DEGREE PROJECT IN COMMUNICATION SYSTEMS, SECOND LEVEL
STOCKHOLM, SWEDEN 2017

Heterogeneous Residential Gateway Design Using OSGi

With multi-user and multi-service capabilities

GANESHKUMAR MANI

KTH ROYAL INSTITUTE OF TECHNOLOGY
INFORMATION AND COMMUNICATION TECHNOLOGY

Heterogeneous Residential Gateway Design Using OSGi

With multi-user and multi-service capabilities

GaneshKumar Mani

2017-06-08

Master's Thesis

Examiner
Gerald Q. Maguire Jr.

Industry Supervisor
Stephane Junique

KTH Royal Institute of Technology
School of Information and Communication Technology (ICT)
Department of Communication Systems
SE-100 44 Stockholm, Sweden

Abstract

As a result of developments, domestic usage of smart appliances by homeowners is increasing drastically. Clustering these appliances together and making them function as an efficient system defines a new place to live or new way of living called a “smart home”. While a smart home provides comfort to homeowners, realizing a smart home involves many technical and business oriented hurdles to be crossed.

The primary goal of this thesis work is to design and evaluate the design of a residential gateway. This gateway should be designed as a standardized, secure, open source, hardware independent, and interoperable Residential Gateway. A service-oriented architecture is proposed using the OSGi framework to design the residential gateway and its individual components. These components include an access control component for homeowner authorization, a resource management component for managing connected devices, an automation component to realize an automation service, and finally a context component to provide context aware services to the homeowner.

The final design proposed tries to solve the issues faced by some automation systems that are available in market. The evaluation of the design includes whether the design satisfies the basic requirements for a home gateway. This is followed by a comparison with existing systems with an emphasis on the improved features. The components proposed in the design could be used to construct a residential gateway that supports multiple services and multiple users. The proposed design will be taken into consideration during the design of Acreo’s home automation system.

Keywords

Home Automation, residential gateway, smart home

Sammanfattning

Som en följd av utvecklingen inom vetenskap och teknik så har användningen av smarta lösningar i hushållen ökat drastiskt. Att samla dessa apparater och få dem att fungera tillsammans som ett effektivt system, skapar ett nytt hem och ett nytt sätt att leva: ett smart hem. Å ena sidan så ger smartare lösningar ett bekvämare boende, men å andra sidan innebär det också många tekniska och affärsinriktade hinder att ta sig över.

Det primära målet med denna avhandling är att utforma en bostadsgateway som är att utforma en standardiserad, säker, open source, maskinvaruoberoende, interoperabel Residential Gateway. En serviceorienterad arkitektur föreslås med hjälp av OSGi-ramverket för utformning av bostadsgateway-komponenter. Komponenterna innefattar behörighetskontroll för husägare för tillgångskontroll, resurshanteringskomponenter för hantering av anslutna enheter, automationskomponent för att inkludera automationstjänst och slutligen kontextkomponent för att tillhandahålla kontextbevakad tjänster till husägaren.

Den slutliga designen som föreslås försöker lösa de problem som vissa automationssystem som finns på marknaden står inför. Utvärderingen av konstruktionen med grundläggande krav för att bygga hemgateways och med befintliga system ger information om de improviserade funktionerna. De komponenter som föreslås i konstruktionen kan användas för att bygga en bostadsgateway som stöder flera tjänster och flera användare. Den föreslagna konstruktionen kommer att beaktas vid utformningen av Acreos hemautomatiseringssystem.

Nyckelord

Home Automation, residential gateway, smart home

Acknowledgments

First of all, I would like to thank my supervisor form Acreo Stephane Junique for providing the opportunity for researching in the area of home automation and for the timely support during the thesis work. I would also like to thank other experts from Acreo who gave valuable input for the thesis work in various topics.

I would like to thank Professor Gerald Q. Maguire Jr. for providing valuable feedback for this thesis project.

Special thanks to Julie Robert for the emotional support during the thesis work and helping me in various complicated situations. And also to all my friends who helped me indirectly for the thesis.

Finally, I owe a very important debt to my parents for their continued support and encouragement without which I doubt that I would be in this place today.

Stockholm, June 2017
GaneshKumar Mani

Table of contents

ABSTRACT ..... **I**
KEYWORDS I
SAMMANFATTNING ..... **III**
NYCKELORD III
ACKNOWLEDGMENTS ..... **V**
TABLE OF CONTENTS ..... **VII**
LIST OF FIGURES ..... **XI**
LIST OF TABLES ..... **XIII**
LIST OF ACRONYMS AND ABBREVIATIONS ..... **XV**
1 INTRODUCTION **1**
1.1 BACKGROUND 2
1.2 PROBLEM DEFINITION 2
1.3 GOALS 4
1.4 PURPOSE 4
1.5 RESEARCH METHODOLOGY 4
1.6 DELIMITATIONS 4
1.7 STRUCTURE OF THE THESIS 4
2 BACKGROUND **7**
2.1 HOME AREA NETWORK 7
2.1.1 Wired *8*
2.1.2 Wireless *8*
2.1.3 Service Discovery and Service Delivery *10*
2.1.4 Remote Management *11*
2.2 HOME AUTOMATION SERVICES 12
2.2.1 What is automation? *14*
2.2.2 Home automation domain actors *14*
2.3 REQUIREMENTS FOR CONSTRUCTING A HOME AUTOMATION SYSTEM 15
2.3.1 Heterogeneity *15*
2.3.2 Mobility *16*
2.3.3 Extensibility *16*
2.3.4 Privacy and security *16*
2.3.5 Usability *17*
2.3.6 Context awareness *17*
2.4 SAMSUNG SMARTTHINGS HUB 17
2.4.1 SmartHub *18*
2.4.2 SmartThings Devices *18*
2.4.3 SmartThings cloud *18*
2.4.4 SmartApps *19*

2.4.5 Architecture ... *19*
2.4.6 Automation Management ... *20*
2.4.7 Evaluation ... *20*
2.5 INSTEON ... 21
2.5.1 INSTEON Hub ... *22*
2.5.2 INSTEON Devices ... *22*
2.5.3 INSTEON Application ... *23*
2.5.4 Automation Architecture ... *23*
2.5.5 Evaluation ... *23*
2.6 QUALCOMM 2NET SYSTEM ... 25
2.6.1 2net Hub ... *25*
2.6.2 2net Platform ... *26*
2.6.3 2net Application ... *26*
2.6.4 Evaluation ... *26*
2.7 HOMER ... 28
2.7.1 Homer Components ... *28*
2.7.2 Services ... *29*
2.7.3 Framework ... *29*
2.7.4 Automation Architecture ... *29*
2.7.5 Evaluation ... *30*
2.8 VIRTUALIZED SERVICE GATEWAY ... 31
2.8.1 Evaluation ... *32*
2.9 SUMMARY ... 33
2.9.1 Design comparison ... *33*
2.9.2 Summary of use cases ... *35*
3 METHODOLOGY ... 37
3.1 SERVICE ORIENTED ARCHITECTURE ... 37
3.1.1 Benefits of SOA ... *38*
3.1.2 SOA using OSGi ... *38*
3.2 ACCESS CONTROL MECHANISM ... 41
*3.2.1 Mandatory Access Control (**MAC**)* ... *41*
*3.2.2 Discretionary Access Control (**DAC**)* ... *41*
*3.2.3 Role Based Access Control (**RBAC**)* ... *42*
3.2.4 eXtensible Access Control Mark-up Language (XACML) ... *42*
3.3 REMOTE CONNECTION ... 42
3.4 AUTOMATION ... 44
3.4.1 Rule-based reasoning using production rules ... *44*
3.4.2 Type of triggers ... *45*
3.5 CONTEXT AWARENESS ... 45
4 EXPERIMENTAL GATEWAY DESIGN ... 49
4.1 THE ESSENCE ... 49
4.2 HOMEOWNER AUTHENTICATION MANAGEMENT ... 50

4.3 HOMEOWNER AUTHORIZATION MANAGEMENT 51
4.3.1 Access control component 52
4.3.2 Authorization procedure 52
4.3.3 XACML Policy Structure 54
4.4 CONTEXT AWARE RESOURCE MANAGEMENT 54
4.4.1 Resource Agent 55
4.4.2 Control Agent 55
4.4.3 Context Agent 55
4.5 HOME AUTOMATION MANAGEMENT 56
4.5.1 Policy Format 56
4.5.2 Automation component 57

5 IMPLEMENTATION AND ANALYSIS 59
5.1 XACML POLICYSET GENERATION 59
5.2 ONTOLOGY IMPLEMENTATION 61
5.3 BUNDLE IMPLEMENTATION 63
5.4 FUNCTIONAL USE CASE ANALYSIS 66
5.4.1 Use Case 1: In home or remote control of devices 66
5.4.2 Use Case 2: Home Automation 68
5.4.3 Use Case 3: Remote Patient Monitoring 71
5.5 NON FUNCTIONAL USE CASE ANALYSIS 73
5.5.1 Heterogeneity 73
5.5.2 Extensibility 73
5.5.3 Security and privacy 73
5.5.4 Context aware 74
5.5.5 Mobility 74
5.5.6 Usability 74

6 CONCLUSIONS AND FUTURE WORK 77
6.1 CONCLUSIONS 77
6.2 FUTURE WORK 77
6.3 LIMITATIONS OF THIS WORK 78
6.4 REQUIRED REFLECTIONS 78

REFERENCES 79

APPENDIX 85

List of Figures

Figure 1-1: Home environment with residential gateway with connected appliances 2
Figure 2-1: Home Area Network technologies 7
Figure 2-2: Types of home automation services 14
Figure 2-3: Samsung SmartHub 18
Figure 2-4: Samsung SmartThings high level architecture 19
Figure 2-5: INSTEON hub 22
Figure 2-6: 2net hub – the connectivity is shown via the indicator on the upper left and the successful transfer of data using the indicator on the upper right 26
Figure 2-7: Homer Architecture 28
Figure 2-8: Virtualized service gateway architecture 31
Figure 3-1: SOA primary entities 37
Figure 3-2: OSGi Layers 38
Figure 3-3: OSGi bundles and services 39
Figure 3-4: High-level ontology 46
Figure 3-5: Device class 47
Figure 3-6: Event class 48
Figure 3-7: Location class 48
Figure 4-1: Overall view of a residential gateway and its context 49
Figure 4-2: Authentication process 50
Figure 4-3: Access control component 52
Figure 4-4: Authorization component 53
Figure 4-5: Authorization process 53
Figure 4-6: Policy set, Policy, and rule 54
Figure 4-7: Resource modules 54
Figure 4-8: Context agent 56
Figure 4-9: Saving a new policy 58
Figure 4-10: Executing a policy 58
Figure 5-1: Policy set example in alpha 60
Figure 5-2: Policy set in XACML 61
Figure 5-3: Class hierarchy view in protégé 62
Figure 5-4: Object Property and Datatype property 62
Figure 5-5: Object Property and Datatype property view in protégé 63
Figure 5-6: OSGi bundles for HTTP service 63
Figure 5-7: UAService interface 63
Figure 5-8: Activator class 64
Figure 5-9: Authenticator implementation 64
Figure 5-10: LoginServelet 65
Figure 5-11: Login Screen using OSGi bundles 65
Figure 5-12: Sequence diagram for controlling a device 66
Figure 5-13: Action request 67
Figure 5-14: Automation request 69
Figure 5-15: Sequence diagram for saving an automation policy 69
Figure 5-16: Sequence diagram for automation rule execution 70
Figure 5-17: Sequence diagram for remote patient monitoring 71

List of Tables

Table 2-1: Summary of the five systems that were examined in this chapter 34
Table 3-1: Advantages and disadvantages for MAC 41
Table 3-2: Advantages and disadvantages for DAC 41
Table 3-3: Advantages and disadvantages for RBAC 42
Table 5-1: Overall summary with the proposed deisgn 76

List of acronyms and abbreviations

AC	Alternating Current
ACL	Access control list
ALFA	Abbreviated Language For Authorization
API	Application programming Interface
BLE	Bluetooth Low Energy
BAS	Building Automation System
BCE	Before Common Era
BPM	Beats Per Minute
CHI	Consumer Health Informatics
CPU	Central processing Unit
CE	European Certification
CO2	Carbon di Oxide
DAC	Discretionary Access Control
EHR	Electronic Health Record
EU	European Union
ECG	Electro Cardiogram
ECA	Event Condition Action
ECHO	Electronic Computing Home Operator
FDA	Food and Drug Administration
Gbps	Gigabytes per second
HAN	Home Area Network
HTTP	Hypertext Transfer Protocol
HVAC	Heating Ventilation Air Condition
HIS	Health Information System
HIPAA	Health Insurance Portability and Accountability Act
HomePNA	Home Phone line Networking Alliance
IP	Internet Protocol
ITU	International Telecommunication Union
ISP	Internet Service provider
ISO	International Standard Organization
ID	Identifier
ICT	Information and Communication Technology
KM	Knowledge management
LAN	Local Area Network
MRUG	Medical Research using Grids
MDDS	Medical Device Data System
MoCA	Multimedia over Coax Alliance
MAC	Mandatory Access Control
NAHB	National Association of Home Builders
NFC	Near Field Communication
NAT	Network Address Translation
OSGi	Open Source Gateway Interface
OWL	Web Ontology Language
PNP	Plug and Play
PEP	Policy Enforcement Point
PDP	Policy Decision Point
PRP	Policy Retrieval Point
PIP	Policy Information Point
PAP	Policy Administration Point

