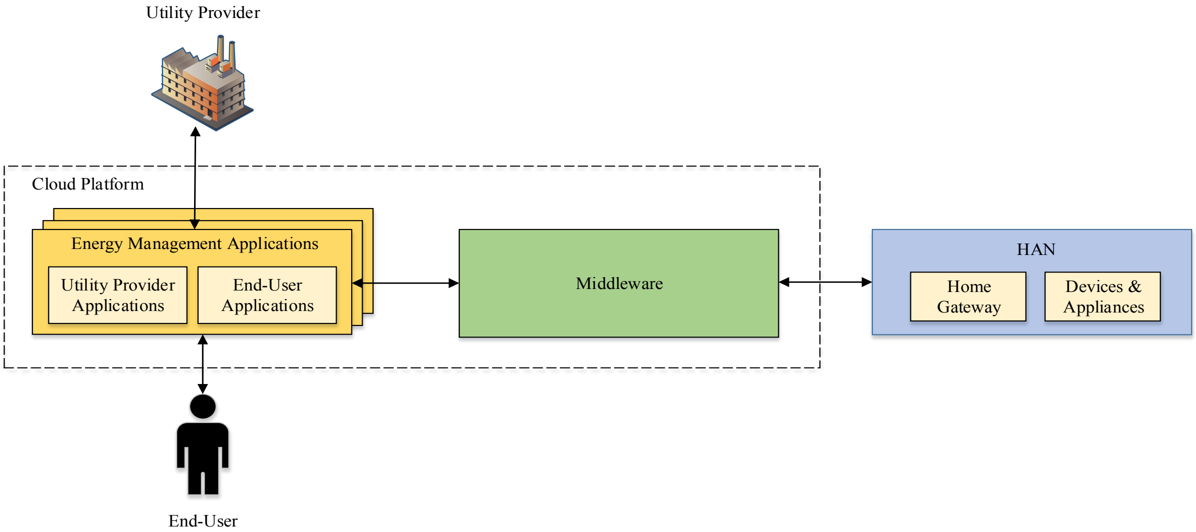


Figure 3-2: Functional System Architecture

3.5 Chapter Discussion

This chapter described the relevant stakeholders of a SHEMS and defined their various roles in the system. These stakeholders included the utility provider, end-users (or consumers) and smart homes represented by HANs. The utility provider was defined as responsible for the generation of DR signals to be used by the end-user and their HAN to manage home consumption. Thereafter, based on their defined roles in the SHEMS, the stakeholder requirements were defined. Furthermore, the technical requirements for the SHEMS were defined to support the proposed energy management services.

It was highlighted that the SHEMS must be able to provide these services to a large number of end-users effectively. Therefore, the use of cloud computing was proposed to provide virtualised computing resources to the utility provider and end-user energy management applications. In addition, middleware was suggested to provide heterogeneity and interoperability between various applications and devices.

The energy management services included the use of two DR signals, i.e. the control DR and ToU signal, which provided the utility provider with the ability to remotely influence and monitor the energy demand in the HANs. The requirements then led to a functional architecture that could meet the defined stakeholder and technical requirements to provide the required energy management services effectively.

The following chapter will provide details on the implementation of the proposed SHEMS. The implementation adopts an OpenMTC middleware platform using an OpenStack cloud platform to support the proposed energy management services. These platforms were chosen because they support the defined requirements and because of their available documentation and support. Furthermore, the testbed and hardware used to prototype and verify the functionality of the proposed SHEMS will be presented.

4. Implementation of the SHEMS and Evaluation Platform

The previous chapter presented the stakeholder's requirements, the SHEMS's technical requirements and the various energy management services. Based on the requirements and the roles of the stakeholders, the functional system architecture was presented. The architecture consisted of a cloud platform, energy management applications, middleware and a HAN.

This chapter presents the implemented SHEMS. The chapter begins by describing the SHEMS and the entities that are implemented to meet the stakeholder and system requirements. Thereafter, the chapter describes a Management Application and Energy Manager application that the utility provider and end-user use to interact with the SHEMS respectively. The HAN is then described as a home gateway, transceivers and load controllers. At the end of the chapter, the performance metrics are defined for the control DR service, the ToU service and the consumption monitoring service. These metrics are defined as the delivery time of a control DR signal, retrieval time of a ToU signal and the time it takes the utility provider to notice the required consumption drop after the transmission of a control DR signal. Finally, the Device Emulator used to investigate the effect of an increasing number of HANs on the defined performance metrics is described and the experimental setup is described.

4.1 SHEMS

The SHEMS is made up of various entities that enable the delivery and support for the control DR, ToU and consumption monitoring energy management services. Figure 4-1 shows the overview of the proposed SHEMS. The system provides services to the utility provider and end-users using energy management applications on virtual machine (VM) instances on a cloud platform i.e. OpenStack Kilo. OpenStack Kilo is the 11th release of the open source OpenStack cloud platform that can be used to build public, private and hybrid clouds [49]. By hosting applications on the cloud platform, virtual computing resources, such as processing and storage, can be leveraged by the energy management applications. Furthermore, OpenStack provides tools that allow the dynamic allocation of resources to VM instances. Therefore, it is possible to dynamically allocate resources to VM instances based on the application's demand for resources.

The utility provider, through the use of their Management Application, provides the ToU, DR and consumption monitoring services. The end-user receives these services and interacts with the

SHEMS via their Energy Manager application. By allocating each utility provider and end-user independent applications, the system ensures that they are independent and the application's functionality is reusable. For instance, with each end-user allocated an Energy Manager that provides the core monitoring and control functionality of appliances, custom functionality can be added to an Energy Manager to meet the specific requirements of each end-user.

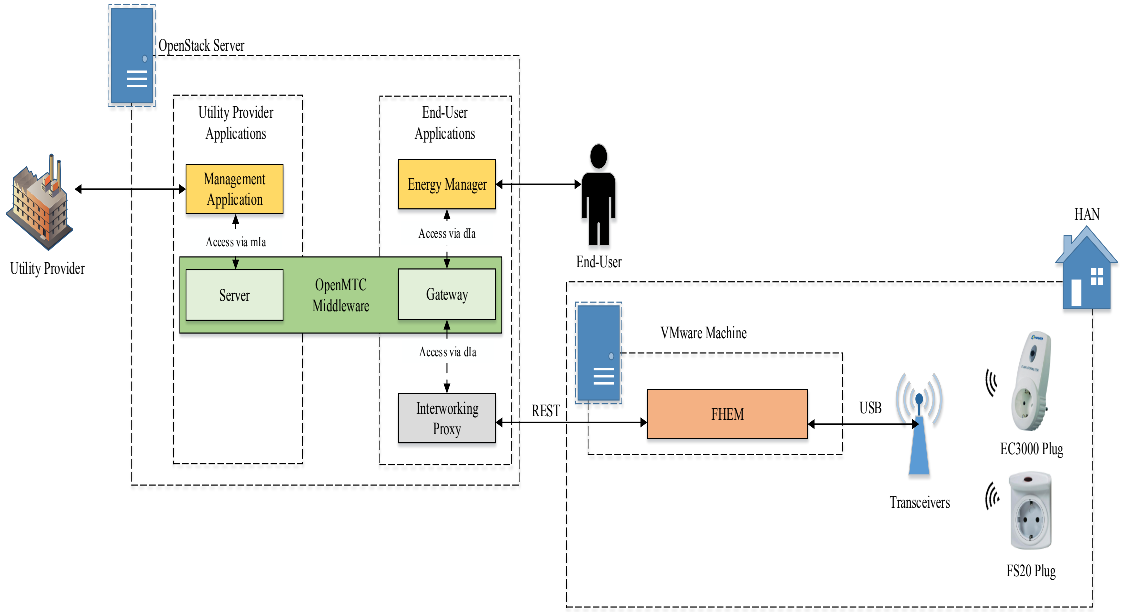


Figure 4-1: Architecture of the SHEMS

The utility provider, based on a SLA, can control and monitor end-user appliances and home energy consumption using their Management Application, while the end-user uses their Energy Manager to control and monitor their home environment or HAN. The energy management applications on the cloud platform communicate with appliances in the home via the Freundliche Hausautomantion und Energie-Messung (FHEM) automation server hosted on a VMware machine. FHEM is perl server that is commonly used for home automation. The server can be used to control and monitor devices in a home remotely via a web application or smartphone application frontend and custom scripts using the available RESTful API. As stated by the requirements in Chapter 3, the utility provider must be able to notice a drop in the energy consumption from the required HANs an hour after the transmission of the control DR signal. In addition, each Energy Manager in the system must be able to retrieve ToU data from the OpenMTC server within 5 minutes after it has been requested. By using a cloud platform, the SHEMS can leverage the virtual computing resources to ensure that the aforementioned requirements are met.

Furthermore, the proposed SHEMS must ensure that applications are able to share data to provide the required services. To enable communication between the energy management application and communication between the Energy Manager and the HAN, the OpenMTC middleware platform was used. The middleware provides a shared platform that the Energy Manager and Management Application can use to access information needed to achieve the required functionality. To provide easy access to data and the functionality offered by the OpenMTC server and OpenMTC gateways, a Python script was written to provide standard access to the OpenMTC API for the different energy management applications for the utility provider and the end-user. Further details of the OpenMTC server and OpenMTC gateway resource structure can be found in Appendix A.

4.1.1 Utility Provider Applications

Figure 4-2 shows the utility provider's Management Application that provides support for the required energy management services with the use of an OpenMTC server. The Management Application consists of multiple Python scripts that provide the utility provider with the required functionality and services. The scripts use HTTP POST and HTTP GET methods to transmit and retrieve the relevant data from the OpenMTC server.

The *grid_monitor* script provides the utility provider with the ability to monitor the electricity on the grid in near real-time. Furthermore, this script allows the utility provider to monitor the effects of a generated control DR signal or ToU signal.

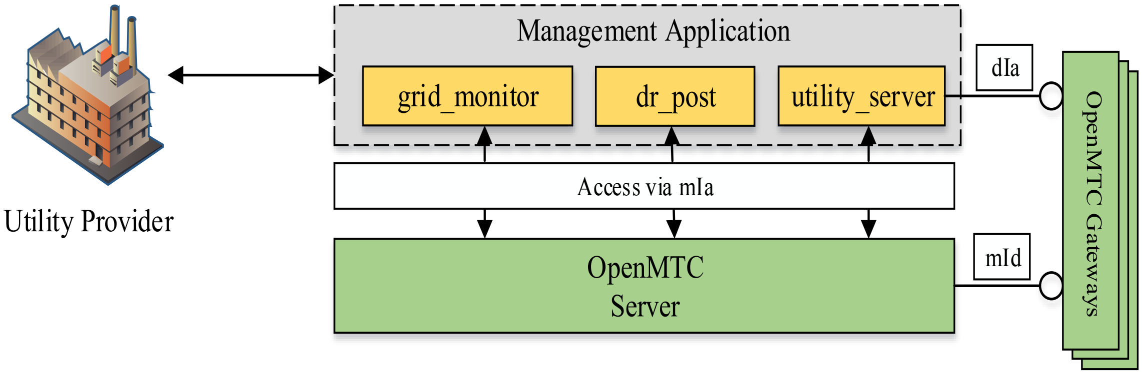


Figure 4-2: Modular diagram for utility provider application

As illustrated in Figure 4-3, the utility provider uses the *grid_monitor* script to monitor their grid by continuously retrieving each home's total consumption information from the OpenMTC server at 5-second intervals. To retrieve content from the OpenMTC server, an HTTP GET request, as illustrated in Example 4-1, is used and the server responds with a JavaScript Object Notation (JSON) object containing the requested data. Each end-user on the system has their

HAN total consumption and specific appliance consumption data stored on the utility provider's OpenMTC server.

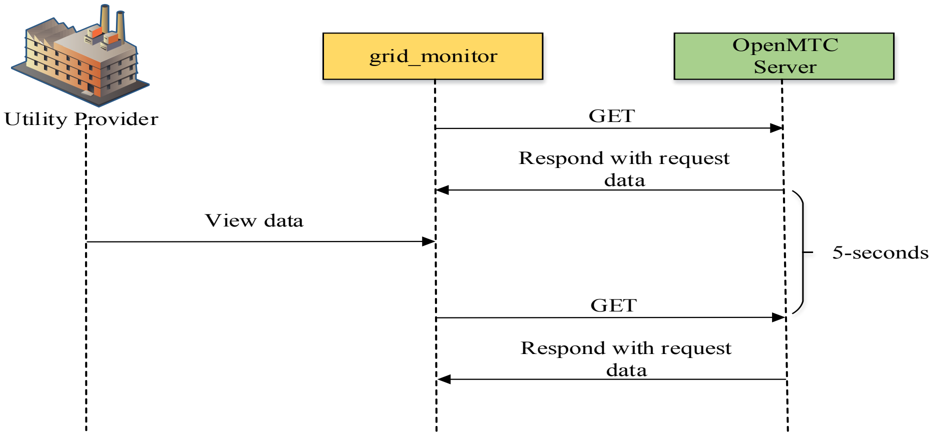


Figure 4-3: Sequence diagram for grid_monitor

```
GET /m2m/scls/Home_1/containers/totals/contentInstances
Content-Type: application/json
```

Example 4-1: HTTP Get to retrieve total data from server container

The OpenMTC server and the OpenMTC gateway implement a container structure to store data. Therefore, data such as the end-user's HAN total consumption is stored in a container on the OpenMTC server. As illustrated in Figure 4-4, the grid_monitor script continuously retrieves each home's total consumption from their containers on the OpenMTC server. After retrieving the total consumption for each home, the *grid_monitor* application sums up the totals to provide a total of the energy demand on the grid.