POM	Project Object Model
RG	Residential gateway
RDF	Resource description Framework
RBAC	Role Based Access Control
RMI	Remote Method Invocation
SSL	Secure Sockets Layer
SOCKS	Secure Socket
STAN	Simple Traversal of UDP NATs
SOAP	Simple Object Access Protocol
SDP	Session Description Protocol
SLP	Service Locating Protocol
SOA	Service Oriented Architecture
USB	Universal Serial Bus
URL	Uniform Resource Locator
UTP	Unshielded Twisted Pair
UPnP	Universal Plug and Play
VHT	Virtual Health Teams
VPN	Virtual Private network
WAN	Wide Area Network
WLAN	Wireless Local Area Network
Wi-Fi	Wireless Fidelity
WWW	World Wide Web
XML	Extensible Markup Language
XACML	eXtensible Access Control Mark-up Language

1 Introduction

According to the Oxford dictionary, “Home” is the place where one lives permanently and the place where one feels safe and comfortable. The human interpretation of comfort varies depending upon the place and time. In pre historic period around 100,000 BCE, early humans lived in caves for protecting themselves from cold, rain, and dangerous animals. Later, due to evolution and need, human beings constructed their own place to live and customized the place according to their needs. In the current years, the human view of the term “Home” and the degree of comfort with respect to this home has changed significantly. The introduction of modern appliances and innovative services using these appliances makes human beings feel that their home is a better, safer, and more comfortable place to stay.

As a result of successful innovation in the field of computer science, appliances such as refrigerator, air conditioner, washing machine, etc., have begun to think on their own and hence became “smart” appliances. For example, a smart air conditioner will maintain constant room temperature while at the same time optimizing its power usage, thereby reducing the home owner’s electricity bill [1]. With the introduction of Internet to the home and internetworking between the home appliances, a variety of services have been researched and today many solutions can be provided to homeowners. The combination of high-speed inter-connection and advanced home appliances has resulted in an advanced, intelligent, and safe place to live called a “smart home”.

In order to control and monitor the smart devices connected in the residence an essential device named “Residential Gateway” could be used [2]. This Residential Gateway (RG) acts as a central medium of communication between the appliances and the homeowner. Even though there are other distributed ways of implementing “Smart Home”, the usage of residential Gateway is taken into account for this thesis work. The concept of residential gateway is not a new idea, it already exist in all homes in the name of network terminal device (a modem or router or internet gateway). As an implicit understanding, the electronic devices can be connected to the residential gateways via wired connection or Bluetooth or Infrared or Wi-Fi or even via Internet.

The usage of residential gateways for connecting devices and controlling the devices is not uncommon. Electricity Company provides each home a smart electricity meter for measuring and billing electricity usage. By using the smart meters, the electricity provider could provide varied services to the homeowner [3],[4]. Medical field is also famous for its smart gateway implementation. In order to monitor patients resting in home, hospitals suggest to use smart devices for connecting the medical measurement devices such as blood pressure monitor, heart beat reader etc., so the data will be stored and doctors could view patient’s details any time they want [5],[6].

So to implement an automation system with electricity monitoring system and Medical monitoring system requires gateway for each system separately. This project aims in providing a solution that could solve this problem by implementing all the systems using one residential gateway. This is possible when each system are considered as individual service implemented in the residence. This ideology of considering systems as services in one box opens up yet another area of innovation where interoperability between services will be possible. Services could communicate with each other and make the home smarter. For example electricity service could communicate with air conditioning service for usage reduction for reducing load. At the same time interoperability requires many conditions to be met like security & privacy of homeowner and service provider. By resolving factors

affecting the implementation, this project provides a design of a residential gateway that will makes the home smart.

1.1 Background

Figure 1-1 presents a generalized overview of how smart appliances and other devices in the house could be connected to a residential gateway. From this figure, it is very clear that the residential gateway is the heart of the home and all of the various devices (i.e., smart appliances) are connected to it. This residential gateway could be considered a network terminal or interconnection device. With suitable bridging techniques, the residential gateway can allow Heating Ventilation Air Condition (HVAC) appliances and medical devices to connect to it [7]. The appliances can be hard wired or connected via any available transport technologies (such as Wi-Fi, Zigbee, or Bluetooth) and thus the residential gateway would be able to communicate with these devices. Each member of the household can access the residential gateway, but the level of usage and control could be set for each individual member of the household. Each home is unique and providing a home automation solution is a huge market-that must covering customers from different backgrounds. The possibilities of providing innovative solutions make this a very broad market. As the residential gateway acts as a primary device for the home automation process, optimizing it could improve the overall efficiency of the home system [8].

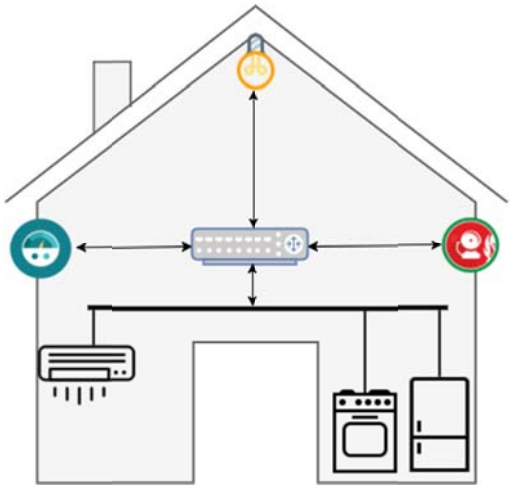


Figure 1-1: Home environment with residential gateway with connected appliances

1.2 Problem definition

Due to technological advancement, we see increasingly sophisticated consumer electronics and services that are operated by the homeowner. This includes various types of home automation, such as energy control, electronic access control (door locks), etc. One of the fundamental drawbacks of these advancements is that, they are evolving independently and under different & separate regulations. For example, home automation and E-health have their own regulations and hardware requirements, but they will need to coexist in the same residence. As quoted from the declaration of the E-Health European Ministerial Conference held in Prague in 2009: **"The lack of interoperability has been identified as one of the main areas to address."** [9]

Given the desire for interoperability as the main factor, a home owner will be very interested in having a residential gateway that could handle all the activities taking place in their home in a way that makes the process of automation simple. Instead of having one gateway for each service, it will be much easier for the homeowner to have a single residential gateway that can handle all the services. As a result, the requirements for a residential gateway have changed and this leads to a new path for innovation.

Providing simplicity for the homeowner comes with lot of other issues, such as trust, security, integrity and robustness. These issues must be taken into consideration both between different service providers and between the customer and each service provider.

Currently, there are several residential gateways available in the market for providing automation solutions [5], [10], [11]. However, they generally lack either interoperability or flexibility, specifically:

- No software independence ⇒ The software is coupled to the hardware,
- In many cases the residential gateway is sealed,
- Very few customer can program their residential gateway, and
- No interoperability between different services.

These missing features (which we will view as problems to be overcome) are further described below:

No Software independence

When the customer buys a residential gateway from a vendor, he/she is forced to buy software from the same vendor. This restricts them from choosing a different software vendor and prevents them from customizing their home. Additionally, this approach leads to chance a few vendors having market dominance [12].

Sealed Residential Gateways

For security reasons, many companies try to keep their residential gateway as closed as possible. In many cases the homeowner cannot even update the software, but rather is forced to replace the entire gateway if there is any problem. These sealed gateways further restrict the customer to running only the software provided by the residential gateway vendor.

Lack of Customer Programmable Gateways

In many cases, homeowners or other programmers cannot write a service application for their residential gateway. The software development team in the residential gateway vendor writes all of the software for their customers. This lack of customer programmability restricts homeowners who are interested in developing their own services from developing and deploying new services. This also results in vendor software dominance. While there are some software development tools available for some residential gateways, there is as of yet no standardization in the programming interfaces available to customers or third party software developers.

No Interoperability

Firstly there is no interoperability between residential gateways developed by different companies. Second there is no interoperability between services that are running in these residential gateways. As a result E-health service and home automation services run on separate boxes and have been developed based upon separate standards.

1.3 Goals

The primary goal of the project as given by Acreo Swedish ICT is to propose a residential gateway design based on the Open Source Gateway Interface (OSGi). Given the advantages offered by OSGi for services running in the residential gateway, this project has utilized OSGi as the base for developing the design.

1.4 Purpose

The resulting residential gateway design will be considered by Acreo Swedish ICT as a base home automation design for constructing their residential gateway that should support different types of services such as home automation and Telecare*.

1.5 Research Methodology

In order to design the residential gateway using OSGi I adopted design science research methodology. Using this method, a set of structured steps is followed in order to achieve the goal like understanding a problem, suggesting & developing a solution for the problem and finally evaluating the proposed solution. For this thesis, to understand the problem and find to the areas of improvement, different solutions were taken into consideration. These solutions were analysed and compared in detail with respect to few parameters or non-functional requirements. Once the areas of improvements are discovered, a residential gateway design is proposed and partially implemented. Finally, the proposed design is evaluated with respect to the non-functional requirements to know if it solves the problem faced by the existing solutions discussed in this thesis.

1.6 Delimitations

This thesis project focused only a high level design for a residential gateway and shows the possibility for a service provider to develop a service for it. How the service provider develops their complete service bundle and other domain specific features are not covered in this thesis. Due to time constrains only some parts of the gateway design were realized so we can only reach conclusions about those parts that were realized.

1.7 Structure of the thesis

Chapter 2 of this thesis describes different types of elements present in a home area network and briefly describes different types of home area network. The second half of the chapter provides the reader with information to understand the different types of residential gateways available in market. Additionally, individual residential gateways are evaluated with respect to different types of smart home requirements. Chapter 3 explains the method used for designing the gateway. This chapter briefly describes the different types of techniques and technologies that were used to design the residential gateway. Chapter 4 explains the residential gateway's design in terms of its individual components. In this chapter each component is explained in detail with figures and sequence diagrams. The first half of the Chapter 5 gives the implementation details of the parts of the residential gateway that were realized. The second half describes the details of how the residential gateway functions for specific use cases followed by an evaluation of the prosed

* Telecare is defined as the service provided monitoring and care to the patients when they remain in their home

residential gateway design with respect to the smart home requirements. Finally, Chapter 6 concludes this thesis and suggests further research.

2 Background

Electronic Computing Home Operator (Echo) IV, created by Jim Sutherland, an engineer from Westinghouse Corporation in 1966. Echo IV is the first known home automation system. In the initial stages, Echo IV automatically handled the Sutherland's family finances and later it was extended for temperature control, weather prediction, appliance usage, inventory tracking, and storing recipes [13].

In 1984, the National Association of Home Builders (NAHB) came up with the idea of a smart home controlled and automated by computers so that the owners can spend less time on the day to day home activities. The introduction of "A low cost and high IQ" device was one the primary ideas behind the scaling up the smart home idea [14].

Home automation does not stop with making the home smart, but also extends to whole buildings. As a result, Building Automation services (BAS) [15] came into existence for maintaining and automating HVAC services. A separate set of standards has been defined for building automation services to deal with medium or large-scale apartments, skyscrapers, or industrial settings. Robert McDowall says, in his book *Fundamentals of HVAC Systems*, "Modern air conditioning is critical to almost every facet of advancing human activity" [16]. According to many building owners, adding intelligence to HVAC systems and enhancing building operations, reduces energy cost, increases the effectiveness of building operations, and supports the building owner's sustainability efforts [17].

2.1 Home Area Network

A Home Area Network (HAN) is one type of computer network. A HAN provides the possibility to connect devices in the home enabling them to communicate with each other. These devices could be computers, smart phones, sensors, home appliances, etc. Depending upon the underlying transmission technology, the HAN could be deployed using wired or wireless technology [18]. A number of different HAN technologies are shown in Figure 2-1.

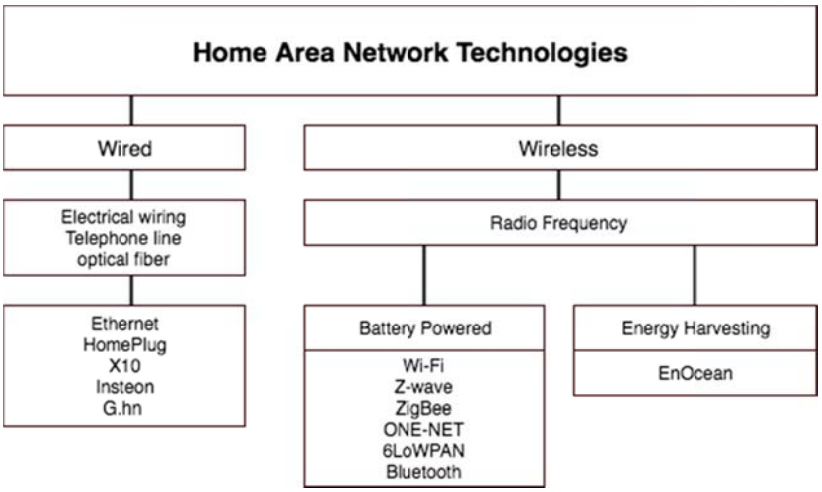


Figure 2-1: Home Area Network technologies

2.1.1 Wired

The most used and famous technology for implementing local area networks (LANs) based on wired networks is IEEE 802.3, also known as Ethernet. This technology is widely used to connect devices such as laptops, printers, audio/video equipment, etc. At the same time due to cost and power requirements (among other factors), this technology may not be interesting for connecting some types of home appliances. Within the home environment, Ethernet is commonly used with unshielded twisted pair (UTP) copper wires. However, this technology can also be used over coaxial cables, shielded twisted pair copper wires, and optical fibers. Fast Ethernet and Gigabit Ethernet support peak data rates of 100 Mbps and 1 Gbps respectively [19].

Although Fast Ethernet and Gigabit Ethernet offer fast and robust connection for services such as IPTV, this technology requires installation of high grade cabling systems (specifically category 5e or category 6 cabling) [20]. In contrast, technologies such as HomePlug [21], Home Phone line Networking Alliance (HomePNA) [22], and Multimedia over Coax Alliance (MoCA) [23] can utilize existing wiring, such as power mains, telephone wiring, and coaxial cables. HomePlug utilizes the existing electrical wiring in the home to communicate. MoCA uses existing coaxial cables for distribution of multimedia content in the home. Finally, HomePNA uses telephone lines and coaxial cables for sharing a single broadband access connection to/from the home.

Using the existing power lines for communicating with appliances in the home is the general purpose of X10 technology. Messages are sent to appliances (for example, to switch off a light) via the power line from custom controllers such as a remote control or a computer's interface. However, with a limited transmission rate close to 20 Bps, limited security features, incompatibility with transport protocols like TCP/IP, and finally issue with noise and attenuation when communicating over power lines, this technology is undesirable compared to its alternatives [24]. Insteon [11] tries to overcome the disadvantages of X10 for communication over power lines. Insteon uses both power line and radio frequency for communication in a peer-to-peer network structure. Each node in an Insteon network acts a receiver/repeater. Additionally, to connect with incompatible networks or to the Internet, certain Insteon devices have serial interfaces (such as USB, RS232, or Ethernet) [25].

International Telecommunication Union (ITU) introduced G.hn or G.9960 to provide secure connection in homes between devices over different media, such as phone lines, power lines, coaxial cables, and category 5 cables. Bridges can transmit information from one network domain to another domain [26]. With data rates up to 1Gbp, secure communication, multiple wiring methods, and interoperability between different networks, G.hn could become one of the main types of wired connectivity in a smart home.

2.1.2 Wireless

One of the greatest disadvantages of using a wired communication technology in a home environment is the installation and maintenance of the network. Once the networks are physically placed, wired connections are difficult to difficult to change or adapt to the growing number of connected appliances. For this reason, wireless LAN (WLANs) can be utilized. A device connected via WLAN could easily move in the home without any disruption while within the coverage range of one of the home access points. The IEEE 802.11x family is one of the popular methods of implementing WLANs. This technology is popularly referred to by the name Wireless Fidelity (Wi-Fi). Starting from the first IEEE 802.11 specification to the recent IEEE 802.11n specification the maximum data rate has

significantly increased from 1 Mbps to 600 Mbps. Although this technology has the advantages of providing wireless connectivity, the power consumption of Wi-Fi enabled devices makes it harder to integrate devices that have power restrictions (i.e., small battery powered devices).

In 1994, Ericsson introduced Bluetooth, a short range wireless communication technology. One of the main ideas behind Bluetooth is to replace the cables that were used to interconnect computers and other peripheral devices (such as a mouse, keyboards, etc.). As the technology is a low power, low cost, and as it uses an unlicensed frequency range the technology became popular with many manufacturers (such as Intel, Lenovo, Motorola, and Apple) implementing it in their devices. Depending upon the version of the Bluetooth the data rate varies from 3 Mbps to 24 Mbps. The recent version 4.0 provides a low energy mode where the device consumes only one hundredth of the energy consumed by its predecessors. Although Bluetooth cannot be used for service that require a very high data rate it can be used to control low power home appliances over short distances.

IEEE 802.15.4 [27] is another low power short-range communication technology well suited for connecting small devices in a home environment. ONE-NET is an open source wireless technology that uses IEEE 802.15.4 transceivers. All the messages in ONE-NET are encrypted using the XTEA2 algorithm and ONE-NET supports user key management. The basic data rate in ONE-NET is 38.4 kbps and could be extended to 230 kbps.

IEEE 802.15.4 is used as the physical and media access and control protocol used under the popular ZigBee trade name. The ZigBee Alliance introduced several standardized application profiles that can be used to implement different smart home scenarios, such as home automation, remote control, and smart energy [24]. ZigBee devices communicate with throughputs ranging between 20 to 250 kbps and maximum range of close to 100 meters. Devices using ZigBee technology take advantage of the fact that the protocol saves energy by effectively using long sleep periods. Similar to ZigBee, Z-Wave [28] is another communication technology based upon IEEE 802.15.4. Z-Wave was created by Zensys and is designed for use in home automation, especially for controlling lights in residential or commercial environments. As the technology is low cost and low power consumption, Z-wave can be easily integrated with battery powered home appliances. Compared to ZigBee, Z-wave faces fewer issues with respect to interoperability as ZigBee has the multi vendor ecosystem [29].

The Internet Engineering Task Force introduced IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) for carrying IPv6 over lower power wireless personal area networks. Using 6LoWPAN devices in a Personal Area Network (PAN) can communicate using IP based technologies. As described in RFC 4944 [30], 6LoWPAN can operate over a IEEE 802.15.4 link. As 6LoWPAN does not come with a routing protocol, it depends on another specification (e.g. IEEE 802.15.5) for mesh routing [31].

EnOcean [32] is a technology designed for ultra low powered network where the elements in the network are powered by energy harvesting (charging from solar power or other environmental means) [33]. The elements in the network communicate (following the IEEE 802.15.4 specifications) with each other distances ranging from 30 to 300 meters depending upon where they are placed. These devices operation in the 868 MHz band in Europe and 315 MHz band in North America [34].

2.1.3 Service Discovery and Service Delivery

In a home network in order to detect the devices and the services provided by the devices, the service discovery and service delivery protocols are used. These protocols provide the possibility for applications to discover which services are available and identify the characteristic features of the services that are available. The following are a few well-known technologies that are used in a HAN to provider service discovery and service delivery features:

Jini In 1988 Sun Microsystems introduced Jini [35], a Java based technology for connecting devices*. If offers features such as auto configuration and installation to share resources. As Jini is based on Java and takes advantage of Java-based systems. Jini is organized in a distributed fashion without a central node, therefore it is quite flexible and adaptable. The three primary protocols for Jini are discovery, join, and lookup. Whenever a Jini based device is connected the network, the discovery and join protocols are executed and then lookup is used when a user wishes to invoke a network service. The communication in Jini network uses Java based Remote Method Invocation (RMI). An RMI stub is used as a service proxy, hence clients can use a service with less information about the network [37].

UPNP Universal Plug and Play (UPNP) is a Microsoft initiated technology focused on providing automatic device discovery and zero configuration features. Using UPNP a network device can easily join a network, receive an IP address, inform other devices about its capabilities, learn other device's capabilities, and the device can leave the network having its footprints in the network [37]. As a web based technology, UPnP uses HTTP, XML, UDP, TCP, IP, SOAP, GENA, etc. Controlled devices and control points are the two main components of a UPnP network. The control points can discover a device in the network or send an action request to a device or listen for notifications from devices by using a subscription mechanism. The devices in a UPnP network can respond to action requests from a control point or send events. In this way devices communicate via a control point in the network, as they cannot directly communicate with each other [38].

Bonjour Similar to UPNP, Apple's bonjour service discovery protocol provides zero configuration and automatic connection features to their devices. Bonjour facilitates the usual network based activities such as sharing or printing a file via Bonjour enabled printers in the network and dynamically discoverable file servers [39].

* Jini has now been taken over by Apache for the project River [36].

SALUTATION The primary purpose of salutation is to provide rules for service discovery among devices that have dissimilar capabilities [40]. Introduced by the non-profit Salutation consortium, the salutation architecture is used for information exchange between different handheld wireless devices and automation devices in an office [41]. The architecture has two primary components: a transport manager and a salutation manager. The salutation manager acts a service broker in the network and provides a transport independent interface called SLP-API for client applications to communicate with a service provider's services. The transport manager is responsible for providing communication channels that are reliable. Using optional components, the architecture can provide a common interface for managing information flow between network protocols that are not similar [42].

SLP IETF's Service Location Protocol (SLP) is a lightweight and decentralized service discovery protocol [37]. SLP uses URL based commands. Using these URLs, a client user or application can access the list of service provided by a device and send request for the service they are interested in. There are three main components or agents in SLP: user, service, and directory. The user is the one who is requesting a service in the network. The service provides a service to other devices in the network. Finally, the directory acts as a centralized repository for storing service information and to cache advertisements from service agents in large networks [43].

Bluetooth SDP Compared to the previously mentioned service discovery protocols, Bluetooth's service discovery protocol (SDP) is specific to Bluetooth devices. This protocol provides service discovery, but does not provide service access, service registration, advertisements, or a notification when a service is available.

2.1.4 Remote Management

The people staying in the home are not always going to be in the vicinity of the devices connected in the home. To communicate with these devices or service those are running in the home there should be some remote management mechanism that implemented in the home. Using this mechanism a homeowner who is not at home could communicate with or control the devices from outside of the home. For example during vacation, a homeowner could monitor their home using a connected camera from anywhere in the world where they have an Internet connection. In order to connect to the home network via a residential gateway from a device connected to an external network, the following challenges should be considered:

Dynamic IP addresses When using a broadband connection, the IP address of the residential gateway (or even the home router) are not static, hence they can change (potentially very often). Devices within the local network could be dynamically assigned an IP address by a home router or by the ISP. These devices are typically located behind a NAT implemented by the home router or the ISP. As a result the IP addresses of devices connected inside the home networks are not fixed; hence this requires an extra step in their remote management. Dynamic DNS could be used to solve this, by running the dynamic DNS

service in the home's residential gateway. In this way the IP address of a device or service inside the HAN could be dynamically found based upon names for the devices or services.

Traversing firewalls and NAT

In order to create a private network with a large number of internal devices with different IP addresses a NAT can be used. Given one public IP address, the NAT enables each of the devices connected to the internal network to access the Internet. Using a rendezvous server, an external device could be accessible to the remote homeowner's device. NAT traversal, such as using the Simple Traversal of UDP NATs (STUN) protocol could be used to enable the internal and external devices to communicate.

A firewall is likely to be used between the interior network and the exterior network for several good reasons. However, firewalls restrict unsolicited network traffic to the home's private network. Punching a hole through the firewall for specific connections could solve this problem and SOCKS protocol could be used for doing this.

Security of the remote connection

Security is an important issue that should be considered when a home network is connected to the Internet. This connectivity potentially exposes the home network to different types of threats such as hacking and viruses. There are several solutions to secure the network when a remote client is to connect to the home network. One of the popular solutions is to deploy a Virtual Private Network (VPN) between the home network and the remote client.

2.2 Home Automation Services

Home automation is a general term for all the services that could be automated in the home. This automation is not restricted to simply controlling lights, but covers other domains such as remote care, smart energy, etc. The following are the different types of home automation services that could be provided in a home (see also Figure 2-2):

Light Control

Starting from basic functionality for a light bulb such as switching the bulb ON and OFF to adjusting the intensity of lights depending upon the intensity of sun light in each room, home automation could provide an intelligent light control feature. Note that such a service is not restricted to one bulb, but rather schemes could be created to set bulbs to different intensities (and even colors) to match the mood in a home. This type of control is implemented by many bulb manufacturers [44]. Using light control together with an automation trigger could provide a solution for a basic problem: turing off unnecessary lights. For example, linking a motion sensor [45] with a light bulb could be used to save energy by switching ON the light only if there is motion in a room.