If the demand has exceeded a given limit, i.e. the generation capacity of the utility provider, the application provides suggestions on the homes and appliances that should be turned off to lower the energy demand. Thereafter, the utility provider can take appropriate action to meet their requirements using the control DR signals to remotely turn off high consumption appliances.

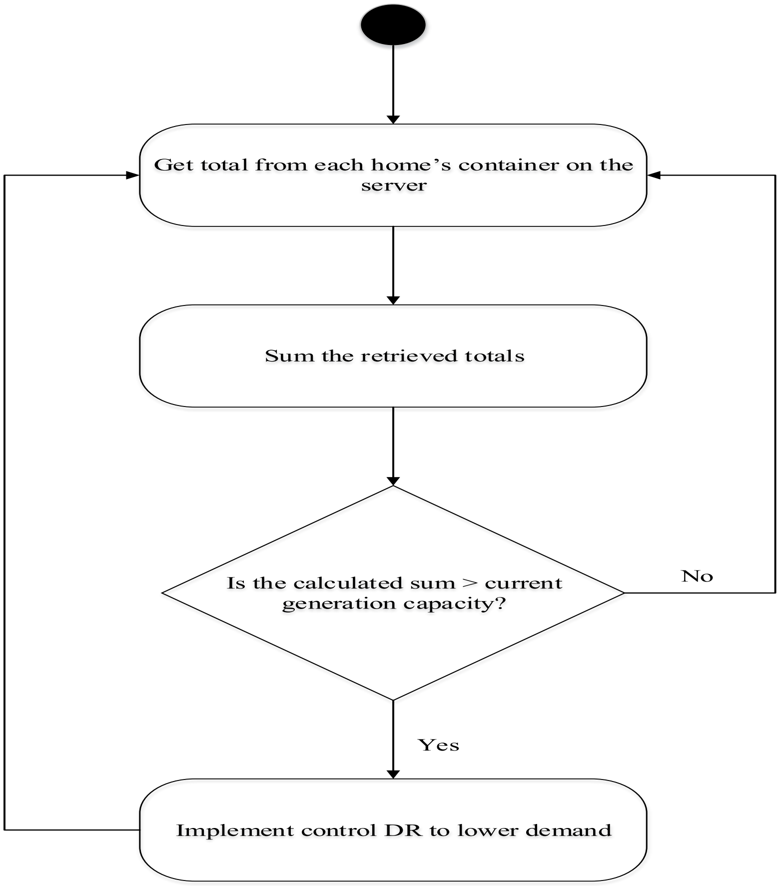


Figure 4-4: Activity diagram for grid_monitor script

The *dr_post* script enables the posting of control DR and ToU signals to the OpenMTC server containers. The script generates the ToU signal as a JSON object, shown in Example 4-2 (1), containing the current rate of electricity in cents/kilowatt-hour and how long the signal is valid for. After generation of the signal, the *dr_post* script POSTs it to the OpenMTC server.

{ 'Rate': 15, 'Expiration': '12-02-2017:12:00' } **(1)**

{ 'Command': 'off', 'Appliance_ID': {'Washing Machine': 'WM_01' } } **(2)**

Example 4-2: ToU (1) and Control DR (2)

The *dr_post* script also generates the desired control signal, illustrated in Example 4-2 (2), and forwards it to the OpenMTC server. The control signal contains the control command, i.e. "ON" or "OFF", and a list of appliances and their unique identifiers that are to be controlled.

Furthermore, these posts are executed with HTTP POST commands, such as the one illustrated in Example 4-3, with the control or ToU data transmitted as JSON objects.

```
POST /m2m/applications/DR_App/containers/ToU/contentInstances
Content-Type: application/json
```

Example 4-3: HTTP Post to add ToU to server container

The *utility_server* script monitors and manages all notifications that are generated by the OpenMTC server. When a script subscribes to a container on the OpenMTC server, it defines a contact server (running as a Python script) URL that notifications must be delivered to. Furthermore, each time a new data entry is made into the subscribed container, the OpenMTC server generates a notification containing the newly added data that is sent to the contact server URL.

Thereafter, based on the received notification, the *utility_server* script takes appropriate action. For instance, as shown in Figure 4-5, the *utility_server* script subscribes to the DR and ToU containers of the OpenMTC server defining its local URL as the contact server. Therefore, whenever a signal is posted to these containers, a notification is sent to the *utility_server* script containing the signal. Thereafter, the script forwards the received signal to the required OpenMTC gateways. Furthermore, it must be noted that any post that is made to the OpenMTC server or gateway generates a response that acknowledges the request being made.

Additionally, it must be noted that the time taken between the transmission of the control DR signal by the *dr_post* script and the *grid_monitor* script to register the desired energy consumption drop is a key performance metric of the SHEMS. In addition, the time taken for the *dr_post* script to deliver the control DR signal to each Energy Manager via the OpenMTC server and OpenMTC gateway is also key to the performance of the SHEMS.

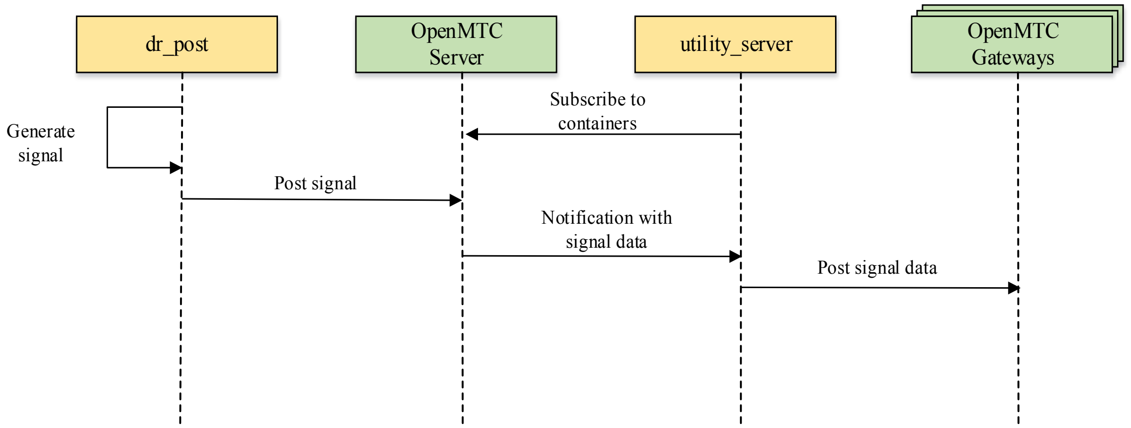


Figure 4-5: Example of notification signalling

4.1.2 End-User Applications

Similar to the utility provider's application, the end-user's application, shown in Figure 4-6, has a number of Python scripts that provide the required services to end-users. These scripts communicate with the OpenMTC gateway to provide the energy management services to the end-user.

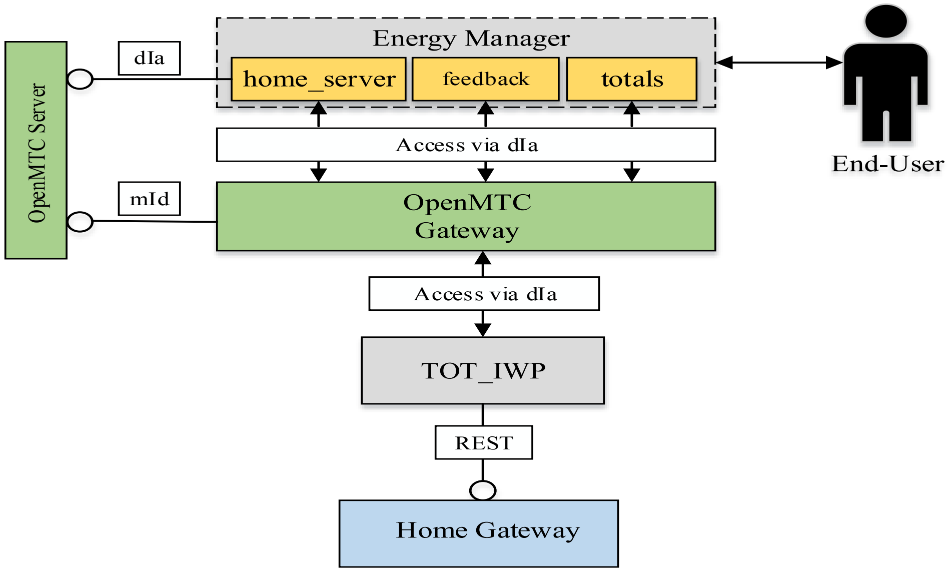


Figure 4-6: Modular diagram for end-user's applications

The *totals* script is responsible for calculating the total energy consumption of each household. It starts by retrieving the consumption (in kilowatt-hours) and power information (in watts) from each appliance's load controller containers on the OpenMTC gateway. Thereafter, the appliance's consumption and power values are summed up and posted into the *totals* container in the relevant home application resource on the OpenMTC server.

The *feedback* script makes it possible for the end-user to monitor their home remotely via a web interface and receive notifications based on their preferred notification methods. As illustrated in Figure 4-7, the script begins by retrieving the relevant appliance information from the OpenMTC gateway using the OpenMTC API script as illustrated in Figure 4-8. Thereafter, the script forwards the collected information to the relevant feedback methods configured for use in the end-user's local configuration file. For the web dashboard, the script forwards the collected information onto the pubnub [50] channel data stream. Pubnub is a global data streaming network (DSN) service that can be used to build secure real-time mobile, web and internet-of-things (IoT) applications.

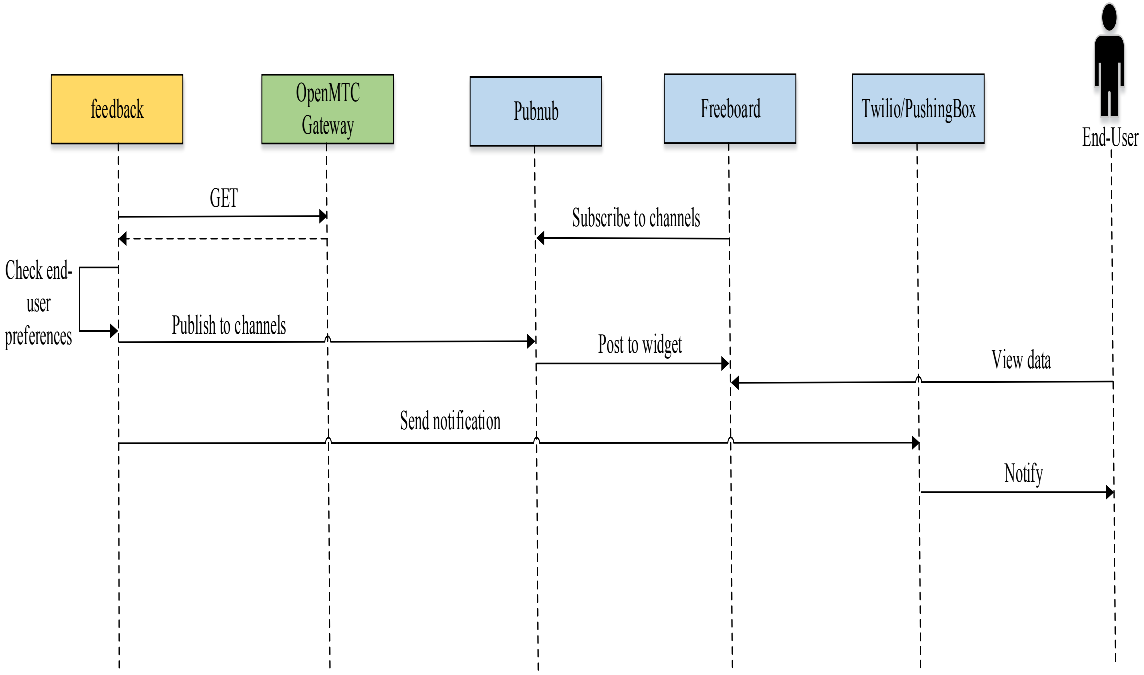


Figure 4-7: Sequence diagram for feedback script

The service uses a publish/subscribe model for data streaming and provides a messaging API that applications can use to send and receive data. The data sent by the feedback script to the web dashboard is sent as JSON objects and includes the current ToU signal and appliance-specific information, which includes each appliance's location in the home, its current consumption and power.