Remote control Remote control includes both in the home and out-of-home control of devices. Inside the home this can be done using wireless technology (such as infrared, Bluetooth, or even Wi-Fi technology). For example, controlling an air conditioner using a small Bluetooth equipped remote control device. When it comes to controlling when out-of-home, a home automation system could enable a home owner to switch on a room's heating before coming home [46] to give a warm ambience or start a kitchen cooker before coming home in order to have dinner ready [47].

Smart Energy As mentioned with respect to lighting control, instead of switching on the lights in a corridor for the complete day - a motion sensor could switch on the lights when there is someone walking in the corridor. This could avoid unnecessary power consumption by the light. Different types of sensors could be placed in an apartment to collect information such as light intensity, presence, humidity, or temperature in order to provide valuable information that could be used to adjust settings of high power appliances in order to reduce energy consumption. Additionally, by identifying peak usage in the home by different appliances, scheduling and plan could be done together with the electricity provider enabling a reduction in usage during the time when the electricity provider faces peak demand. The cost savings by avoiding the need for special peaking power sources could reduce the cost of electricity for the homeowner [48].

Remote Care Medical device technology has developed greatly. Today patients in their home can use devices that were previously only used by doctors and laboratories. This technology allows a patient to take their own blood pressure [49], while a doctor sitting on the other side of the world could follow up. Other patient measurements such as insulin level, skin temperature, heart rate, activity, and ECG could be taken by a patient in their home as part of remote care. Finally, for elderly people, a monitoring nurse could remotely be notification if the person falls down so that any necessary action could be taken.

Security and Safety Finally, a home automation system could ensure that it is possible to guarantee the safety and security of the residents. Threats can range from a gas or water leak in the home to attempted burglary. By using smart smoke detectors, water leak detector, carbon monoxide detectors and other, a homeowner could receive a notification via their smart phone as soon as the incident happens. Moreover, a door opening sensor or glass breakage sensor could be used to monitor the home when the homeowner is out of town. Finally, the home could be monitored remotely using smart cameras in order to follow up with loved ones or to in conjunction with a burglary.

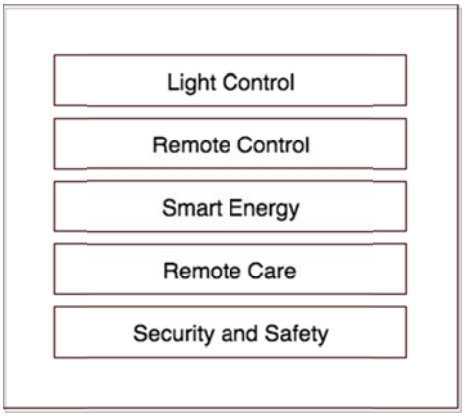


Figure 2-2: Types of home automation services

2.2.1 What is automation?

In this thesis the noun "automation" refers to a programmed action that will be invoked by a home automation service. For example, the homeowner using automation services could be allowed to switch on light when the luminosity in the living room decreases.

2.2.2 Home automation domain actors

As different types of technologies from different service domain enter home the actors that are linked to the home automation sector are not just restricted to few. The following are the few major actors that will be considered in this thesis:

Homeowner — A homeowner is a person who lives in the home where a residential gateway is placed for home automation purposes. All the services provided by the residential gateway have the goal of making the life of homeowner simpler. We will assume that homeowner owns the residential gateway and chooses the different services that it executes. There could be several homeowners within the same home each accessing different services provided by the residential gateway. Additionally, there could be different levels of service usage by different homeowners hence the residential gateway should ensure that there is no unauthorized access to any services.

Gateway manufacturer / Operator — The gateway manufacturer is the company who manufacturers the residential gateway. The main responsibility of the gateway manufacturers is to design and manufacture the residential gateway to support different types of services. We will assume that, some of these services are provided by companies other than gateway manufactures. For this thesis project, we assume that the gateway manufacturer provides the operating system for the gateway and acts as a low level operator for the residential gateway. That is, the gateway manufacturer acts as an operator for different purposes, such as solving gateway related issues, providing value added services, gateway level authentication, etc. Note that the gateway operator might also be different from the

manufacturer, but we have not considered this case in this thesis project.

Service provider A company that provides a service to the homeowner via a software program running in the residential gateway is a service provider. This service provider is assumed to be a domain expert in the type of service it provides to homeowners. For example, a service provider could focus on kitchen automation services, health monitoring services, etc. The service provider could use 3rd party services for developing their services (i.e., by using software produced by another software development company) or they may use a data hosting company. However, in thesis project, we assume that the service provider performs all the activities related to their service.

Appliance manufacturer An appliance manufacturer manufactures different types of home appliances that could be connected to the residential gateway and could be used by the homeowner using one of the services running in the residential gateway. Such a manufacturer could manufacture connected bulbs, smart refrigerators, etc. In many cases, the appliance manufacturer may also act as a service provider and hence provide service directly to the homeowner. As the appliance manufacturer is an expert concerning their device and device level communication with their devices, they should provide appropriate interface programs to the residential gateway in order to enable it to interact with the appliance they manufactured. In this thesis project the appliance manufacturer is assumed to be a service provider of any services concerning the devices that they manufacture.

2.3 Requirements for constructing a home automation system

In order to design or evaluate a residential gateway a few basic and general requirements are necessary. The following subsections give the major requirements for a smart home system as proposed by Hui, Sherratt, and Sánchez [50]. These requirements will be used in the subsequent sections to evaluate existing residential gateway designs and also used to designing and evaluate the residential gateway proposed in this thesis.

2.3.1 Heterogeneity

In order to cope up with the technological advancements and satisfy the needs of the homeowners, supporting heterogeneous is one of the primary requirements for constructing a smart home system. The system should be able to communicate and be compatible with different types of connected devices from different manufacturers. The connected devices could range from a basic lamp (which has operations such as ON and OFF) to home appliances such as a refrigerator with complex operations.

Communicating with different types of home appliances is one type of heterogeneity. The more general vision of this is interoperability. According to Perumal, et al. [51] there are three types of device interoperability:

Basic connectivity	A basic connectivity requirement is the ability to communicate with devices that could be connected via different types of physical connectivity. For example, a home appliance could be connected using wireless connectivity (such as Bluetooth or Wi-Fi) or using wired connections (such as Ethernet).
Network interoperability	Network interoperability focuses more on the communication between different types of networks available in a smart home environment.
Syntactic interoperability	Sitting on top of the other two types of interoperability, is application level interoperability. Syntactic interoperability should ensure proper communication between different types of applications running in the smart home environment. For example, an application controlling the window shades should be able to communicate with the weather application to decide how to adjust the shades. For this thesis project syntactic interoperability was a primary requirement.

2.3.2 Mobility

The activities of a homeowner are not restricted to their home, as the homeowner can move to different places such as an office or shopping mall. Although home appliances are located in the home, the homeowner should have the freedom to control them from outside the home. For example, if the homeowner forgot to switch off a lamp before leaving for their office, he/she should be able to remotely switch off the lamp. By supporting mobility, the homeowner can be virtually present in the home and do most of the tasks just as if he/she were physically present.

2.3.3 Extensibility

Extensibility in terms of a smart home system is the ability of the system to adopt a new technology to extend the services that are provided. The system should not be restricted to one set of devices from one device manufacturer or restricted to one type of service. Extensibility of the smart home system is the first step towards achieving interoperability. One way to achieve extensibility is by developing a modular design. These modules should function independently to provide different type of services at the same time and to provide ways to communicate with other services in order to provide interesting solutions.

2.3.4 Privacy and security

On one side, although there has been a large-scale technological development there are major concerns about security and privacy as side effects of these developments. This is especially true when it comes to smart home systems, regarding security and privacy issues that could still pose a serious threat to homeowners. For example if a microwave application in the home is hacked by a hacker, then an attacker could cause serious damage to the appliance, to the home, or even to the home owner. Also if no privacy protection exists in the smart home, then hackers could easily monitor different types of activities and share personal data with unauthorized parties.

Taking into account the previously mentioned interoperability and extensibility requirements for constructing a smart home system, both security and privacy should be

taken seriously. The system should ensure that the different services and devices in the home should not turn rouge and hence they should protect the homeowners from different types of threats.

2.3.5 Usability

Not all homeowners are technically able to control, configure, or monitor their smart home system. Therefore, the smart home system should be designed in such a way that anyone irrespective of his/her age or technical experience should be able to use it. Even if the system is technically advanced or could provide a very interesting smart home solution, if the homeowners are unable to understand and utilize the service, then the technology is liley to be useless. One way to achieve usability is to provide a good user interface to the homeowner where they can control their smart home. The major usability aspects specified by Moeller, et al. [52] are consistency, transparency, and personalization. All of these should be considered when planning for usability.

2.3.6 Context awareness

In 1994, Schilit, Adams, and Want [53], introduced a system that could examine the user's activities and react accordingly. This system was the first to use the user's context (such as proximity) to determine what kind of actions to perform. These reactive actions could be incorporated using a simple condition of the form: IF<<Condition>> THEN<<Action>>. For example, using the location context information of the homeowner, a context aware system could switch OFF all the lamps in the home when the homeowner is not in or near the house. Moreover, context is not restricted to location, but could include who is close to the user, what types of devices are in the proximity of the user, information from a connected device, and more. By collecting different types of context information about the user, a smart home system could provide a more personalized service. As a context-aware solution might be seen as privacy invasive, the smart home system should find an appropriate balance between privacy and suitably adaptive services.

2.4 Samsung SmartThings Hub

Samsung's SmartThings [54] home monitoring kit is a hub-based system used to provide home automation and security services. Starting from controlling a light bulb to checking whether there is any motion detected the home, the system provides varied services for homeowners. Using an application installed in a smart phone a homeowner can control devices, receive notification from devices, or even set up automation programs (For example, switching OFF all the lights when the main door is locked). Compared to other home automation hubs, the Smart things hub can communicate with close to 200 different smart devices. The SmartThings environment contains four primary components:

- SmartHub,
- SmartThings Devices,
- SmartThings cloud, and
- SmartApps.

2.4.1 SmartHub

The SmartHub is the central component in Samsung's SmartThings environment. The SmartHub acts as a bridge between the devices and SmartThings cloud. The SmartHub can be connected with devices that support ZigBee and Z-Wave technologies (such as GE's Z-Wave power sockets or in-wall switches). In addition, this hub can also be connected to devices that support IP based protocols, such as smart devices connected via Wi-Fi. In terms of future improvements, the hub is expected to host Bluetooth low energy support – however, this is not yet activated. The physical interfaces to this hub are shown in Figure 2-3.

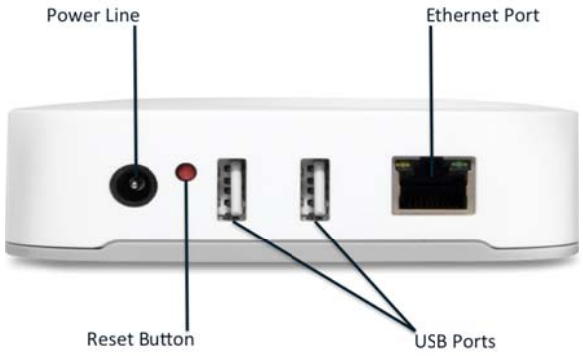


Figure 2-3: Samsung SmartHub

The hub receives power from an AC to DC power supply and it also supports battery-powered operations. Thus the hub could function using the batteries when there is a lack of mains power and could even autonomously execute certain automation functions. Finally, the hub is connected to the SmartThings cloud via a broadband Internet connection using the dedicated Ethernet port on the rear side of the hub (as shown in Figure 2-3.).

2.4.2 SmartThings Devices

Using a SmartThings system a homeowner can connect several varieties of smart devices and appliances in their home. These devices include, devices from the same manufacturer for example SmartThings motion sensor or SmartThings dimming outlet. The SmartThings environment also supports devices from different manufacturers that can be connected using LAN, cloud-to-cloud integrations, or devices that support ZigBee and Z-Wave technologies. A SmartThings Hub is used when a device uses ZigBee, Z-Wave, or LAN to communicate. When the device supports cloud-to-cloud integration, the SmartThings cloud communicates with the device manufacturer's cloud using a special Application Programming Interface (API).