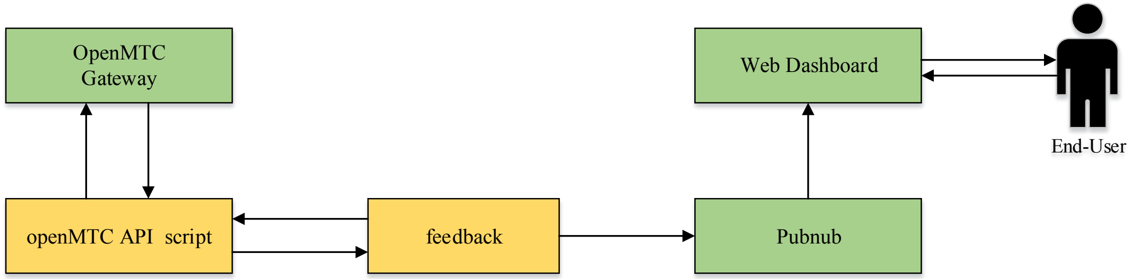


Figure 4-8: Feedback script modules

To receive messages, applications subscribe to channels. The feedback script publishes the ToU signal data and appliance-specific information on 2 different channels using the allocated publishing keys. Thereafter, the information is pushed onto the relevant dashboard widgets and made available to the end-user. The web dashboard is implemented using Freeboard [51], an open source web dashboard that can be hosted online or locally and makes use of multiple panes that contain widgets to display data. Each widget can display data from various data sources, including pubnub streams, and can display the data in multiple formats, e.g. using graphs, plain text, gauges, etc.

Furthermore, the *feedback* script also implements methods that are responsible for notifying the end-user on actions that take place in their HAN and signals that are received. The script makes use of the PushingBox [52] API service to provide the end-user with emails and Twilio [53], a cloud communication platform as a service (PaaS) that provides a short message service (SMS) API. Depending on the end-user's preferences, the *feedback* script either sends the end-user notifications using one of the aforementioned services, or both.

The end user's *home_server* script receives all the notifications that are generated by the OpenMTC gateway as illustrated in Figure 4-9. When a ToU signal and control DR signal is posted onto the OpenMTC gateway, a notification is sent to the *home_server* script containing the relevant data. To receive notifications, the *home_server* must subscribe to the relevant containers on the OpenMTC gateway. After subscription, the script can receive notifications and decode the received data to identify the type of signal and determine what action must be taken. If a ToU signal is received, the server script retrieves the end-user's appliance operation preferences from a local configuration file and determines which appliances should be turned off or on based on the current electricity rate and the retrieved preferences.

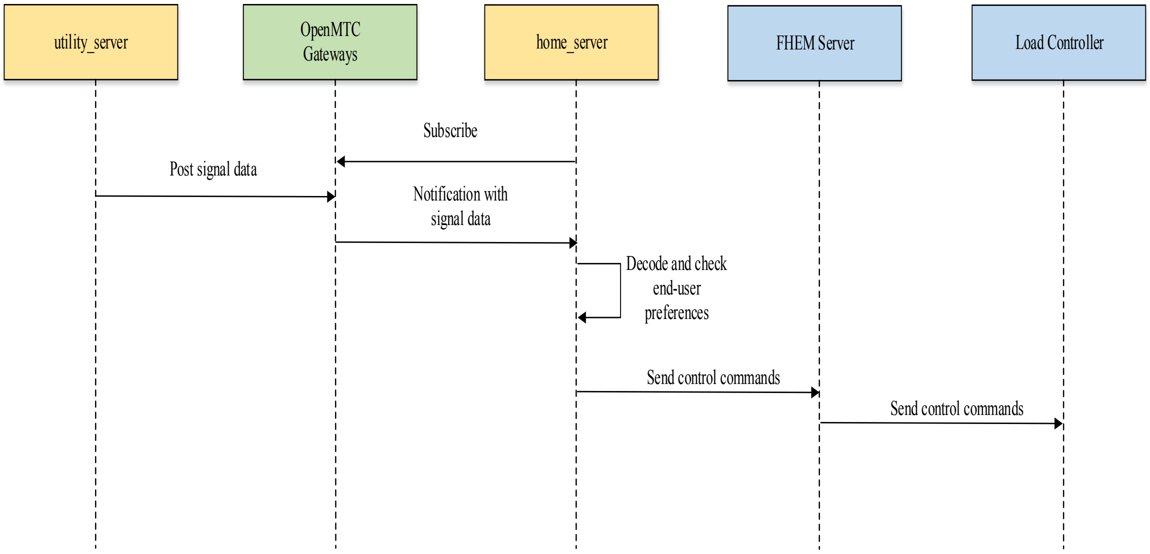


Figure 4-9: Sequence diagram for home_server

However, if it is a control DR signal, the *home_server* script decodes the signal and checks if the appliances listed in the received signal are authorised for control. If they are, the *home_server* POSTs the control signal to each of the appliance's OpenMTC gateway appliance containers. Thereafter, the interworking proxy (IWP) application controls the appliances appropriately via the automation server. If an appliance is not authorised, the utility provider is informed of the violation and only the authorised appliances are controlled. The logical operations of the *home_server* are described in Figure 4-10.

Instead of receiving notifications when a ToU signal is posted onto the OpenMTC gateway, a ToU signal can also be requested from the OpenMTC server by the Energy Manager application using the *home_server* script. In this case, the *home_server* must be able to retrieve the ToU signal from the OpenMTC server within the required time frame i.e. 5 minutes. Similarly, for the delivery of ToU signals via the use of notifications, the ToU signal must be delivered to the *home_server* 5 minutes after its transmission.

Figure 4-10: Activity diagram for home_server script

The IWP makes it possible for the home gateway and the Energy Manager application to communicate with each other via the OpenMTC gateway. The IWP application was implemented using The Open Transporter (TOT) plugin [54] that makes use of the FHEM RESTful interface to provide the automation server's functionality to applications. As illustrated in Figure 4-11, the IWP starts by collecting appliance-specific information from the FHEM server. The server then responds with the requested information collected from load controllers and thereafter, the IWP application posts this information onto the relevant containers on the OpenMTC gateway.

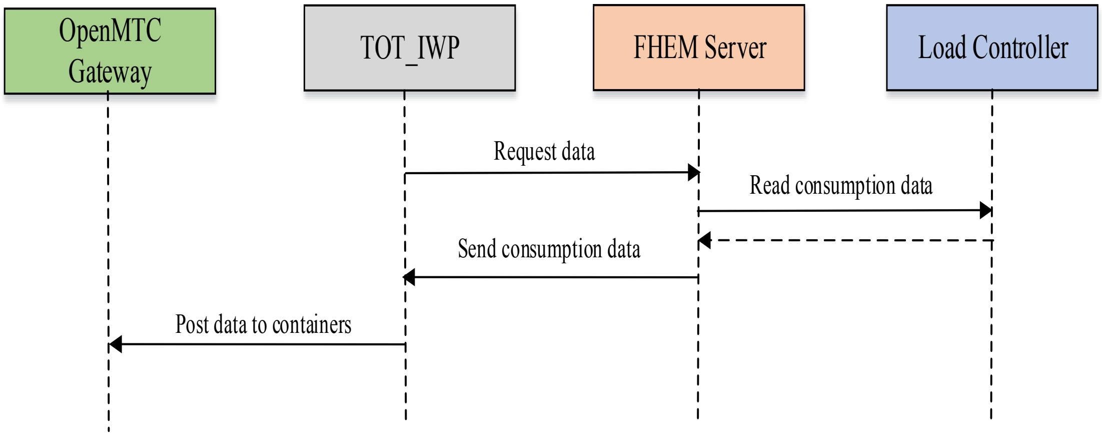


Figure 4-11: IWP sequence diagram

4.2 HAN

The HAN consists of the home gateway, load controllers and appliances, as shown in Figure 4-12. The home gateway contains the automation server that provides a RESTful interface used by the IWP application. The end-user's smart home energy management applications communicate with the home gateway, which in turn wirelessly communicates with the load controllers via a USB transceiver.

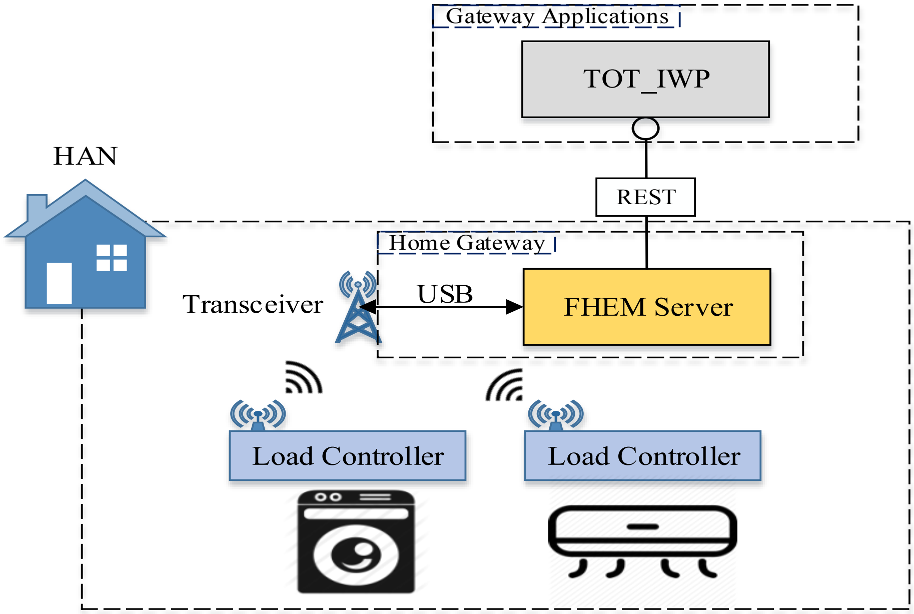


Figure 4-12: Home Area Network

4.2.1 Home Gateway

The utility provider and end-users are able to control and monitor appliances with the energy management applications via the automation server hosted on the home gateway. The automation server is implemented using FHEM because it is open source and supports a wide range of devices and protocols [55].

FHEM is a lightweight perl server used for home automation projects and enables the logging of events such as temperature, humidity and power consumption from deployed sensors [55]. Furthermore, FHEM enables the control of appliances in the home with the use of load controllers and offers a web and smartphone frontend that end-users can use to control and monitor their home environment via telnet, TCP/IP, SSL or HTTP.

In addition, the FHEM server supports multiple protocols and devices used in home automation and implements a modular approach that enables developers to add devices if they are not already supported by the server. FHEM enables developers to build applications on top of it to extend its functionality.

FHEM was launched on a virtual Ubuntu Server with 1 GB of memory (RAM) and 1 processor. Furthermore, the server communicates with the load controller devices using USB transceiver devices connected to the physical machine hosting the FHEM server. The IWP uses HTTP requests to send and receive data to and from the FHEM server. When a command needs to be sent to the server to control an appliance or data needs to be retrieved from the server, an HTTP-GET request is sent to the server. Thereafter, the server responds to the request with the appropriate action or data.

4.2.2 Load Controllers

The load controllers collect appliance-specific information and transmit the collected data to the FHEM server on the home gateway. The load controller was implemented as two separate devices, the FS20 switchable plug and an EC3000 energy measurement plug, connected together. The FS20 switchable plug enables control (i.e. “ON” and “OFF”) of appliances by receiving signals from the FHEM server using a radio frequency (RF) signal sent via a CUL USB transceiver. Furthermore, each FS20 switchable plug is uniquely addressable by a 10-character identifier and is only capable of receiving commands and provides no feedback on whether or not a command has been executed correctly.

The EC3000 energy measurement plug collects appliance-specific information such as the current and maximum power of the connected appliance in watts and its energy consumption in watt-hours. Each EC3000 plug then transmits the information it collects to the FHEM server at 5-second intervals. The EC3000 plug is uniquely addressable by a 4-character identifier.

As illustrated in Figure 4-13, it must be noted that appliances are connected into the FS20 plug, which is then connected into the EC3000 plug to ensure that energy measurement data can be transmitted even when the FS20 plug is off. If the order is reversed, any time the FS20 plug receives an "OFF" command, the EC3000 would also be turned off and no data would be transmitted from the EC3000 plug. Both the FS20 plug and the EC3000 plug have a maximum rating of 16W and together enable the control and monitoring of appliances as a load controller.

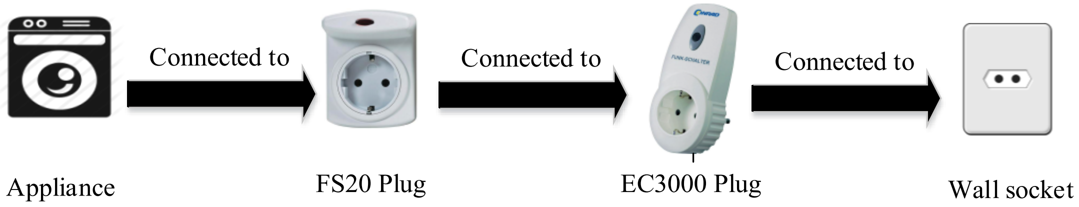


Figure 4-13: Connection arrangement of load controller

4.2.3 Transceivers

Transceivers enable communication between the FHEM server and the load controllers. The transceivers are connected to the FHEM server via USB and either receive or transmit data to and from the load controllers.

A CUL USB transceiver [56] device is used to send control signals to the FS20 switchable plug via the FHEM server. The CUL device supports both transmission and reception of signals but with the implemented devices, it only communicates with the FS20 switchable plug to enable "ON" and "OFF" operation.

Similar to the CUL USB device, a Jeelink receiver [57] communicates with FHEM server via USB. Each EC3000 plug transmits its appliance-specific data to the FHEM server via a Jeelink USB receiver. However, as has been mentioned, the plug only transmits data and therefore, the Jeelink USB device is only used to receive data.

4.3 Performance Metrics and Evaluation Tools

To verify the functionality and performance of the prototyped SHEMS and its ability to provide the described energy management services, different performance metrics were used.