2.4.3 SmartThings cloud

The SmartThings cloud hosts all the services that are provided using the hub and the devices. All the information necessary about the devices, automations (i.e., scripts), and configurations are stored in the SmartThings cloud. In order to view or make changes to their home environment a homeowner uses the SmartThings mobile client application running on an IOS or Android device. As the system follows a "Cloud first" [55] approach, the hub should always be connected to the Internet in order to connect to the smart devices, to perform automation, to view the current state of the home, etc. As said before, a

battery-powered hub could be used to provide a few automation features without being connected to the Internet.

2.4.4 SmartApps

In order to perform device control or automation in a SmartThings environment, one must use SmartApps that are installed in their mobile client application. These SmartApps can be downloaded from Samsung's market place. SmartApps are classified into three main categories depending upon their usage. These categories are:

Event Handlers	Event Handler applications wait for a certain event to happen and then execute respective actions or handlers. Waiting for the event is done using a subscription method.
Solutions module	Solutions module application makes the combination of two or more SmartApps possible that work together to provide a solution. For example: Lights and Switches.
Service manager	Service manager applications are used for connecting the hub to the appliances via a LAN or the cloud.

2.4.5 Architecture

The SmartThings system tries to separate the intelligence from the device and provides basic devices features such as ON/OFF or open/close. Higher layers of the architecture manage the intelligence for the device. By doing this Samsung claims that it reduces maintenance complexity, lowers the device's cost (as the devices are only supposed to do basic activities), and increases battery lifetime (for battery powered devices) [55].

In the SmartThings architecture (shown in Figure 2-4), the device handler acts as an interface between the SmartApps and the devices. The device handler translates device specific messages to normalized SmartThings events and vice versa. This means that there will be many different device handlers when connecting different types of smart devices.

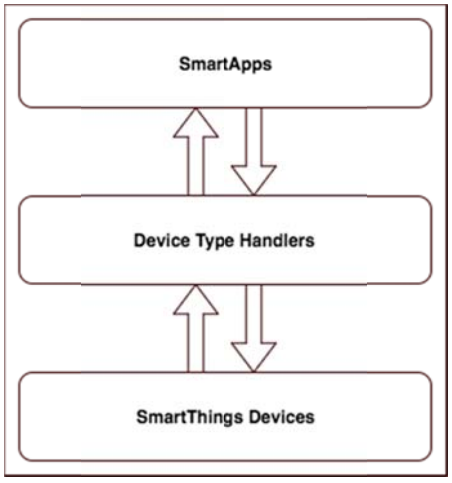


Figure 2-4: Samsung SmartThings high level architecture

To standardize the communication with the devices, each device is defined using "capabilities". These "capabilities" standardizes the attributes of a device for example the

state of the device (such as ON/OFF) or the commands that are used to execute tasks using the devices. Each device in a SmartThings environment will have:

Commands The actions that could be performed using the device.

Attributes The value from the device.

The SmartThings internal development team is responsible for creating these "capabilities" and maintaining them. This means that for any new type of device to get connected with the SmartThings system, the internal development team should create the "capabilities".

2.4.6 Automation Management

Listening to the events stream is one of the primary activities that a SmartApp should perform. Whenever an event has occurred it will be published in the event. For example, an event could be switching on the thermostat or an intrusion is detected. When a SmartApp listens to the event stream and finds a relevant event, then the associated action is executed. Automation using SmartApps could also be triggered by external sources or a scheduled event.

2.4.7 Evaluation

The following subsection detail the evaluation of the Samsung SmartThings system according to the home automation requirements given in Section 2.3.

2.4.7.1 Heterogeneity

Samsung's SmartThings hub with the help of its complete home automation system provides a good level of heterogeneity. A SmartApp that controls the heating system in the home will process temperature events to adjust the room's temperature [56]. In terms of service level heterogeneity, the SmartThings system can collect information from external services in order to collect information such as weather or CO2 levels to inform the homeowner using a notification message. As mentioned before, the smart home with a hub can connect to more than 200 devices via different connection methods.

2.4.7.2 Mobility

Using the SmartThings client mobile application, the homeowner can control devices located in their home. As all the control and automation information about the smart home system are stored in the SmartThings cloud, these client applications simply connect to the cloud and send a request, later the cloud transfers the request to the hub placed in the home. There is no direct communication between the client application and the hub as the communication is always via the cloud.

2.4.7.3 Extensibility

A homeowner can browse through the list of existing applications in the SmartThings application store and install the selected app to provide different types of service to their existing smart home system. In this way the system guarantees service level extensibility.

2.4.7.4 Security and Privacy

Each SmartThings hub has a unique key that is used for mutual authentication and encryption, thus preventing malicious attacks. All communication between the cloud and the hub is encrypted using SSL.

Each user in the home using the same SmartThings system needs to create an account in the SmartThings cloud in order to access services. All the homeowners who are logged in to the SmartThings cloud have complete access to the devices and the services. This means, that a child in the family will have the same access rights for some appliances as elder members of the household. Furthermore, there is no SmartApp level access control that defines which SmartApp can access what type of feature of a device.

When a SmartApp requests to control a device, the system gives access to the complete set of actions that could be carried out with the device. According to a study to analyse security threats in the SmartThings environment conducted by Fernandes, Jung, and Prakash [57] of all the available SmartApps, around 42% of SmartApps were given more privileges to access a device's capabilities than what they requested and ~68 SmartApps already exploited this flaw.

Automation management using event subscription allows the SmartApps to listen to all the types of notification coming from the device they have subscribed to. Moreover, a SmartApp can listen to notifications from a device they are not supposed to listen to. Additionally, a SmartApp can fake a notification and send spoofed events to notify other SmartApps causing serious security threats. Unfortunately, when using this automation management system *without* verifying the integrity of the origin of the event, both other automation and devices can be exploited.

2.4.7.5 Usability

The client mobile application from SmartThings is the primary link between the homeowner and the system. This application also provides a means to view the top-level status of the home using a dashboard view. All the services offered by the home system are accessed using a single application and using one single sign on. Moreover, the application does not adapt itself depending upon on who is using the application, hence it does not provide personalized service.

2.4.7.6 Context-aware

The system provides geographical location based contextual information such as if the devices are placed in a home or in an office. Devices in each location can be clustered into groups depending upon on the room they are in or their physical location. This enables request to be given at the group level or the location level. For example, a SmartApp could switch ON all the appliances in a group "living room" if there is any motion detected in the room.

2.5 INSTEON

In 2005, Smartlabs introduced their Insteon home automation system to overcome the weaknesses posed by other systems. Insteon focused on providing responsive, reliable, easy to install, simple to use, and affordable home automation for regular homeowners. This automation system uses the trademark INSTEON™ standard for advanced services [58]. INSTEON™ technology is a peer-to-peer dual band mesh topology where the smart

devices in the system communicate via power line, radio frequency (RF), or both at the same time. It is possible to connect up to 1024 smart Insteon devices to a single home automation network. Furthermore, according to Insteon the more devices that are connected the network the more robust the network becomes.

In order to provide smart home services, INSTEON system the following components:

2.5.1 INSTEON Hub

The INSTEON hub (shown in Figure 2-5) acts as a bridge between the connected appliances and the homeowner. This hub enables the homeowner to perform basic actions such as configuring advanced settings for the connected devices using the INSTEON mobile application installed in a smartphone or tablet. The hub connects with other home appliances via wireless radio frequency or via the wired power line.

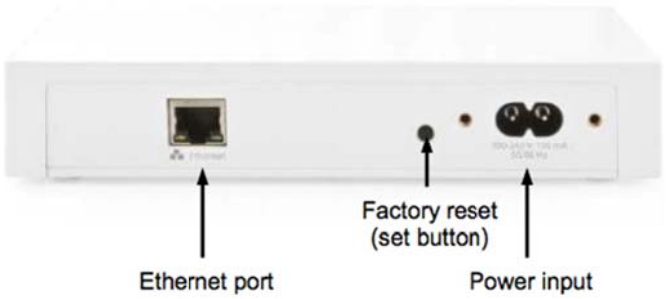


Figure 2-5: INSTEON hub

The homeowner connects the INSTEON hub to their existing network router using an Ethernet cable. Via this connection the can request and receive an IP address and subsequently can connect to the cloud. The power input socket is used to power up the hub as well as to communicate with other INSTEON devices that are connected using power lines. Compared to the Samsung SmartThings hub, this hub does not rely completely on cloud. To control or configure home devices, the INSTEON mobile application communicates with the hub using a HTTP service running in the hub. The hub stores all the necessary information for performing automation in itself and communicates with the cloud only if necessary, such as for authentication or sending notifications

2.5.2 INSTEON Devices

Many devices manufactured by INSTEON can be used in home without a hub. Using the standard INSTEON™ technology a lamp and a dimmer could communicate directly with each other enabling the homeowner to control the lamp using the dimmer. The dimmer automatically finds a compatible lamp connected in the home and establishes a link between them with the help of the homeowner. By using the INSTEON application, the capabilities of a device could be fully utilized and the service offered in the home can be improved.

In order to connect 3rd party devices to a smart home using INSTEON hub, cloud-to-cloud based connection should be established. Thus when the homeowner controls the 3rd party device using the INSTEON mobile application, the request is sent to the INSTEON

cloud, then the INSTEON cloud sends the request using the provided API to the 3rd part device cloud, and finally the request will be sent to the device.

2.5.3 INSTEON Application

A homeowner downloads the INSTEON application to a tablet or smart phone to connect with their hub (or as noted above even directly to devices). One of the primary uses of the application is to authenticate the user. In order to use the hub via the application, the homeowner should first authenticate using valid credentials. The application sends these credentials to the INSTEON cloud to verify whether the person trying to connect to it is a valid user of this hub.

Using this application the homeowner can see the list of devices connected to the hub along with their status. Using the application the user can also control the devices. The interface for controlling each device varies depending upon the type of service the device provides. For example, a lamp interface will have an ON/OFF button, while a dimmer interface will have a dial to set the intensity.

2.5.4 Automation Architecture

Compared to the SmartThings architecture, the INSTEON architecture is simpler as it uses only preparatory application and devices. The INSTEON hub can provide the following types of automation:

Schedules The homeowner can configure a time-based automation for each device. For example the homeowner could configure an automation to switch on a light at 7:00 AM every morning. The schedules can be configured for a specific time, specific days, a part of day (morning, afternoon, or night), or sunset/sunrise.

Scenes The homeowner can configure a set a devices in their home with specific values and define this configuration as a scene. This scene can be activate manually using the application of based upon specific triggers from sensors. For example, a “Good Evening” scene could be configured with all of the lights switched ON and this scene could be linked with sensor placed to measure the light intensity of light coming into the house. In this way, if there is exterior darkness, the “Good Evening” scene can be trigger and all of the light turned on automatically.

2.5.5 Evaluation

The following subsections detail the evaluation of the INSTEON system according to the home automation requirements given in Section 2.3

2.5.5.1 Heterogeneity

When it comes to device level heterogeneity Insteon’s smart home system supports devices manufactured by them and a few other 3rd party devices. Different devices can communicate directly with each other to provide automation solutions, such as a motion sensor communicating with a light bulb. As the service provided by the system is restricted

to home automation communication between different types of other services is not applicable for INSTEON.

2.5.5.2 Mobility

Using the mobile application the homeowner can control the connected devices in their home when he/she is in the home or even outside the home. A homeowner using the remote access feature of the hub can access the hub from outside of the home by configuring their home router to do port forwarding. There is a detailed user guide that describes how to setup the application and the router to activate this feature.

2.5.5.3 Extensibility

As said earlier, in order to add a new feature to the system, INSTEON must either release a new variety of device or upgrade their existing application so that the homeowner can utilize the new service. Unlike SmartThings, INSTEON does not have a SmartApps store that provides different types of service, but it greatly relies on a single integrated application.

When it comes to the extensibility of service provided by Insteon system this system is restricted to home automation. Therefore, services such as Telecare cannot be installed in the existing system to function in the hub. However, INSTEON provides APIs to other companies to integrate 3rd party solutions, but the level of integration is restricted to viewing the status of the device and cannot be used for controlling these devices.

2.5.5.4 Security and Privacy

Each homeowner must create an account in the INSTEON cloud in order to access the services provided by the hub. The homeowner authenticates using their login credentials via the application in order to access the hub. This system also suffers from the same issue as the SmartThings system, as it does not differentiate based upon the user controlling the application.

Communication between the INSTEON devices and the hub is secured by encryption, but the communication between the mobile application and the hub is insecure. Using software such as Wireshark [59] the HTTP traffic between the mobile application and the application can be eavesdropped.

According to an exposed vulnerability [60], a remote unauthenticated hacker can access the hub using the web interface. In this way the hacker can gain complete control of the hub and all of the devices connected to it. By exploiting this vulnerability, a hacker could switch ON/OFF home appliances, change the temperature, or even lock/unlock the doors of the home.