4.3.1 *Energy Management Services*

With the number of end-users that utility providers may have subscribed to their services, it is important to ensure that service delivery is effective and reliable regardless of the number of subscribed end-users and HANs. Therefore, to investigate the performance of each of the defined services with an increase in the number of subscribed end-users, i.e. HANs, the time that it takes for each service to be delivered to the relevant stakeholders within the SHEMS will be investigated.

The control DR service involves the generation and transmission of a control signal to a smart home's Energy Manager application. The Energy Manager makes appliance operation decisions based on the received control DR signal. Using the IWP and the FHEM server, the Energy Manager is able to control the relevant appliances using the load controllers.

Therefore, as illustrated in Figure 4-14, the metric used to evaluate the performance of the control DR service is considered as the time taken between the transmission of the control signal and the Management Application noticing a drop in the energy consumption. Furthermore, the time taken for the Energy Manager to notice a consumption drop after its transmission of the control signal is also a metric that will be considered.

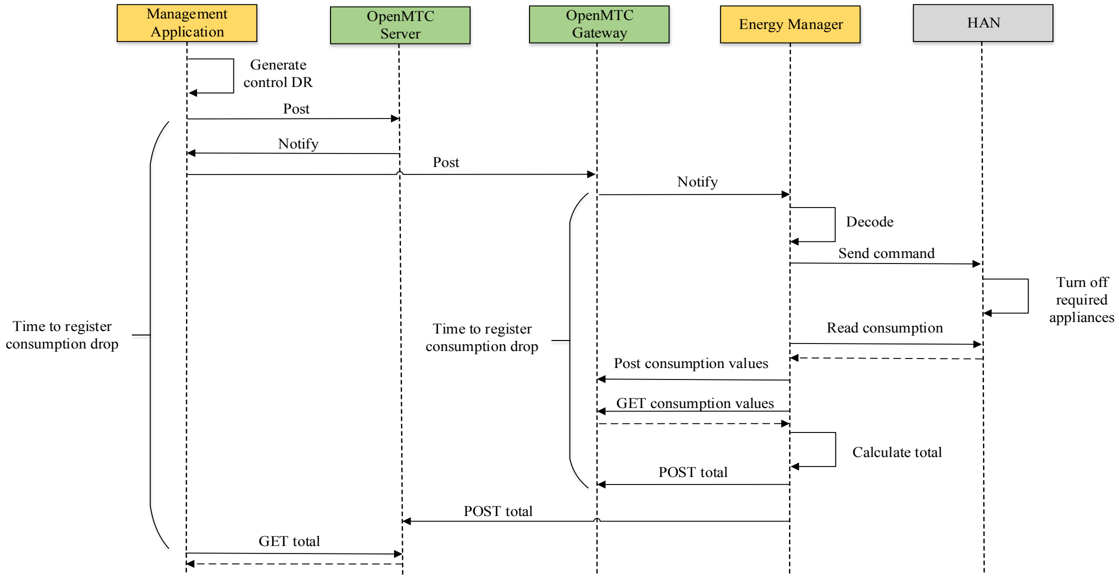


Figure 4-14: Control DR sequence diagram

The ToU service requires the utility provider to make electricity rate available to the end-user's Energy Manager application via the OpenMTC server. The Energy Manager application then makes the appropriate appliance operation decisions based on the end-user's preferences and the received rate. Therefore, as illustrated in Figure 4-15, the performance metric considered for this service is the time that it takes for an Energy Manager to retrieve a ToU signal from the OpenMTC server.

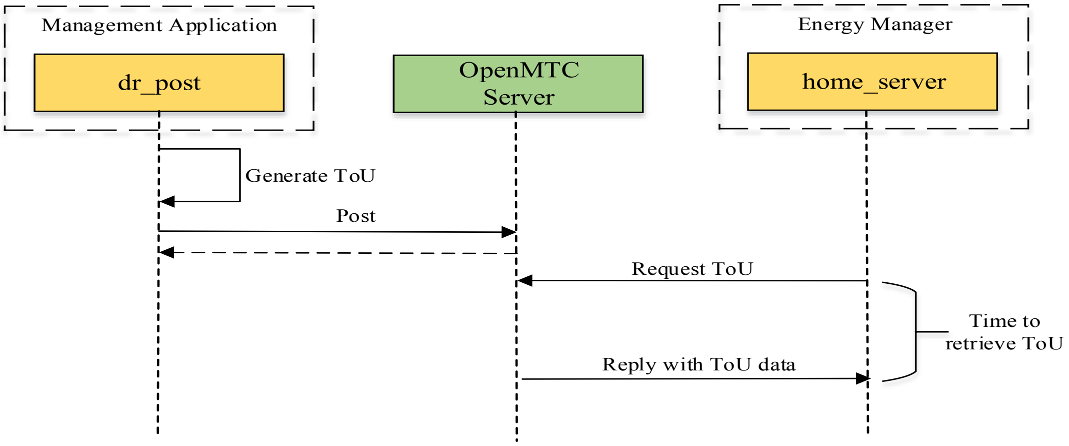


Figure 4-15: ToU retrieval sequence diagram

Lastly, the consumption monitoring service offers the utility provider feedback on the effects of their implemented services on the energy demand on the grid. The utility provider's Management Application continuously monitors the total energy demand on the grid by periodically retrieving the total consumption from each home's container on the OpenMTC server and calculating the total grid consumption. Therefore, the performance metric considered for this service is the time that it takes the utility provider to notice a change in total consumption after the generation of a control DR or ToU signal.

4.3.2 Device Emulator

Due to hardware limitations, the prototyped SHEMS is only capable of being implemented as a single home with a limited number of appliances. To investigate the effect that an increasing number of end-users, HANs and appliances has on the effectiveness of the energy management services, more dedicated hardware would have to be used to represent each HAN. However, this dissertation is focused on the functionality of the load controllers and not their performance. Therefore, a Device Emulator was developed to provide virtual appliances that could be used to emulate an infinite number of load controllers connected to virtual appliances.

Figure 4-16 illustrates the interaction between the device emulator, the OpenMTC gateway and the end-user's Energy Manager. The Device Emulator is implemented to replicate the data and functionality that the HAN provides and is also hosted on the same OpenStack VM instance as the Energy Manager application and the OpenMTC gateway. Furthermore, it must be noted that because the emulator generates virtual appliances, there is no need for an IWP. Each VM instance hosts a Device Emulator that represents the HAN environment. With multiple VM instances of the Energy Manager, OpenMTC gateway and the Device Emulator, it is possible to investigate the effect that an increasing number of HANs has on the aforementioned parameters.

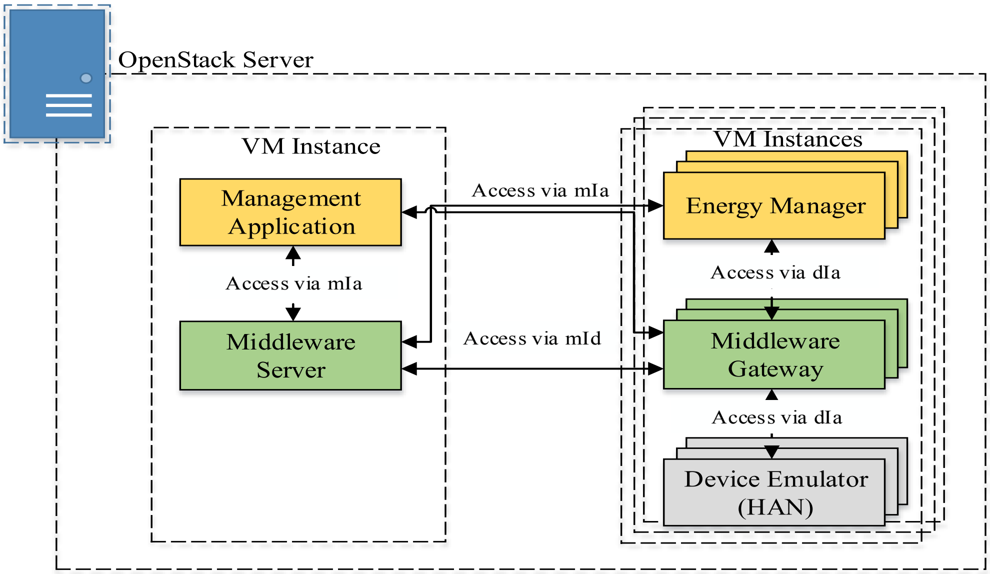


Figure 4-16: Modular diagram for Device Emulator application

The Device Emulator generates virtual load controllers that provide the same functionality as the EC3000 and FS20 physical plugs. The EC3000 virtual device generates consumption and power values based on the appliance it is configured for and transmits readings at five-second intervals. For example, if the emulator is configured to generate a load controller to monitor a washing machine, the EC3000 plug generates consumption and power values that a typical washing machine would produce.

In addition to the virtual EC3000 plug, the virtual FS20 plug is generated to control the virtual appliance. When first generated, the virtual FS20 plug is in an off state with a corresponding EC3000 plug and consumption value of zero. The Device Emulator generates virtual load controllers based on the defined appliances in its configuration file. Each virtual plug is allocated a unique identification string that maps it to an appliance. For example, a washing machine will have ‘EC3000_WM01’ and ‘FS20_WM01’ as the identification strings for the virtual EC3000 plug and FS20 plug mapped to it.

Once generated, the device emulator creates the device containers on the OpenMTC gateway and populates them with the consumption, power and state for each virtual EC3000 plug and FS20 plug. The virtual EC3000 plug is initiated with consumption and power values of zero and the virtual FS20 plug is initiated with a state value of “OFF”. Furthermore, the device emulator updates each device every 5 seconds by first checking if the state of the FS20 plug has been changed. If the state remains as “OFF”, the device emulator leaves the consumption and power values in the virtual EC3000 plug container as zero. However, if the state of the virtual FS20 plug has changed to an “ON” state, the virtual EC3000 plug containers are updated to contain power and consumption data that reflects that of an appliance that is on.

4.3.3 Experimental Setup

With the various elements of the proposed SHEMS described in detail, this section describes the infrastructure used to prototype the proposed system. The SHEMS was prototyped using VMs hosted on a VMware Workstation [58] host machine and the OpenStack server machine. VMware Workstation is a hosted hypervisor that enables users to launch and manage virtual machines on a single physical machine. The VMware host machine acted as the home gateway in the HAN and hosted the FHEM server. The OpenStack server provided the cloud environment to host the network and gateway applications used to provide the energy management services.

The utility provider and end-user’s applications were hosted on separate VM instances on the OpenStack server. This ensures that the applications are secure, isolated and easier to maintain and provision [10]. Furthermore, having applications on VMs allows for different applications that may provide different services to run on the same host. This means that a utility provider does not have to invest in new hardware to provision new services.

Figure 4-17 illustrates the experimental testbed setup used in the implementation of the proposed SHEMS. The OpenStack server and the VMware host communicate using Gigabit Ethernet connections via a layer 3 switch, while the VMware host allows the VM hosting FHEM server to communicate with load controllers via the USB transceivers. It must be noted that the FHEM server is lightweight and can be hosted on microcontroller devices such as Raspberry Pi’s, as it does not require significant computing resources.

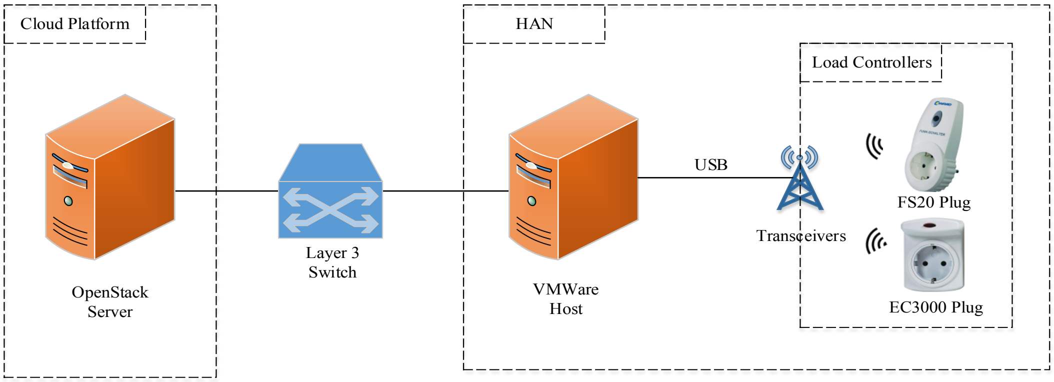


Figure 4-17: Experimental Testbed Setup

4.4 Chapter Discussions

This chapter presented the implementation of the proposed SHEMS by discussing the cloud platform and the applications used to provide the defined energy management services. It also described the HAN and its components. Furthermore, the utility provider and end-user's applications were described. These applications were hosted on a cloud platform and made use of a distributed middleware, i.e. OpenMTC, to store and manage the data. Hosting the energy management applications on a cloud platform allowed them to leverage the virtualized processing and storage resources offered by cloud computing. In addition, the application implementation followed a modular approach, which involved multiple Python scripts that served specific functions. Using this approach ensured that individual functions were easily extendable and customisable without affecting other functionality of the system.

In the HAN, the home gateway and its automation server (FHEM) were described. FHEM provides support for various devices and protocols and is easily extendable to provide support for other devices. The chapter then described the performance metrics for the control DR, ToU and consumption monitoring service. It was highlighted that the utility provider is required to receive feedback after the transmission of a control DR signal within an hour. Furthermore, it was also highlighted that the Energy Manager must be able to request and receive a ToU signal from the OpenMTC server within 5 minutes. To investigate the effect of an increasing number of Energy Managers on these requirements, the Device Emulator was introduced. The Device Emulator was described as an application that provides virtual instances of appliances that will be used to represent multiple HANs in the investigation of the effects of an increasing number of Energy Managers and HAN on the defined performance metrics. Finally, the experimental setup to illustrate the testbed environment used to prototype the SHEMS was presented. It included an OpenStack server, a VMware host representing the home gateway and load controllers.

The next chapter will present the evaluation results of the different energy management services. The functionality of the prototype will be presented for a single HAN implementation and thereafter, the effect of an increasing number of HANs on the performance of the energy management services will be presented and discussed.

5. Evaluation Results & Analysis

The previous chapter presented the implementation of the proposed SHEMS and the evaluation platform. The chapter also presented the energy management applications used by the utility provider and end-users. Finally, the performance metrics and evaluation tools were presented and discussed.

This chapter provides results and analysis for each of the DR services focusing on the described performance metrics. The chapter begins by providing a general overview of the scenarios that will be used to investigate the effectiveness of the proposed SHEMS. Each scenario describes how the performance metrics for each energy management service will be investigated. The chapter then presents results and discussions for a single and multiple HAN implementation. The single HAN implementation made use of a VMware host, load controllers and USB appliances to represent a home while the multiple HAN implementation made use of the Device Emulator to represent homes in the system. The effect of increasing the number of HANs in the system on the defined performance metrics was investigated using the Device Emulator. Finally, the chapter describes the methods used to provide the end-user with feedback on their home environment and discusses the connectivity, reliability, heterogeneity and flexibility of the prototyped SHEMS.

5.1 Experimental Scenarios

The proposed SHEMS was deployed in a testbed environment that included an OpenStack server, a VMware host, USB transceivers, the Device Emulator and load controllers. In this section, the experimental scenarios used to investigate the effectiveness of the energy management services with an increasing number of Energy Managers are described.

5.1.1 Delivery time of control DR

In this scenario, the utility provider generates and transmits a control DR signal to the end-user's Energy Manager. The Energy Manager must then check that the appliances listed in the control DR are authorised for automated control under this service and forward the control signal to the automation server, FHEM, in the HAN to control the appropriate appliances. This scenario investigates the effect of the number of Energy Managers and HANs in the system on the delivery time of the control DR signal to a single Energy Manager. The control DR signal is generated by the utility provider's Management Application and is transmitted to each Energy Manager in the system via their OpenMTC gateways as illustrated in Figure 4-14.

5.1.2 Response time to received control DR

In this scenario, the utility provider expects a drop in consumption from the affected HANs after the transmission of a control DR signal. This scenario investigates the effect of an increasing number of Energy Managers and HANs in the system on the time taken for the utility provider to register a drop in the consumption in each HAN. In addition, the end-user's Energy Manager must also be able to register the required consumption drop after the transmission of the control DR signal to the HAN. The effect of additional Energy Managers on the performance of a single Energy Manager will also be investigated. Figure 4-14 describes the times that are considered for the Management Application and Energy Manager to register an energy consumption drop after the reception of a control DR signal.

Like the previous scenario, the Device Emulator will be used to investigate the effect of an increase in the number of HANs while actual devices and the FHEM server will be used to illustrate the functionality and performance of the system with a single HAN implementation.

5.1.3 Retrieval time

In this scenario, the utility provider generates a ToU signal and makes it available for the end-user's Energy Manager via the OpenMTC server, as illustrated in Figure 4-15. Each of the end-users Energy Managers will retrieve the ToU signal from the utility provider's OpenMTC server. This scenario investigates the effect that an increasing number of end-users has on the time it takes for a single end-user's Energy Manager to retrieve data from the utility provider's OpenMTC server. The Device Emulator will be used to represent each HAN.

5.2 Control Demand Response

In the single HAN implementation, the control DR signal used in these experiments contained a command that switched off three appliances on the testbed based on the end-user's preferences. Table 5-1 illustrates the effects of the control DR signal before and after the Energy Manager transmits it to the load controllers via the FHEM server. It must be noted that USB appliances were used to illustrate the functionality of the load controllers.

Table 5-1: Effect of control DR on appliances

	Geyser	Light	Fan	Total
Before control DR	3.1 W	5.4 W	4.5 W	13 W
After control DR	0.3 W	0.3 W	0.3 W	0.9 W

The power values shown do not reflect the power consumption of the actual listed appliances but the consumption of the USB devices used to represent them. The geyser, light and fan have a consumption of 3.1W, 5.4W and 4.5W respectively before the transmission of the control DR signal. Thereafter, the Energy Manager sends an "OFF" signal to all the load controllers of the appliances and each of them are turned OFF. The power consumption received from the load controllers was not recorded as zero when off due to the fact that the load controllers themselves consumed power (0.3W).

As has been mentioned, the measured latency is between the time the control DR signal was transmitted and the time it took for the utility provider's Management Application and the end-user's Energy Manager to register the drop caused by the received control DR signal. As shown in Table 5-2, it was noted that the latency was an average of 12.82 seconds in the single HAN deployment for both the Energy Manager and the Management Application to register that the total power consumption had dropped to 0.9W. There were 10 test runs performed to obtain the average latency.

Table 5-2: Control DR Latencies

Description	Average Latency (seconds)
Energy Manager to notice the drop after reception of control DR	12.82
Management Application to notice the drop from DR	12.82

5.2.1 Effect of the Number of Energy Managers on the Delivery of a Signal

This subsection investigates the effect that an increase in the number of HANs has on the time it takes to deliver a control DR signal to each Energy Manager and HAN, as described in section

5.1.1. The time between the generation of the control DR signal by the Management Application's *dr_post* script and the reception of the signal by the Energy Manager's *home_server* script was recorded. The experiment was conducted with a control DR signal sent to a single Energy Manager. Thereafter, the number of Energy Managers was increased. It must be noted that a single client on the OpenStack server was limited to 10 active VM instances. Therefore, the experiments were conducted with 1 VM instance hosting the utility provider's Management Application and 9 VM instances hosting the Energy Managers. It was expected that with an increase in the number of end-user Energy Managers, there would be an increase in the time taken for a DR signal to be delivered due to the additional processes that would have to be executed.

Figure 5-1 illustrates the relationship between the number of Energy Managers receiving control DR signals and the time that it takes to deliver the signal to each of them. As expected, with an increase in the number of Energy Managers, the time that it takes to deliver the signal to each one of them increases.

The delivery time was measured from a single Energy Manager's *home_server* script. When there was only one Energy Manager, the delivery time was recorded as 0.073 seconds and steadily increased to a delivery time of 0.684 seconds with 9 of them in the system. The *dr_post* script retrieves a list of registered home Energy Managers from the OpenMTC server and uses a separate process for each Energy Manager to deliver the required control DR signal.

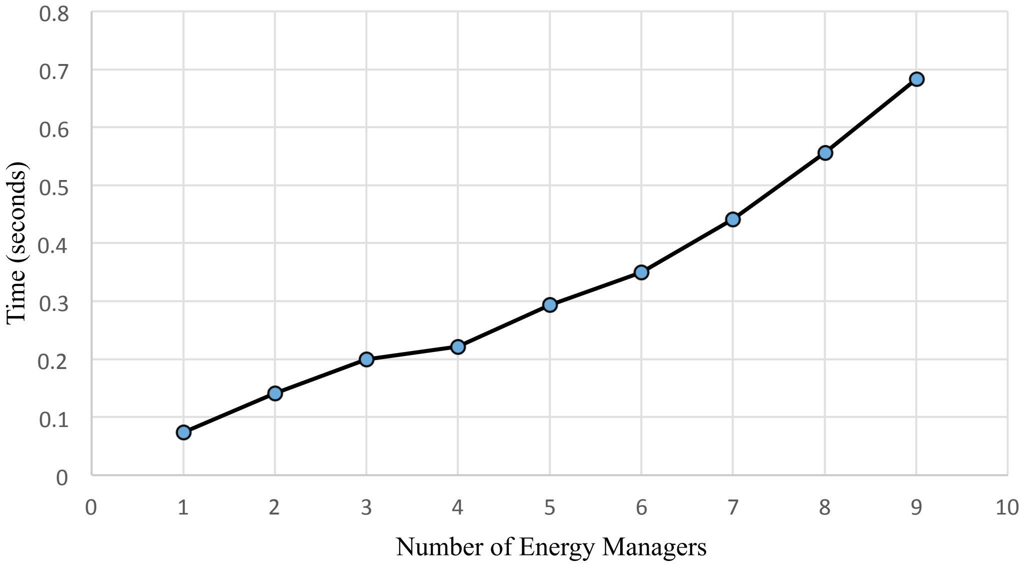


Figure 5-1: DR Delivery Latency for a one HAN

Therefore, the increase in delivery time can be attributed to the additional time that it takes to generate a new process to deliver a signal to the additional Energy Manager by the *dr_post* script. In addition, it is safe to assume that the delivery of a ToU signal will be affected in the same manner except for the fact that the utility provider expects no immediate action is expected from the Energy Manager.

With an increase in the delivery time to each Energy Manager, it can be assumed that with a significantly larger number of Energy Managers in the system, the delivery time would eventually exceed the maximum acceptable response time. This dissertation requires the system to provide feedback on the results of a transmitted control DR signal within an hour after its transmission. Therefore, the increase in delivery time would affect the time that it takes each Energy Manager to act on the received control DR signal and in turn increase the time it would take for the Management Application to register the consumption drop. Making use of a multiple VM instances each hosting a Management Application could reduce the effect that increasing the number of Energy Managers has on the delivery time and ensure the scalability of the system.

Furthermore, cloud computing offers the use of automated load balancers that can be used to launch new Management Application VM instances and distribute the load. The load balancer could be configured to monitor the delivery times and response times of each household and automatically allocate Energy Managers to new Management Applications when the delivery and response times exceed the required limits.

In addition, the load balancer could be used to monitor the computing resources allocated to VM instances and dynamically manage the allocation of resources. If a VM instance requires more storage or processing power, the load balancer could automatically allocate additional resources to that instance to ensure that the applications have adequate resources to meet their performance requirements. However, it must be noted that the use of load balancers is outside the scope of this work and may be considered for future work.

5.2.2 Effect of the Number of Energy Managers on the Control DR Response Time for a Single HAN

In addition to the delivery of the control DR signal, it is essential that the utility provider is able to register the expected drop in consumption after the transmission of a control DR signal. Figure 5-2 illustrates how increasing the number of Energy Managers in the system affects the time that it takes for the utility provider's *grid_monitor* script to confirm that action has been taken based on the transmitted DR signal i.e. the total consumption has dropped to zero watts

(0W). It must be noted that due to the fact that the Device Emulator was used to represent the HANs and the load controllers for the multiple home deployment, the consumption values for "OFF" virtual load controllers are 0W.

The experiment involved the generation of a control DR by the utility provider's *dr_post* script and the reduction of each HAN's power consumption to 0W. However, instead of monitoring the entire grid, this experiment records the time that it takes for the *grid_monitor* script to register the drop in consumption of a single HAN after the transmission of the control DR signal.

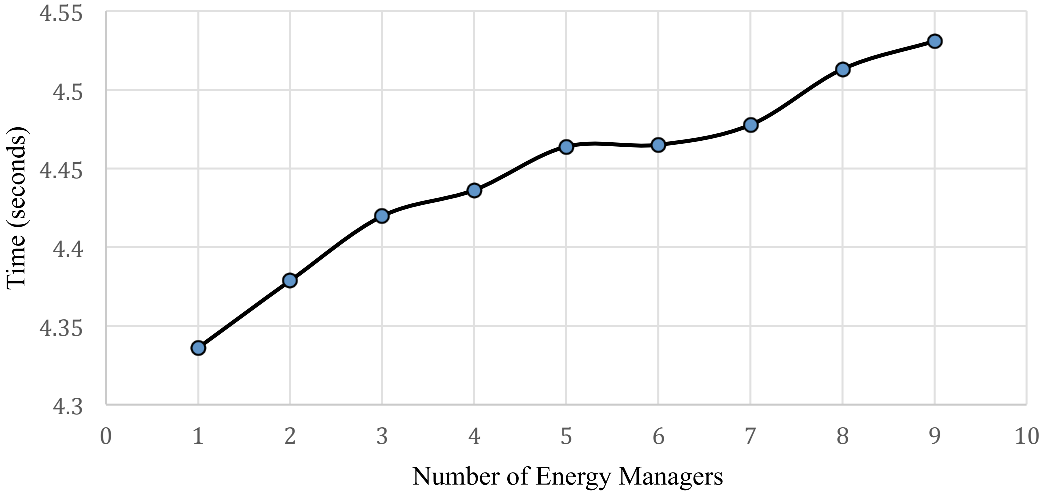


Figure 5-2: Time for an Energy Manager to Respond to control DR

As the number of Energy Managers increases, the time that it takes for the *grid_monitor* script to notice the drop in power consumption also increases. With a single Energy Manager, it takes 4.336 seconds for the *grid_monitor* script to notice the drop, while it takes 4.531 seconds for the script to notice the drop when there are 9 Energy Managers.