2.5.5.5 Usability

INSTEON advises the homeowner to use the mobile application to connect with the hub, but it is also possible to connect using a web browser. Using the mobile application, the homeowner can take control of the system and configure it.

Similar to SmartThings, there is no personalization offered by the INSTEON mobile application, hence a child with an INSTEON credential has the same level of access as an adult.

2.5.5.6 Context-aware

The INSTEON system does not provide any contextual information.

2.6 Qualcomm 2net system

Qualcomm introduced the 2net system to provide patient care in a home with connected medical devices. Using this system, a patient can capture vital signs and make this data available for interested parties such as doctors, hospitals, or care givers. Enabling patients to return to their homes and remotely monitoring their health has a good impact on reducing hospital readmission. This impact is positive in that it reduces the cost for the hospital in comparison with the cost of having the patient in the hospital's own premises [61]. The Qualcomm 2net platform is a good example of how to provide a remote patient monitoring service using a hub. The 2net environment has three components: the 2net Hub, 2net Platform, and 2net Application.

2.6.1 2net Hub

The Qualcomm 2net Hub is placed in the patient's home. This hub is a simple device providing plug and play type communication with the patient's medical devices. One of the main advantages of this hub is that, there is no need for using an external application to take a measurement. A patient simply switches on the connected medical device and takes a vital sign measurement. The 2net hub will establish a connection with the medical device and then collect the vital sign measurement and transmits this data to the 2net platform cloud.

For example, a wireless blood glucose monitor, such as Entra Health Systems' MyGlucoHealth, is a stand-alone medical device that measures the patient's glucose level. The device displays the result of the measurement on a small screen on the device. In addition, it transmits this value to interested parties using Bluetooth. The 2net hub takes advantage of this Bluetooth connection to receives the glucose measure value every time a patient measures their glucose level [62].

The 2net hub is small and has a power plug just behind it, hence the hub can easily be plugged into a convenient wall socket. In order to connect and transfer the medical data to the 2net cloud, the hub uses a cellular connection. This means that the hub must be placed in the home where there is cellular coverage. The hub uses various access technologies such as Bluetooth, Bluetooth smart, Wi-Fi, or a wired USB connection to communicate with the medical devices. Two notification lights on the hub (see Figure 2-6) are used to indicate that the hub is connected to the cloud and whether the data is collected and sent to the cloud. By viewing these lights the patient can learn that their measurements have been taken and transferred properly.

Figure 2-6: 2net hub – the connectivity is shown via the indicator on the upper left and the successful transfer of data using the indicator on the upper right.

2.6.2 2net Platform

Similar to the SmartThings system, the 2net system also follows cloud-first approach. Once the 2net hub collects vital sign measurements from a medical device, the hub transfers these measurements to the 2net platform cloud. The 2net platform cloud then securely stores the vital sign measurements under the patient's account. Using secure connections with the 2net platform interested parties such as doctors or hospitals can view these measurements or asynchronously receive notifications when the measurement lies outside of a specified range. The hub to cloud and cloud to application communications are encrypted in order to protected the patient's privacy.

This Health Insurance Portability and Accountability Act of 1996 (HIPPA) approved cloud platform supports different types of E-health use cases such as remote patient monitoring, connected medical management, consumer engagement, etc.

2.6.3 2net Application

The 2net application has two primary functions in the 2net system. The first function is to provide a patient visualization tool by which the patient can see their history of medical measurements. To produce this visualization the application securely connects with the 2net cloud to collect to patient's historical medical data. The second function is to connect with any of the medical devices that the 2net hub could *not* connect with. Using this function the patient can collect vital sign measurements from a connected medical device and then the application sends this data securely to the 2net platform cloud.

2.6.4 Evaluation

The following subsection detail the evaluation of the 2net system according to the home automation requirements given in Section 2.3.

2.6.4.1 Heterogeneity

With regard to device level heterogeneity, with the help of 2net hub, patients can use many types of medical devices that support data connectivity via wired or wireless connections. In order to connect additional devices to be supported by the 2net hub the hub's firmware should be updated.

When it comes to service level heterogeneity, the 2net hub is limited to Telecare services. As a result, this hub does not support home automation services and communication between the Telecare and home automation service is not possible.

2.6.4.2 Mobility

In order for a patient to measure their vital signs when using the medical device proximity to 2net hub is necessary. This is necessary because the medical device must be within range of the 2net hub. To access and visualize prior measurements, a patient can use the 2net application from anywhere with Internet connectivity.

2.6.4.3 Extensibility

The service offered by the 2net hub and its system is restricted to Telecare services and the hub does not provide any communication with home automation services.

2.6.4.4 Security and Privacy

As the 2net system is providing Telecare services as part of patient home care and this involves medical data, the system must follows strict standards to provide security and privacy for the patient and their data. The system provides two levels of security: one for data communication and another for data storage. All communication between the medical device and the 2net hub to collect the vital sign from the medical device is encrypted. Additionally, the communication between the 2net hub and 2net platform cloud over a cellular connection is also encrypted end-to-end.

In order to store the medical data, the system uses the 2net cloud platform that is HIPAA approved for patient data storage in order to protect the patient's privacy and their data.

2.6.4.5 Usability

Using medical device that uses complex connection mechanisms could be complicated for a patient in their home. Therefore all of this complexity is hidden from the patient. The patient simply takes a vital sign measurement using the medical device and then the 2net hub automatically detects that there is new data and it retrieves the measurement. In this way the patient does not need to worry about the connection mechanism and this avoids and extra stress for them. However, the hub only supports one user at a time; thus, if there are more than one resident in a home, then multiple hubs must be used.

2.6.4.6 Context-aware

There is no context-aware method established in the 2net system.

2.7 Homer

To bring different services such as home automation and Telecare services under one roof, Claire Maternaghan proposed a policy based smart home design called Homer [63]. Using this design a homeowner can control the connected devices in their home and write policies (user defined rules) to add an automation service using these devices. Homer provides developers with the functionality needed to develop and install custom services with the connected devices and to enable the homeowners to use the resulting service. The core framework and different services offered by the design are implemented using OSGi to provide modularity. With the help of OSGi these services can run independently, while at the same time they can communicate via predefined interfaces. Unlike the previously discussed devices, Homer is not a commercially available home automation system.

The overall Homer architecture is shown in Figure 2-7. There are three primary entities in the homer design and these are described in the following subsections.

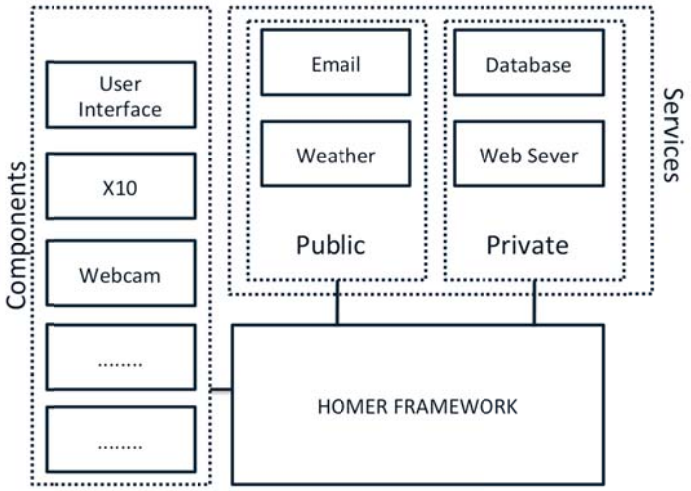


Figure 2-7: Homer Architecture

2.7.1 Homer Components

Components in Homer act as an interface for the residential gateway to communicate with the connected devices within the home and the user services that are running in the home. An example, of such a service is the weather service shown in Figure 2-7. By taking advantage of the modularity of OSGi, the homeowner can any time install or remove a connected device simply by adding or removing the component responsible for the device.

The main tasks of the component responsible for each device is to translate the request coming from the system to device understandable commands and to collect information from the device and then convert these into system compliant commands and data. There can be multiple device components each with different functionality running in the residential gateway at the same time. These components can communicate with each other via defined interfaces. For example, there could components responsible for a medical device and a home appliance running in the same residential gateway. Additionally, there can be components responsible for non-device related services such as weather, email, etc. Developers can write components to collect information from external services and pass this data to other components within the system.

2.7.2 Services

Services are very similar to components in Homer; the only difference is that the services do *not* directly communicate with the user or devices. Other components use the services in the system to provide their intended service. Some of the different types of general services include a web server, Homer database service, logger service, etc. Using the web server service, a homeowner can access the functionality offered by the residential gateway. The services in turn can be classified into private and public services depending upon whether the Homer components can directly access the service or not. For example, the logger service in the Homer framework is access by all the components, this it is a public service., while the web server service can only be accessed via the Homer framework so it private service.

2.7.3 Framework

The Homer framework encapsulates all the core functionalities and act as the brain of the system. This framework contains:

Component and System Bridge	The framework to interact with other parts of the system uses these bridges.
Policy server	The policy server stores all the home automation policies.
Event hub	The module that is responsible for handling events in the system. This hub is responsible for receiving event-based information (movement sensed by a motion sensor) from the device components and carries out respective policy that is store in the policy server.

2.7.4 Automation Architecture

To add automation policies (for example, if motion is detected by the motion sensor in the room, then a lamp in the room should be switched ON) the user interface component in the architecture is used. Once the user enters the automation policy, the policy will be stored in the policy server and the event hub will wait for an event notification from the motion sensor to trigger the automation policy. Once the motion sensor detects motion, then the component responsible for the motion sensor sends an event notification to the event hub in the homer framework. Once the event hub receives this notification, it checks for a corresponding policy and executes the corresponding action. In this case, the event hub passes the request switch ON to the component responsible for the lamp.

Each policy in a Homer system is expressed using simple key words such as WHEN, IF, and THEN. To combine multiple conditions or multiple actions, the terms AND and OR can be used to express the policy. For example: WHEN there is a motioned detected in the home, IF the light bulb is OFF, THEN switch ON the light bulb. A similar policy could also be written for a Telecare services. For example, WHEN there is a fall detected, IF the heart rate is more than 90 BPM(Beats per minute), AND IF skin temperature is greater than 38°C, THEN send a notification to (the designated) doctor AND send a notification to (the designated) caregivers.

2.7.5 Evaluation

The following subsection details the evaluation of the Homer system according to the home automation requirements given in Section 2.3.

2.7.5.1 Heterogeneity

The design of Homer makes it possible for different types of smart home services to communicate with each other and can be used to produce an interesting solution [64]. As Homer is a policy based home automation system that supports a variety of devices, a homeowner can write heterogeneous policies to be executed in their home. For example, homeowner could define a policy to adapt the temperature level according to the homeowner's measured skin temperature. As Telecare automation involves sensitive data, therefore appropriate care should be taken when writing a policy. The Homer system also allows a caregiver or medical professional to configure service specific policies and push these policies to the system for use with their patients. In addition to policy descriptions, the Homer system also provides the possibility for the developers to write components that provide heterogeneous services. All together the Homer system provides both service level and device level heterogeneity.

2.7.5.2 Mobility

Homer provides the user with mobile as the user can access the system via a smart phone or a web-based interface. This system provides a HTTP based API to allow homeowners to control and configure their home. Additionally, any client entity that can make HTTP calls with JSON objects can communicate with the Homer system.

2.7.5.3 Extensibility

The Homer system satisfies the extensibility requirement by providing sufficient flexibility for developers to write components to add different types of smart home services with or without devices. In order to add a new device, the homeowner simply adds a new device component (in the form of a OSGi bundle) and then they can start using the device in their home *without* modifying any other component. This applies even for non-device based components such as email and twitter. A developer can develop various services using the OSGi framework and later a homeowner can add this service to their residential gateway.

2.7.5.4 Security and Privacy

The focus of Homer is primarily to implement an open and heterogeneous automation system to address both security and privacy related issues. However, the only level of security is based upon authorizing an application using an application key and a secret key.

2.7.5.5 Usability

Using the IOS based application designed by Claire Maternaghan, a homeowner can view the list of connected devices in their home and view other non-device based services such as receiving a live twitter feed. Using the same application the homeowner can add automation polices to their home and view them any time.

2.7.5.6 Context-aware

The Homer system uses basic location-based context information and each device is tagged with location-based information. For example, the location “living room” could contain lamps, coffee machine, etc. Using one high-level automation command such as “Switch OFF all appliances in living room” the context information is used to apply this command to of the appliances in the living room.

2.8 Virtualized service gateway

In order to run services from different service providers in a same residential gateway concurrently, Yvan Royon and Stephane Frenot [65] proposed a service oriented gateway design using virtualization techniques together with OSGi technology. This architecture is shown in Figure 2-8. Each service* offered by a service provider for the smart home is executed in a virtual service gateway sharing a single JVM in the residential gateway. For example, a home automation service will be executed in “virtual service gateway 1”, while the Telecare service will be executed in “virtual service gateway 2”. Due to this isolation the entities relating to the virtual service gateway can only see its own resources and services. Even though the virtual gateways are isolated an inter-virtual gateway communication system makes it possible to pass data (by reference) between services.