Like the previously described experiment, the increase in Energy Managers causes an increase in the number of processes that the *grid_monitor* script has to generate. Furthermore, it must be noted that the utility provider's domain is represented by a single virtual machine that makes the OpenMTC server available through its TCP port 15000. Therefore, all the requests go through a single port and may further contribute to the increase in the time it takes for the *grid_monitor* script to register the consumption drop.

An increase in response time affects the time that it takes the utility provider to notice a drop in consumption. Therefore, a utility provider may not be able to achieve the required drop in consumption within the required time. A real world implementation will require thousands of homes and may further increase in the time that it takes the Management Application to register the drop in consumption beyond the 1 hour requirement defined in Chapter 3. Therefore, as previously described, the system could meet the required response times in a real world implementation by providing multiple VM instances and Management Applications that are each responsible for a limited number of Energy Managers. This will ensure that the response time does not exceed the required 1-hour for each Management Application and its Energy Managers. In addition, a load balancer could be used to monitor and control the allocation of Energy Managers to Management Applications, as described in the previous section.

5.2.3 Effect of the Number of Energy Managers on the Time for the Total Consumption to Drop

In this scenario, the utility provider uses the control DR signal to reduce the power consumption of all the HANs in the system to 0W. The time that it took the Management Application, using the *grid_monitor* script, to register the total consumption drop was recorded for each increment in the number of Energy Managers in the system.

Figure 5-3 illustrates the relationship between the number of Energy Managers and the time taken for the utility provider's *grid_monitor* script to register the required power consumption drop. Similar to the behavior presented in the previous experimental scenarios, as the number of Energy Managers increases, the time that it takes to notice a drop in the total consumption also increases.

With one Energy Manager (or HAN), the time taken to notice the drop was 4.336 seconds and gradually increases to 4.796 seconds with 9 Energy Managers being monitored. This average increase can be attributed to the time that it takes for each Energy Manager to receive the control DR signal and act on it by turning off all the virtual load controllers and therefore reducing the power consumption to the 0W. It must be noted that the fact that the time taken for 8 energy managers is less than that of 7 energy managers is an anomaly that occurred during the test runs.

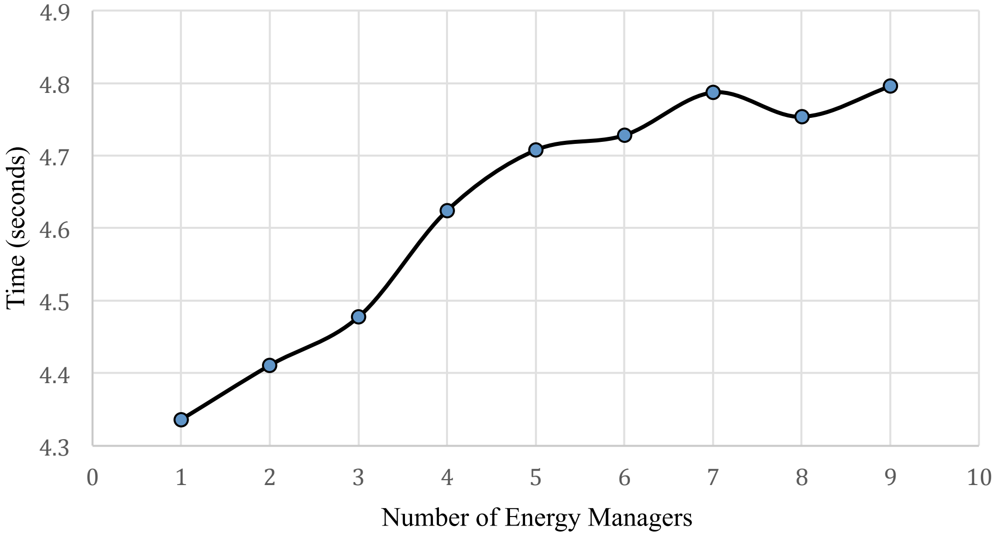


Figure 5-3: Time for Total Consumption to Drop

As has been discussed, an increase in the time that it takes for the Management Application to register the drop in total consumption on the grid will affect the ability of the utility provider to adequately manage their electricity grid. A longer response time would delay the time that it takes the utility provider to notice the effect of their transmitted control DR signal and confirm that the required consumption drop has been achieved. Therefore, the system could make use of multiple utility provider VM instances and an automated load balancer to ensure that each Management Application is responsible for a limited number of Energy Managers and therefore, monitor the grid consumption more effectively.

5.3 Time-of-Use

This scenario involves the retrieval of a ToU signal from the utility provider's OpenMTC server by each end-user's Energy Manager. The utility provider decides on the ToU electricity rate and POSTs it to the OpenMTC server using the *dr_post* script. Thereafter, the *home_server* script GETs the signal from the ToU container on the OpenMTC server.

The time it takes a single Energy Manager to retrieve the ToU signal from the OpenMTC server is recorded as the number of Energy Managers retrieving from the same container increases. It must be noted that no control action is taken in this experiment; the focus is to investigate how an increase in the number of Energy Managers and HANs affect the retrieval time of ToU signals from the OpenMTC server.

Figure 5-4 illustrates the effect that increasing the number of Energy Managers has on the time that it takes an end-user's *home_server* script to retrieve the ToU signal from the utility provider's OpenMTC server. As the number of Energy Managers increase, the time taken to retrieve the ToU signal from the OpenMTC server remains relatively the same i.e. between 0.01 seconds and 0.014 seconds.

With a single Energy Manager retrieving the ToU signal from the container on the utility provider's OpenMTC server, it takes 0.0108 seconds, while it takes 0.0116 seconds to retrieve the ToU signal with 9 Energy Managers in the system. This is because the GET requests for the signals do not happen at the same time and therefore, the time taken to GET the signal from the OpenMTC server should remain relatively the same.

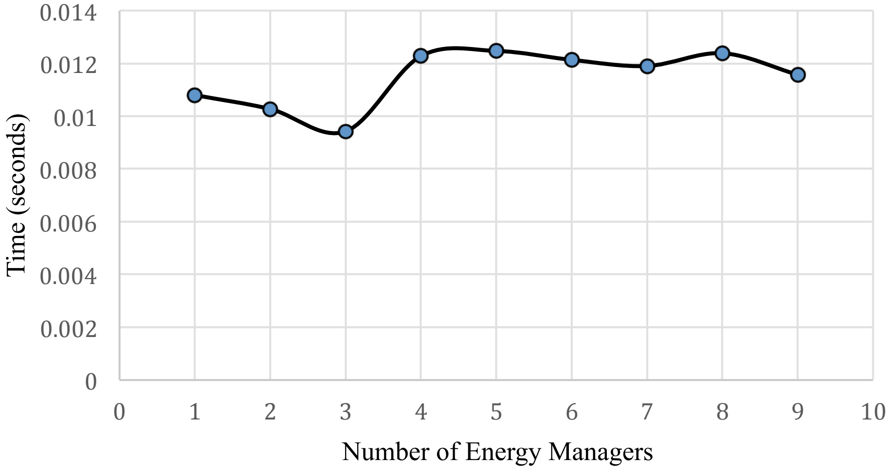


Figure 5-4: ToU Latency with Increasing Homes

Unlike the previous experimental scenarios, an increase in the number of Energy Managers does not affect the time that it takes each Energy Manager to directly retrieve information from the OpenMTC server. Therefore, using multiple Management Applications on different VM instances would have no effect on the time it takes an Energy Manager to retrieve information from the OpenMTC server.

5.4 Feedback

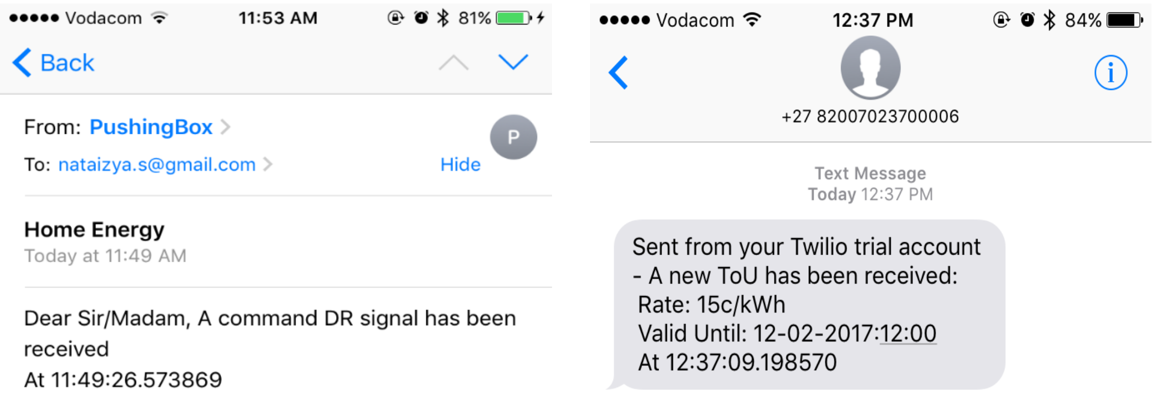
The ability for the system to provide feedback to end-users is a key function for the proposed SHEMS. It was described in the previous chapter that the end-user is offered feedback through their home Energy Manager application, specifically the *feedback* script. This script is responsible for providing the end-user with information on the consumption in their HANs and notifications on events that take place regarding the energy management services.

The feedback methods described in this section make use of various third party services to provide the end-user with information on their home power consumption and the energy management services. This shows that the prototyped system is able to meet the interoperability requirement described in Chapter 3.

5.4.1 Notifications

To provide notifications, the *feedback* script implemented an email and short message service (SMS) method to deliver notifications to end-users based on their preferred method. To illustrate the versatility of the prototyped SHEMS, two different notification methods were implemented, shown in Figure 5-5.

One method involved the use of the PushingBox [52] API service to provide email notifications to the end-user while the other method in the script provides notifications to the end-user via SMS using the Twilio [53] service. It must be noted that the Twilio service requires a paid subscription to provide notifications without the trial statement included in the generated SMS. However, for the purpose of this work, the trial Twilio account was enough to illustrate the notifications functionality via SMS.



From: PushingBox

To: nataizya.s@gmail.com

Home Energy

Today at 11:49 AM

Dear Sir/Madam, A command DR signal has been received
At 11:49:26.573869

+27 82007023700006

Sent from your Twilio trial account
- A new ToU has been received:
Rate: 15c/kWh
Valid Until: 12-02-2017:12:00
At 12:37:09.198570

Figure 5-5: Email and SMS notification screenshots

Table 5-3 shows the time it takes for notifications to be delivered using the two methods. It was noted that the time taken between the transmission of the notification from the *feedback* script to its reception by the end-user either via email or SMS was always under a minute. The resolution of the measured times could not exceed a minute precision because of the limitations of the mobile phone’s clock that was used to measure the latencies.

Table 5-3: Delivery times for notifications

	Sent at:	Delivered at:
PushingBox	11:49 AM	11:49 AM
Twilio	12:37 PM	12:37 PM

5.4.2 End-User remote consumption monitoring and appliance control

In addition to receiving notifications, the end-user must be able to remotely monitor and control appliances in their HAN according to the requirements and objectives of this dissertation. To enable this functionality, Freeboard was used to display appliance-specific information on separate widgets. In addition, the current ToU rate and other HAN related information is displayed on the web dashboard. This dashboard is accessible through a public uniform resource locator (URL) i.e. *https://freeboard.io/board/RAPnA4*. In addition, control buttons were added to the dashboard to enable remote control of the listed appliances.

Figure 5-6 illustrates the implemented Freeboard web dashboard. Data was streamed to the dashboard using the Pubnub service and reflected on the dashboard instantaneously from its transmission by the Energy Manager *feedback* script.

Region
Cape_Town
Gateway ID
gateway_1234
Area
15

Current Total Consumption
11.042 kWh
Current Total Power
9.5 Watts

6EBF - Location
Roof
6EBF - Alias
Geyser
6EBF - Consumption
6EBF - Power
6EBF - Max Power

70B3 - Location
Living Room
70B3 - Alias
Light
70B3 - Consumption
70B3 - Power
70B3 - Max Power

5C83 - Location
Living Room
5C83 - Alias
Fan
5C83 - Consumption
5C83 - Power
5C83 - Max Power

Figure 5-6: Freeboard web dashboard

5.5 Connectivity & Reliability

The connectivity of the system is evident in the end-to-end communication between the applications and the HAN (both the Device Emulator and the load controllers via the FHEM server). In addition, the reliability of the system, as defined in Chapter 3, is inherent in the ability of each stakeholder receiving the correct information. For instance, the appliance-specific data collected from the HAN was consistent with the data displayed on the web dashboard. This was verified using FHEM's web interface that is accessible through a URL, i.e. *http://localhost:8083/fhem*, on the VMware host.

The interface lists all the devices connected to the FHEM server and their related data. The consumption data collected by each of the EC3000 devices is shown on the interface and the state of the FS20 devices is also displayed on the interface. It was apparent that the data collected by FHEM was consistent with the data collected by the Energy Manager.

5.6 Heterogeneity & Flexibility

The abstraction provided by the OpenMTC middleware framework supports the integration of various applications and devices within the proposed SHEMS. Therefore, as has been demonstrated, the system is capable of integrating diverse applications to provide custom energy management solutions to utility providers and end-users. Furthermore, the support of third party applications adds to the heterogeneity and flexibility of the system and ensures that developers and utility providers can tailor each Energy Manager to the specific requirements and preferences of end-users.

The FHEM server's support for custom device modules further extends the flexibility of the proposed system for end-users. This ensures that other load controllers, which may not be supported by FHEM by default, can be integrated into the system with custom modules.

5.7 Chapter Discussion

This chapter presented the evaluation results and analysis of the effects that an increasing number of Energy Managers and HANs has on the time it takes to deliver the energy management signals to each Energy Manager.

The chapter first described the experimental scenarios used to investigate the effect of an increasing number of Energy Managers on the energy management services. The scenarios involved investigating the effect on the delivery time of the control DR signal to the Energy

Managers in the system. The time it took the Management Application to register a drop in consumption after the generation of a control DR was also investigated with an increasing number of Energy Managers. Lastly, the time it took to retrieve data, i.e. a ToU signal, from the OpenMTC server by an Energy Manager was investigated with an increasing number of Energy Managers also requesting the same data from the OpenMTC server.

It was shown that with an increase in the number of Energy Managers, there was an increase in the time that it took for the signals to be delivered to each Energy Manager and the time that it took for each of them to act on the control DR signal. Therefore, the time that it took for the utility provider to register a change in the consumption from the generated control DR was linearly affected by the number of Energy Managers in the system. This was attributed to the overhead introduced by the additional Energy Managers. The control DR requirements in Chapter 3 state that the Energy Manager must be able to receive a control DR signal, operate the required appliances and provide feedback to the utility provider within an hour after the transmission of the signal. Therefore, the acquired results illustrate that this whole process meets the requirements set out for the control DR signal. Similarly, the ToU signal requirements state that the Energy Manager must receive the signal at least 5 minutes after its transmission and the achieved retrieval times show that the system meets the requirements.

Furthermore, the connectivity and reliability of the system were discussed. The system was found to meet these requirements based on the successful end-to-end communication between the stakeholders. The reliability of the system was verified by comparing the data collected by the FHEM server with the data collected by the Energy Manager. The heterogeneity and flexibility of the system were provided through the OpenMTC middleware and the FHEM server and the integration of different third party applications.

To illustrate the feedback functionality of the system, the notification methods, i.e. Email and SMS, were presented and their respective delivery times assessed. It was noted that each of the notification methods provided feedback to the end-user within a minute of transmission of the feedback data from the Energy Manager. In addition, the web dashboard (Freeboard) that provided remote monitoring and control of the home environment for the end-user was presented. Appliance-specific information, the current ToU rate and other HAN related information is displayed using various widgets on the web dashboard.

The next chapter will present conclusions and recommendations for future work.

6. Conclusions & Recommendations

6.1 Summary

It was highlighted in the first chapter that utility providers could leverage SHEMSs to better manage the electricity grid and match the energy demand to the available generation capacity. Furthermore, a thorough literature review was conducted to investigate current SHEMS solutions and their implementation strategies. It was discussed that some of the current solutions do not address issues such as scalability, heterogeneity and flexibility. Therefore, the chapter went on to discuss cloud computing and standards-compliant middleware to provide scalable and flexible energy management services to the various stakeholders.

Thereafter, the stakeholders of the system and their requirements were defined and presented in chapter 3. These stakeholders included the utility provider, the consumer, defined as an end-user, and the HAN. Furthermore, the system's technical requirements were also presented and three energy management services defined i.e. the control DR service, the ToU service and the consumption monitoring service. Finally, based on the presented requirements and the defined services, the functional system architecture to meet the requirements and support the functionality of the proposed SHEMS was presented. Using the proposed architecture, utility providers can use the defined energy management services and leverage a standardised distributed middleware platform and cloud computing to increase consumer participation in grid management. The standardised distributed middleware increases the heterogeneity of the system while cloud computing supports the system's scalability by providing virtual computing resources to the energy management applications.

Chapter 4 presented the implementation of the proposed SHEMS and the evaluation platform. The utility provider was allocated a Management Application on one VM instance on the OpenStack cloud server, while the end-user was allocated an Energy Manager application on a separate VM instance on the same cloud server. Furthermore, the Management Application made use of an OpenMTC server and OpenMTC gateway to provide and manage the energy management data. The Management Application and Energy Manager application were implemented using a modular approach with multiple scripts each responsible for separate functions. This allows the system to be flexible and adaptable to the varying needs of consumers as each script can be customised to meet the required functionality of a given end-user. Furthermore, the flexibility and adaptability of the system was further increased with the use of the FHEM server that also follows a modular implementation approach to ensure that a wide range of devices can be used within the HAN. To conclude the chapter, the performance metrics and evaluation tools used to verify the functionality and performance of the prototyped system were then presented.

Finally, Chapter 5 presented the results and a discussion on each of them. The effect of an increasing number of HANs on the delivery time and effectiveness of the control DR and ToU services was investigated. Furthermore, the functionality of the SHEMS was verified and illustrated using screenshots of the notification methods and the web dashboard.

6.2 Conclusions

This dissertation proposes the use of a SHEMS to improve consumer participation in DR programs. The use of a SHEMS leveraging cloud computing and a distributed standards-compliant middleware was proven to be capable of supporting the defined energy management services. The effect of an increasing number of HANs in the system on the effectiveness of the control DR, ToU and consumption monitoring services was investigated.

It was noted that the number of Energy Managers in the system affected the delivery time of control DR signals and the time it took for the utility provider to register a drop in consumption after the transmission of the control DR signal. However, as highlighted in chapter 5, the number of Energy Managers was limited to 9 due to the client constraints on the OpenStack server. The time that it took the utility provider to deliver the control DR signal to each Energy Manager and register a consumption drop increased with the number of Energy Managers in the system. This was attributed to the overhead introduced by servicing an additional Energy Manager with an additional process. To remedy this, limiting the number of Energy Managers allocated to a single Management Application was suggested. This would ensure that the processing overhead does not exceed the required delivery and response times.

However, it was shown that the effect on the response time and delivery time of the consumption monitoring and control DR services does not exceed the required time constraints within the experimental conditions. Similarly, the time it took one Energy Manager to retrieve data from the OpenMTC server was not affected by the number of Energy Managers in the system and met the defined requirement.

The prototyped SHEMS makes the deployment of services easy and customisable through the use of energy management applications and modular programming. Furthermore, the use of cloud computing allows utility providers to deploy applications and services to market quickly while providing efficient management of computing resources. The middleware of the system provided the necessary abstraction to integrate various applications, devices and protocols into the system.

Therefore, based on the prototyped system's performance and functionality, this dissertation has shown that a SHEMS can be used to support energy management services by leveraging cloud computing and a standards-compliant distributed middleware platform. Furthermore, this work has illustrated the effect that an increasing number of subscribed end-users, represented by their Energy Managers, has on the energy management services. The work illustrates that the system meets the defined requirements within the experimental constraints and provides a basis for future work.

However, it must be noted that this work could not adequately investigate the scalability of the system due to hardware limitations and the maximum number of VM instances that could be launched on the OpenStack server. In addition, the use of a load balancer to manage the allocation of Energy Managers to Management Applications was briefly discussed but not implemented. Furthermore, this work was conducted in a highly experimental environment, which provided high connectivity between the various stakeholders of the system. This may not reflect the conditions of a real world implementation that could be dependent on different ISPs and varying network conditions that may affect connectivity and the reliability of the system.

6.3 Recommendations

Despite meeting the defined requirements, the prototyped system could be extended in several ways to improve its viability and effectiveness for real world implementation. Furthermore, there are other issues that were not addressed in this dissertation that are worth exploring.

The first recommendation is that future work considers the extension of scalability tests beyond 9 Energy Managers, as highlighted in chapter 5. This would allow the system to be tested under conditions that closely reflect the conditions of a real-world implementation. In addition, the use of additional FHEM servers and load controllers to represent HANs, instead of using a Device Emulator, would more accurately reflect an actual implementation. In addition, the automatic allocation of Energy Managers to Management Applications based on performance metrics could be investigated and implemented in future using cloud computing load balancers.

With highly connected systems such as the proposed SHEMS, security is a key issue that needs to be adequately addressed. Liu et. al. [16] and Wang et. al. [59] highlight key cyber security issues that must be considered in distributed energy management systems. Availability, confidentiality and integrity are the three high-level security objectives emphasized. In addition, future work could consider securing the data by encrypting it before it is transmitted and further explore the network and communication requirements of smart home energy management systems.

This work discussed the availability of the SHEMS to the various stakeholders but did not adequately address the confidentiality and integrity of the system. The integrity of the system refers to the prevention of unauthorised access and modification of data, while confidentiality prevents the unauthorised disclosure of information. Therefore, security across the system needs to be investigated to ensure that the end-user data, appliance-specific data and functionality of the system is protected from unauthorised access and malicious use. In addition, implementing a failover framework to increase the robustness of the system is also another feature that could be considered for future work.

Lastly, to increase the effectiveness of energy management based on ToU rates and end-user consumption patterns, advanced algorithms could be used to adapt the automated control of appliances to various conditions e.g. the number of end-users in a single home. These algorithms could be implemented in the Energy Manager and Management Application to provide analysis of consumption data and autonomous management of various control elements in the system.

References

---

[1] "SAPP Annual Report 2015," 2015. [Online]. Available: http://www.sapp.co.zw/areports.html.

[2] R. E. Brown, "Impact of Smart Grid on distribution system design," in *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, 2008, pp. 1–4.

[3] P. Siano, "Demand response and smart grids—A survey," *Renew. Sustain. Energy Rev.*, vol. 30, pp. 461–478, 2014.

[4] B. Shen, G. Ghatikar, Z. Lei, J. Li, G. Wikler, and P. Martin, "The role of regulatory reforms, market changes, and technology development to make demand response a viable resource in meeting energy challenges," *Appl. Energy*, vol. 130, pp. 814–823, 2014.

[5] C. Wilson, T. Hargreaves, and R. Hauxwell-Baldwin, "Smart homes and their users: a systematic analysis and key challenges," *Pers. Ubiquitous Comput.*, vol. 19, no. 2, pp. 463–476, 2015.

[6] A. A. Khan, S. Razzaq, A. Khan, F. Khursheed, and Owais, "HEMSs and enabled demand response in electricity market: An overview," *Renew. Sustain. Energy Rev.*, vol. 42, pp. 773–785, 2015.

[7] G. Fersi, "Middleware for internet of things: A study," in *Proceedings - IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS 2015*, 2015, pp. 230–235.

[8] "European Telecommunications Standards Institute (ETSI)." [Online]. Available: http://www.etsi.org/. [Accessed: 13-Apr-2017].

[9] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of Cloud computing and Internet of Things: A survey," *Futur. Gener. Comput. Syst.*, vol. 56, pp. 684–700, 2016.

[10] B. M. Salih and H. A. Edreis, "Comparison between Virtualization and Cloud Computing," vol. 5, no. 6, pp. 195–199, 2016.

[11] B. Zhou, W. Li, K. W. Chan, Y. Cao, Y. Kuang, X. Liu, and X. Wang, “Smart home energy management systems: Concept, configurations, and scheduling strategies,” *Renew. Sustain. Energy Rev.*, vol. 61, pp. 30–40, 2016.

[12] J. Han, C.-S. Choi, W.-K. Park, I. Lee, and S.-H. Kim, “Smart home energy management system including renewable energy based on ZigBee and PLC,” *IEEE Trans. Consum. Electron.*, vol. 60, no. 2, pp. 198–202, 2014.

[13] A. Mahmood, H. Fakhar, S. H. Ahmed, and N. Javaid, “Home Energy Management in Smart Grid,” in *The International Industrial and Information Systems Conference*, 2014, pp. 1–2.

[14] Y. Ozturk, D. Senthilkumar, S. Kumar, and G. Lee, “An intelligent home energy management system to improve demand response,” *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 694–701, 2013.

[15] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, “A Survey on Demand Response Programs in Smart Grids: Pricing Methods and Optimization Algorithms,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 1, pp. 152–178, 2015.

[16] J. Liu, Y. Xiao, S. Member, S. Li, W. Liang, and C. L. P. Chen, “Cyber Security and Privacy Issues in Smart Grids,” *IEEE Commun. Surv. Tutorials*, vol. 14, no. 4, pp. 981–997, 2012.

[17] Z. Wen, D. O’Neill, and H. Maei, “Optimal Demand Response Using Device-Based Reinforcement Learning,” *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2312–2324, Sep. 2015.

[18] B. Chai, J. Chen, Z. Yang, and Y. Zhang, “Demand Response Management With Multiple Utility Companies: A Two-Level Game Approach,” *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 722–731, Mar. 2014.

[19] S. Nolan and M. O’Malley, “Challenges and barriers to demand response deployment and evaluation,” *Appl. Energy*, vol. 152, pp. 1–10, 2015.

[20] F. Shariatzadeh, P. Mandal, and A. K. Srivastava, “Demand response for sustainable energy systems: A review, application and implementation strategy,” *Renew. Sustain. Energy Rev.*, vol. 45, pp. 343–350, 2015.

[21] S. Sofana Reka and V. Ramesh, “A Smart Survey on Demand Response Potential in

Global Energy Market," *Indian J. Sci. Technol.*, vol. 8, no. S9, p. 474, May 2015.