To make a resource available for the virtual service gateways running in the residential gateway the system depends on the software element called the core service gateway. The core service gateway makes available resource such as hard disk storage, network bandwidth, CPU cycles, etc. The core service gateway service is also responsible for sharing certain common services such as the HTTP server or logger with the different virtual service gateways.

Royon and Frenot even define who has what level of access and responsibility in the residential gateway, hence the service provider is responsible for maintaining their own virtual service gateway and the residential gateway provider is responsibility for maintaining the core service gateway related services. Finally, the gateway operator does not have any access to the services running in the virtual service gateways.

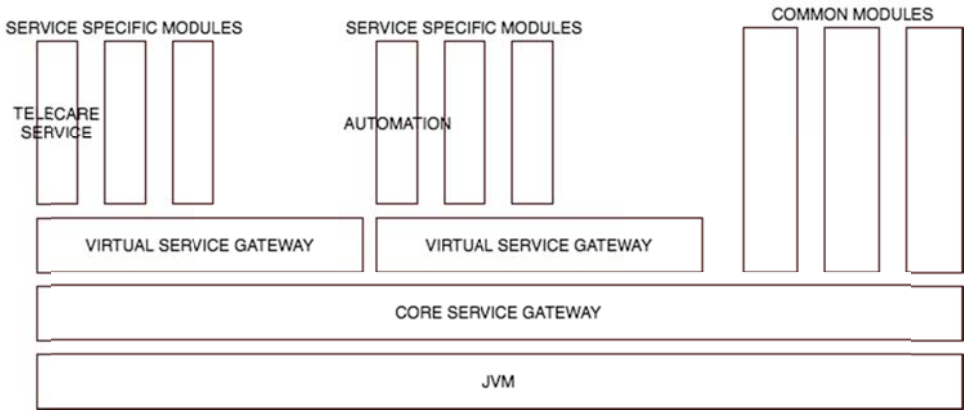


Figure 2-8: Virtualized service gateway architecture

* A service can be a single module or a combination of modules)

2.8.1 Evaluation

The following subsections detail the evaluation of the virtualized residential gateway system according to the home automation requirements given in Section 2.3.

2.8.1.1 Heterogeneity

The proposed virtualized residential gateway design enables different types of connected devices to be used in the smart home and the core service gateway ensures that the correct device is available only to the correct virtual service gateway. The proposed design can accommodate different types of service, such as home automation and Telecare services – with each running in its own virtual service. As mentioned before, these services can communicate by passing references thereby providing interoperability between the services.

2.8.1.2 Mobility

The design allows the service provider to install an HTTP based web server in a virtual service gateway, then the homeowner can communicate directly with this service. If each virtual service gateway installs a webserver, then the homeowner can access each of these individual services.

2.8.1.3 Extensibility

One of the primary goals of the design is to extend the functionality of a residential gateway in order to provide different types of services. Using an experimental setup Royon and Frenot were able to execute 10 different virtual services in a single simple residential gateway.

2.8.1.4 Security and Privacy

There are two types of isolation provided by this design. The first is isolation between the different service providers' services running in the residential gateway. Separating the core gateway services from the service providers' services provides the second type of isolation. In this was the functionalities of the service providers and the gateway providers are each restricted.

Unlike Homer, Royon and Frenot propose a method for securely downloading and installing services in the residential gateway. Their proposed method uses cryptographic techniques to solve the problem of authenticating a service provider and checking the integrity of the downloaded service module. However, they considered security features such as user authentication and secure remote management as future work.

2.8.1.5 Usability

Royon and Frenot do not give details regarding the user interface in their design. One thing that can be inferred is that the homeowner can have a separate user interface for each service running in the gateway.

2.8.1.6 Context-aware

The design does not support any context aware features.

2.9 Summary

The first part of this chapter explained in detailed the different types of HAN technologies together with related concepts (including service discovery and remote management). The second part of this chapter described the home automation concept in detail together with stating the requirements for a home automation system. In the final part of the chapter, the functional architecture of five different residential gateways were explained with a brief analysis of each with respect to the home automation requirements.

2.9.1 Design comparison

Table 2-1 summarizes the list of non-functional requirements discussed earlier in this chapter.

Table 2-1: Summary of the five systems that were examined in this chapter

	Samsung SmartThings	Insteon	2net Hub	Homer	Virtualized Residential Gateway
Heterogeneity	No	No	No	Yes	Yes
Mobility	Yes	Yes	Yes	Partially implemented	Partially implemented
Extensibility	No	No	No	Yes	Yes
Security and privacy	Authorization issues	Access issues	No known issues	No known issues	No known issues
Usability	Using mobile Application	Using mobile application	Using mobile application	Using mobile application	Using mobile application
Context aware	Location based Context	No	No	Location Based Context	No

2.9.2 Summary of use cases

Other than the non-functional requirements described earlier in this chapter the following use cases were selected to understand the *functional* requirements of a residential gateway. These general use cases give an overall idea of the functions that a residential gateway could perform.

Use case 1: In home or remote control of devices

A homeowner should be able to control devices when in the home or from a remote location. Examples of such control include:

- Switch on the heater
- View images from a security camera

Use case 2: Home Automation

An adult homeowner should be able to configure automation rules using connected devices to run autonomously. Examples of such rules are:

- Run dishwasher and washing machine during night-time to reduce peak hours energy consumption
- Adapt the shades and lights depending on the natural light.

Use case 3: A doctor can monitor their patient in the patient's own home

It should be possible for a designated physician to monitor a patient when they are in the patient's own home. Examples of such monitoring include:

- The patient should be able to take vital sign measurements and the doctor should be able to view the results of these measurements
- The doctor should be notified if the patient falls

3 Methodology

This chapter gives an overview of the method of choosing the design for each of the individual components in a residential gateway. Each section explains why the specific method or technology was chosen with an explanation of its advantages. This chapter contains section explaining the use of a Service Oriented Architecture based OSGi framework, access control mechanisms, remote access methods, and a method for automation execution.

3.1 Service Oriented Architecture

Service Oriented Architecture (SOA) is an enhanced version of distributed computing for designing modular and flexible systems. According to the World Wide Web consortium (WC3), the definition of a SOA is "*A set of components which can be invoked, and whose interface descriptions can be published and discovered*" [66]. SOA provides a logical method for modularizing a software system into components or services. An end user using a client application or other services can access the services using discoverable interfaces.

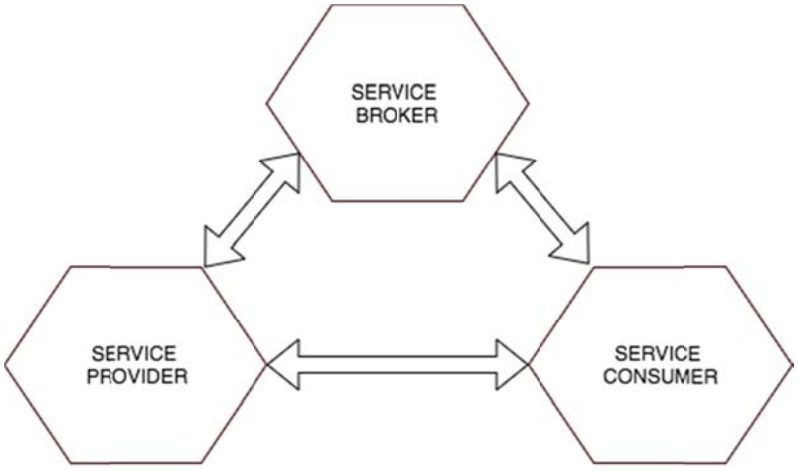


Figure 3-1: SOA primary entities

In SOA based design, there are three primary entities (see Figure 3-1) that interact with each other [67]:

Service consumer	The entity in the system that requests a service and gets serviced. In the context of this thesis, the service consumer could be a homeowner or another service running in the home.
Service provider	A service provider in the system is a module that provides service. This module could be a class or node or a program or combination of them that provides a specific type of service to the service consumer.
Service broker	All of the services in the system are registered with the service broker. This enables a service consumer to search for a service by utilizing the service broker to find available services in the system.

3.1.1 Benefits of SOA

Dynamism and substitutability are the main advantages of SOA [68]. A system based on SOA provides the ability for the service provider to provide or withdraw a service at any time and for the service consumer SOA provides the ability to bind at will with an available service. Substitutability arises because the service descriptions are represented as contracts and any service that follows the contract could be substituted for another service that provides the same service – even though a different service provider provides it. Using the service-oriented approach for designing a residential gateway will offer a flexible infrastructure, which can easily adapt to different users' requirements.

3.1.2 SOA using OSGi

The OSGi alliance was founded in 1999 by different types of industrial stakeholders, such as home automation companies, Internet service providers, and consumer electronic companies (such as Philips, Panasonic, Motorola, IBM, etc.). The main agenda for the alliance is to develop a framework that can be used for building modular dynamic applications. According to Parks Associates, the Open Service Gateway Initiative is the most recent effort that tries to support different IP-based services deployed in the home. Unlike other technologies OSGi targets a standard platform that will be used for appliance connectivity via both LANs and WANs. The OSGi framework provides a good platform for Java developers to develop complete runtime modularity and reusable programs [69].

The characteristics of a SOA based system are inherited by OSGi based systems. For example, the service provider can add a service or remove a service anytime without disrupting other services running in the system. Additionally, OSGi enables the service providers to provide services that can communicate with each other and provide interoperable solutions. OSGi is a Java based modular system. This system can allow different types of services from different providers to run in one residential gateway. The OSGi framework provides a lightweight and simple base for creating service-oriented applications [68]. The OSGi framework defines four different layers to manage the services running in the residential gateway, namely, services, life cycle, modules and execution environment. Figure 3-2 displays the different layers in OSGi.

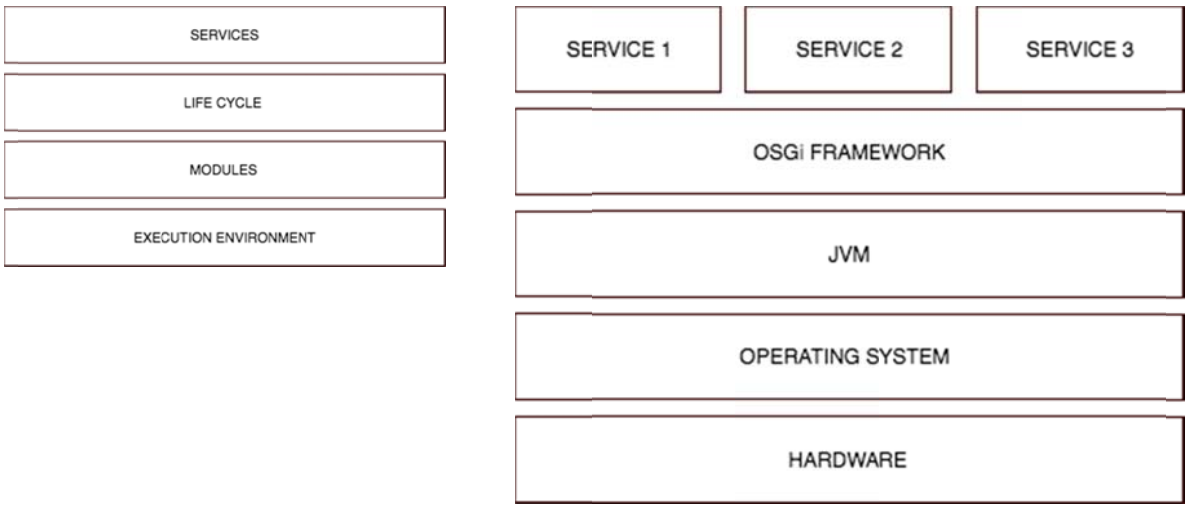


Figure 3-2: OSGi Layers

3.1.2.1 Services layer

The service providers use the service layer for registering (in a service registry) the service they provide in the residential gateway. Subsequently, the service consumers use this layer to collect information about the services available in the gateway.

3.1.2.2 Life cycle layer

The lifecycle layer defines how the service modules can be installed, uninstalled, started, and stopped dynamically in the OSGi framework.

3.1.2.3 Module layer

The concept of modularity provided by OSGi is defined the module layer. The service provider creates service modules or bundles so the homeowner can install them in the residential gateway in order to utilize the service. The module layer provides private classes for controlled linking between the different service bundles.

3.1.2.4 Execution layer

The execution layer provides the specification details of the Java environment using which all the bundles will be executed. The specification details include the minimum requirements that are necessary for the bundles to run in the gateway.