[22] L. Gkatzikis, I. Koutsopoulos, and T. Salonidis, "The role of aggregators in smart grid demand response markets," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 7, pp. 1247–1257, 2013.

[23] D.-M. Han and J.-H. Lim, "Design and implementation of smart home energy management systems based on zigbee," *IEEE Trans. Consum. Electron.*, vol. 56, no. 3, pp. 1417–1425, 2010.

[24] Y.-G. Cheong, Y.-J. Kim, S. Y. Yoo, H. Lee, S. Lee, S. C. Chae, and H.-J. Choi, "An ontology-based reasoning approach towards energy-aware smart homes," in *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, 2011, pp. 850–854.

[25] H. Lee, W. Park, and I.-W. Lee, "A Home Energy Management System for Energy-Efficient Smart Homes," vol. 2, pp. 142–145, 2014.

[26] O. A. Sianaki and M. A. S. Masoum, "Versatile energy scheduler compatible with autonomous demand response for home energy management in smart grid: A system of systems approach," in *2014 Australasian Universities Power Engineering Conference (AUPEC)*, 2014, pp. 1–6.

[27] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Hardware demonstration of a home energy management system for demand response applications," *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 1704–1711, 2012.

[28] "Zuli." [Online]. Available: https://zuli.io/technology/. [Accessed: 22-Sep-2017].

[29] "Wi-Fi Smart Plug with Energy Monitoring - TP-Link." [Online]. Available: http://www.tp-link.com/en/products/details/cat-5258_HS110.html. [Accessed: 25-Oct-2016].

[30] M. M. Rahman, M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Architecture of web services interface for a Home Energy Management system," in *2014 IEEE PES Innovative Smart Grid Technologies Conference, ISGT 2014*, 2014.

[31] M. Liska, M. Ivanic, V. Volcko, and P. Janiga, "Research on Smart Home Energy Management System," pp. 1–5, 2015.

[32] A. Saha, M. Kuzlu, and M. Pipattanasomporn, "Demonstration of a home energy

management system with smart thermostat control," *2013 IEEE PES Innov. Smart Grid Technol. Conf. ISGT 2013*, pp. 1–6, 2013.

[33] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Cla, "Middleware for internet of things: A survey," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 70–95, 2016.

[34] L. Zhou and J. Rodrigues, "Service-oriented middleware for smart grid: Principle, infrastructure, and application," *IEEE Commun. Mag.*, vol. 51, no. 1, pp. 84–89, 2013.

[35] M. A. Selvan and V. Prasanna Venkatesan Associate Professor, "A Generic Middleware Model for Smart Home," *J. Comput. Netw. Inf. Secur.*, vol. 8, no. 8, pp. 19–25, 2014.

[36] M. R. Hoque, M. H. Kabir, J. H. Jang, and S. H. Yang, "Middleware aided context-aware service for smart home," in *Proceedings of the International Symposium on Consumer Electronics, ISCE*, 2014, pp. 1–2.

[37] B. Bitzer and E. S. Gebretsadik, "Cloud Computing Framework for Smart Grid Applications," *2013 48th Int. Univ. Power Eng. Conf.*, pp. 1–5, 2013.

[38] P. Naveen, W. K. Ing, M. K. Danquah, A. S. Sidhu, and A. Abu-Siada, "Cloud computing for energy management in smart grid - an application survey," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 121, p. 12010, 2016.

[39] Y. Y. Y. Yang, Z. W. Z. Wei, D. J. D. Jia, Y. C. Y. Cong, and R. S. R. Shan, "A Cloud Architecture Based on Smart Home," *Educ. Technol. Comput. Sci. (ETCS), 2010 Second Int. Work.*, vol. 2, no. 60970130, pp. 440–443, 2010.

[40] S. Bera, S. Misra, and J. J. P. C. Rodrigues, "Cloud Computing Applications for Smart Grid: A Survey," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1477–1494, 2015.

[41] P. Mell and T. Grance, "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology."

[42] Y. T. Lee, W. H. Hsiao, C. M. Huang, and S. C. T. Chou, "An integrated cloud-based smart home management system with community hierarchy," *IEEE Trans. Consum. Electron.*, vol. 62, no. 1, pp. 1–9, 2016.

[43] M. Soliman, T. Abiodun, T. Hamouda, J. Zhou, and C.-H. Lung, "Smart Home: Integrating Internet of Things with Web Services and Cloud Computing," *2013 IEEE 5th Int. Conf. Cloud Comput. Technol. Sci.*, pp. 317–320, 2013.

[44] X. Ye and J. Huang, “A framework for cloud-based smart home,” in *Proceedings of 2011 International Conference on Computer Science and Network Technology, ICCSNT 2011*, 2011, vol. 2, pp. 894–897.

[45] M. L. Tuballa and M. L. Abundo, “A review of the development of Smart Grid technologies,” *Renew. Sustain. Energy Rev.*, vol. 59, pp. 710–725, 2016.

[46] H. I. Joshi and H. R. Choksi, “Development of Infrastructue for Residential Load to Reduce Peak Demand and Cost of Energy in Smart Grid,” vol. 3, no. 3, 2015.

[47] A. R. Khan, A. Mahmood, A. Safdar, Z. A. Khan, and N. A. Khan, “Load forecasting, dynamic pricing and DSM in smart grid: A review,” *Renew. Sustain. Energy Rev.*, vol. 54, pp. 1311–1322, 2016.

[48] “OpenMTC M2M Solutions for Smart Cities and the Internet of Things.”

[49] “OpenStack Open Source Cloud Computing Software.” [Online]. Available: https://www.openstack.org/. [Accessed: 11-Aug-2016].

[50] “PubNub.” [Online]. Available: https://www.pubnub.com/. [Accessed: 08-Nov-2016].

[51] “freeboard - Dashboards For the Internet Of Things.” [Online]. Available: http://freeboard.io/. [Accessed: 16-Nov-2016].

[52] “PushingBox.” [Online]. Available: https://www.pushingbox.com/. [Accessed: 18-Jun-2017].

[53] “Twilio.” [Online]. Available: https://www.twilio.com/. [Accessed: 18-Jun-2017].

[54] Oliver Jathe and Marcus Viererbe, “TheOpenTransporter.” [Online]. Available: http://www.theopentransporter.de/. [Accessed: 25-May-2017].

[55] “Home of FHEM.” [Online]. Available: http://fhem.de/fhem.html. [Accessed: 15-Aug-2016].

[56] “Busware.” [Online]. Available: http://busware.de/tiki-index.php?page=CUL. [Accessed: 06-Jun-2017].

[57] “JeeLabs.” [Online]. Available: http://jeelabs.net/projects/hardware/wiki/JeeLink. [Accessed: 06-Jun-2017].

[58] “VMWare Workstation.” [Online]. Available:

http://www.vmware.com/products/workstation.html. [Accessed: 06-Jun-2017].

[59] W. Wang and Z. Lu, “Cyber security in the Smart Grid: Survey and challenges,” *Computer Networks*, vol. 57, no. 5. pp. 1344–1371, 2013.

Appendix A

OpenMTC Resource Structure

OpenMTC implements a Create, Read, Update and Delete (CRUD) model that can be directly mapped to the commonly used HTTP RESTful methods i.e. POST, GET, PUT and DELETE. The proposed solution uses the OpenMTC platform as a distributed middleware to integrate the energy management applications with devices in the home. It does this by storing and accessing data on the platform using standardised data structures. Data from the home environment, the utility provider and end-users are stored on the middleware platform. In addition, applications are able to receive notifications for certain data related events that occur and thereafter, take appropriate action.

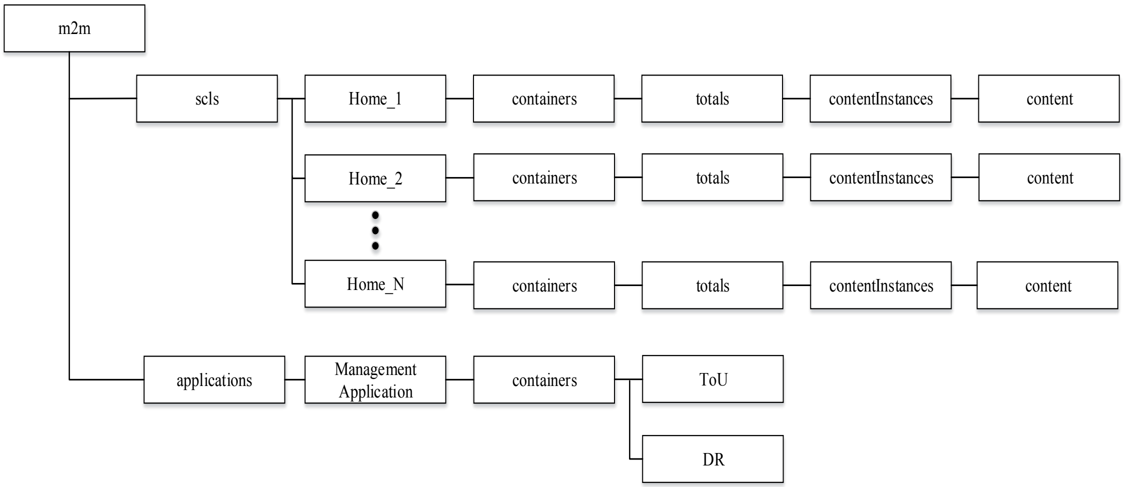


Figure 0-1: OpenMTC server resource tree

Figure 0-1 illustrates the OpenMTC server resource tree that defines how the energy management data is stored on the server. OpenMTC makes use of an application-container model where data is stored in containers within their appropriate applications. The utility provider posts information on the ToU and the DR in the *Management Application* containers while they access the *totals* container of *Home_N* to retrieve consumption information for a given home. Furthermore, each home has its OpenMTC gateway registered on the utility provider's server under the *scls* resource. Therefore, the energy consumption of each home is stored in its appropriate container on the OpenMTC server and the utility provider's Management Application can access this information to enable the utility provider to monitor each home's consumption.

To access store data on the OpenMTC server, an HTTP POST is sent to the server's URL i.e. *http://<server_base_url>/application_name/containers/container_name*. The <*server_base_url*> is the server's IP address and its port (e.g. *localhost:15000*), while the *application_name* and the *container_name* are the relevant application and container that the data is to be stored in. Table 0-1 lists some examples for the URLs used to get and retrieve data from the OpenMTC server and OpenMTC gateway. To retrieve data from the server, the HTTP GET method is used with the above URL and the *contentInstances* resource specified. Data is stored on the OpenMTC server and gateway in an encrypted Base64 format. When data is retrieved, the OpenMTC API script decodes the data and makes it available to the relevant scripts in a decoded format.

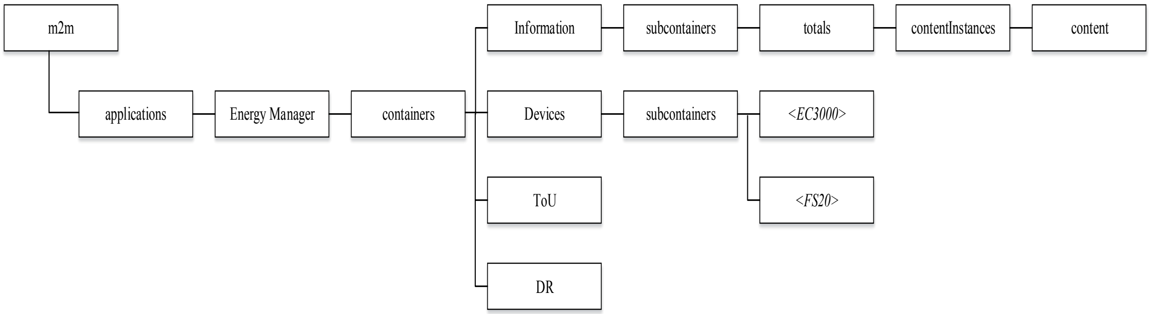


Figure 0-2: OpenMTC gateway resource tree

Figure 0-2 illustrates the OpenMTC gateway resource structure used by the end-user's Energy Manager application. Each Energy Manager is registered on the gateway as an application that consists of containers and subcontainers. The IWP makes use of the *Devices* container to store appliance specific data in the relevant EC3000 and FS20 subcontainers. The *totals* script makes use of the *Information* container to store the home's total consumption while the *ToU* and *DR* containers are used to store the current ToU rate and the control DR signal.

Furthermore, each of the containers and subcontainers has the data stored in Base64 format in the *contentInstance* resource. This resource contains current and historical data of the container in chronological order. To access the latest instance of data posted into the *totals* container, the *http://localhost:5000/m2m/applications/EnergyManager/containers/Information/subcontainers/totals/contentInstances/latest* URL was used. Other containers can be accessed in a similar manner.

Table 0-1: Examples of URLs used to get and retrieve data from OpenMTC server and OpenMTC gateway

	HTTP Method	Container	URL
OpenMTC Server	GET	*totals*	*http://<ipaddress>:15000/m2m/scls/Home_1/containers/totals/contentInstances/latest*
	POST	*ToU*	*http://<ipaddress>:15000/m2m/applications/ManagementApplication/containers/totals*
OpenMTC Gateway	GET	*totals*	*http://<ipaddress>:5000/m2m/applications/EnergyManager/containers/Information/subcontainers/totals/contentInstances/latest*
	POST	*DR*	*http://<ipaddress>:15000/m2m/applications/EnergyManager/containers/DR*