The fundamental component of any design with the OSGi framework will be constructed using bundles and services. Figure 3-3 shows a pictorial representation of the bundles and services.

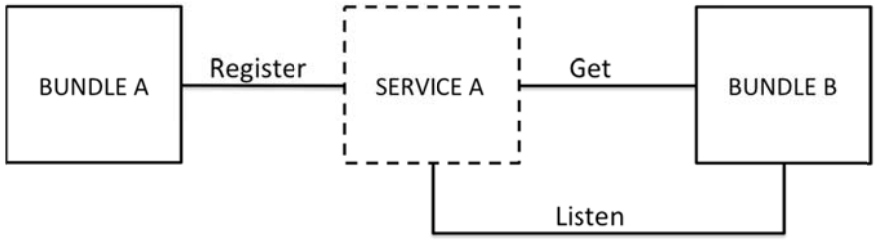


Figure 3-3: OSGi bundles and services

3.1.2.5 Bundles

The modules in the OSGi framework are called bundles and developers develop (service providers) with them. Bundles are a collection of Java classes, resources, and a manifest file. Each bundle contains metadata, such as name, version, import/export details, minimum supported Java version details, and some human readable information. There are different types of bundles, for this thesis project, we focus on application bundles [70] as these will be used for creating vendor applications.

3.1.2.6 Service

Services are the OSGi component that dynamically connects bundles and act as an interface between bundles. A bundle can listen for or register for a specific service and use the service to implement an action. As the services are dynamic, a bundle could anytime

terminate its connection to the service and the other bundles that are using the service could continue running [71].

3.1.2.7 Benefits for using OSGi

According to Marples and Kriens the advantages of using OSGi in designing a residential gateway are [72]:

Platform Independence — As the OSGi framework is based on Java, a system based on OSGi can be implemented in different kinds of hardware environments and on top of different operating systems. This platform independence can be extended to include the independence when communicating with different types of connected devices and supporting different types of LAN technologies. As the OSGi framework provides different types of device access APIs, a homeowner can communicate with a variety of connected devices. The OSGi specification provides the possibility to support different types of LAN technologies including wired and wireless connectivity [73].

Application Independence — OSGi achieves application independence by providing implementation APIs that are common for different applications from different markets. OSGi is not restricted to home automation type of services, it can also be extended to other service markets such as Telecare, automobile, etc.

Service collaboration — An OSGi based residential gateway could host more than one type of service. In this case difference service can discover and communicate with other services running in the same gateway. The interface between services could be defined to standardize service collaboration. This standardization would enable the homeowner to install different types of services in a residential gateway, while enabling service providers to develop different varieties of heterogeneous services.

Security — As it is possible to run different types of services in one gateway security is an important requirement. The specification provided by OSGi defines different levels of security, including access control, digital signing of downloaded services, etc.

Simplicity — OSGi is a small and simple framework that could be deployed with just one 300KB in size JAR file [74]. The service environment provided by OSGi delegates all the complex management of services to professionals such as the gateway operator, thus making life simpler for the homeowner [72].

Key companies support — OSGi Alliance has diverse members including IBM, Nokia, Samsung, Deutsche Telecom, Redhat, Ericsson, etc. [74]

3.2 Access Control Mechanism

In order to ensure that only the correct person has rights to access services that are running in a residential gateway an access control mechanism should be deployed. There are different types of access control models available: Mandatory Access Control model, Discretionary Access Control, Role Based Access Control, and the eXtensible Access Control Mark-up Language.

3.2.1 Mandatory Access Control (**MAC**)

The Mandatory Access Control (MAC) [75] model is a system-centric access control model where the administrator of the system predefines the permissions of the users after an administrative procedure. Even the owners of the residential gateway do not have control over these predefined permissions to access or deny the services. When using this model, the subjects (which could be a service or an appliance) are tagged with a security clearance. In order to access a service running in the residential gateway, the person must have the proper clearance. The MAC based model is mainly used when confidentiality and integrity are the primary requirements. Table 3-1 summarizes the advantages and disadvantages of the MAC model in the context of a residential gateway.

Table 3-1: Advantages and disadvantages for MAC

Advantages	Disadvantages
• Straightforward method used in hostile systems where the risk for attack is high • Confidentiality and integrity are the primary concerns	• Expensive and difficult to implement • Not very adaptable for dynamic changes to permission policies

3.2.2 Discretionary Access Control (**DAC**)

Discretionary Access Control (**DAC**) [76] is a homeowner centric access control model where the owner of the residential gateway sets the permission for each user to access different services. The homeowner has full discretion to allow or deny access another home resident access to a resource (which could be a service or an appliance). When using this access control model, there should be a homeowner who takes responsibility as the residential gateway administrator to assign permissions in the home by maintaining an access control list (ACL). Many operating systems *such as Linux, Windows, and Macintosh OS) utilize a DAC based access control mechanism. Table 3-2 summarizes the advantages and disadvantages of the DAC model in the context of a residential gateway.

Table 3-2: Advantages and disadvantages for DAC

Advantages	Disadvantages
• User has complete control over access rights • Suitable for a small environment such as a home	• User must add all the access control policies. • User should be aware all the types of security threats when writing these access control policies. This could lead to potential openings for unauthorized access by malicious players.

3.2.3 Role Based Access Control (**RBAC**)

Role Based Access Control (**RBAC**) [77] is an improved version of DAC and MAC models where each homeowner is assigned a role based on pre-defined permissions. The administrator (homeowner) can set the roles for the other residents of the home. When the administrator assign individual access rights, the administrator simply sets their role and in turn the preconfigured access rights will be taken from the roles. The predefined permissions for these roles should be properly managed and checked for conflicts to avoid unauthorized access. As an example of home context, the roles could be “Father”, “Kid”, “Mother”, “Son”, etc. Table 3-3 summarizes the advantages and disadvantages of the RBAC model in the context of a residential gateway.

Table 3-3: Advantages and disadvantages for RBAC

Advantages	Disadvantages
• Less burden on home owner • Changes to the access policies are easy • If specific stakeholders predefine access rights for roles in a specific domain, the process of creating the domain specific access rights could be relevant.	Fine grained access control will be complex

Considering the advantages and disadvantages of different method of access control, the RBAC method was selected and implemented in the residential gateway using XACML.

3.2.4 eXtensible Access Control Mark-up Language (XACML)

Introduced and ratified by OASIS [78], the eXtensible Access Control Mark-up Language (XACML) is a generalized and fine-grained authorization standard that makes the process flexible hence it can accommodate requests from different types of services and environments [79]. The syntax for defining the access control rules and the access control request/response are XML-based. The standard also suggests a reference architecture with different types of elements that performs actions, such as receiving access requests, processing access request, and managing access rules [80]. The advantages of using XACML are:

- XACML is a standardized and widely adopted authorization approach
- XACML is generic, thus different types of services running in the same residential gateway can use this method of authorization. Access control policies from different service providers can be expressed in one common language.
- XACML extends the RBAC method of authorization and can use different attributes from internal or external sources to perform dynamic authorization. In this way the standard can provide fine-grained authorization.

3.3 Remote Connection

The remote connection feature of a residential gateway should allow a homeowner to remotely connect to and control the gateway. There are many ways to implement a remote access feature; the following are just a few of the methods that could be used:

VPN based remote access

The idea behind using a Virtual Private Network (VPN) [81] is to make two or more devices from different networks separated by a public network look as they are part of the same network by creating a virtual private network. In order to implement this type of remote accesses, the residential gateway needs to support VPN services and a VPN client is needed at the remote device. In this way a secure end-to-end tunnel is created between the remote client and the residential gateway and the remote client is seen as if it is locally present.

A service discovery protocol (such as UPnP) can be used together with a VPN, as this makes it easy for the remote client to discover the home network's services [82].

There are two basic issues that should be considered when implementing VPN based remote access. First, the initial set up procedure for setting up a VPN could be too complicated for a normal home user, so the setup method should be made easy. The second issue arises when using UPnP and VPN together, as UPnP devices send advertisements for the available service and this could create additional traffic in the VPN.

Cloud hosted resources

Using a cloud-based solution for performing remote access provides a solution that does not dependent on the underlying ISP's topology or even the home infrastructure – other than that there is Internet access to/from the residential gateway [83]. A cloud server acts an intermediary between the remote client and the home network. This means that the cloud server acts as a proxy for the resources available in the home network. In this way when the remote client tries to access the home network, it communicates with the cloud server using special APIs and sends requests. In turn, the cloud server communicates these requests to the home network. This can be done by synchronization of content between the home network and the cloud server. From the viewpoint of the client it seems that he/she is communicating directly with the home network. Samsung SmartThings is one commercially available example of such an implementation. Although this solution provides an easier implementation and scalability, the solution comes with the disadvantage of using 3rd party data storage, specifically privacy and storage cost [84].

Application dedicated web service

By taking advantage of entities, such as proxies or bridges, that can communicate with the devices in the private home network using UPnP or Bonjour a remote interface using HTTP or JavaScript pages could be accessed using a standard web browser. This would allow the remote client to access the devices in the home network. To secure this communication standard and well-known encryption standards such as TLS/SSL can be used.

Instead of using a web browser based solution, it is also possible to use APIs such as REST that could be embedded in a smart phone application to perform remote access [85]. When it comes to utilizing the full potential of a service provided by a device in a home network, this method of remote access may restrict the user because of restrictions posed by a web browser based user interface. Even with the restrictions posed by the web browser, the advantages of implementation and usage of this method for HTTP based remote access lead to this method being chosen for this thesis project.

3.4 Automation

To implement automation rules in residential gateway rule-based reasoning could be used to execute actions depending upon certain conditions.

3.4.1 Rule-based reasoning using production rules

Production rules contain basically two components: the set of conditions on one side and the set of actions on the other side. These production rules are generally represented in the format of “IF <Condition> THEN <Action>”. The way in which these rules are activated is called chaining. This chaining is classified into two types: forward chaining and backward chaining. Forward chaining first considers the left side of the rule, i.e. the condition of the production rule, then whenever the left side is true, then the respective action on the right side will be executed. In contrast, backward chaining is a goal driven method that tries to do backward reasoning. When it comes to diagnosis, the backward chaining method is useful, but forward chaining is interesting to make decisions based on available information. Forward chaining is more relevant the home automation scenario to execute automation rules.

To realize forward chaining there are two different types of implementation: inference rules and Event Condition Action (ECA) rules. These are further described as:

Inference rules — Inference rules is the simplest forward chaining mechanism. It consists simply of a condition and respective action. The general structure is expressed as:

IF <Condition> THEN <Action>

ECA rules — An extension to the inference rules, ECA provides one more condition that is based on event. The event part specifies for which trigger this specific rule should be executed. The general structure is expressed as:

WHEN <Event> IF <Condition> THEN <Action>

3.4.2 Type of triggers

The following types of triggers or events could be used in the automation condition to execute an automation rule:

Time based trigger When an automation rule is triggered due to time based events. Such a time based trigger could be generic (for example morning, evening, or day based) or the trigger could be more specific, such as on Monday at 10:00 AM.

Sensor based trigger When an automation rule is triggered due to a value returned by a sensor or appliance. For example, the value from a temperature sensor could trigger a automation rule to change the temperature in the home.

State based trigger Homeowner could configure a state (or even predefined) and state transitions could be used as a trigger. For example the states could be "when the home is empty" or simply "the bulb is ON".

Service based trigger A service-based trigger is very similar to a sensor-based trigger, but the input value is not from a sensor but rather from another service running in the same residential gateway. For example, an email service could send a trigger that indicates that an email was received.

3.5 Context Awareness

Schilit, Adams, and Want [53] introduces context aware computing and defined it as the ability of a system to adapt to the environment it is in. The context could be location-based, people or object nearby, changes in the prosperities of the objects, etc. A smart home that is context aware collects information from the appliances and different sensors in order to make intelligent reasoning and decisions.

As a first step towards constructing a context aware component for the residential gateway, a formal context model should be established. For this thesis, OWL based semantics have been used for developing ontology based context model. The advantage of OWL is that it is more expressive than other ontology based languages. The second advantage is that OWL provides semantic level interoperability between different domain services for sharing context knowledge and providing automated reasoning.

The ontology is constructed to describe the status of home. This ontology is expressed as RDF triplets: subject, predicate, and object. The object and subject are the ontology entities, while the predicate is the property relationship between the subject and object. This ontology describes the home domain and is used to maintain high-level information for reasoning and querying. The constructed smart home ontology has three first level classes: device, location, and event (as shown in Figure 3-4). The device class is used for defining different types of appliances and sensors that could be connected to the residential gateway. The event class describes the events generated by a device or combination of devices. Finally, the location class is used to describe the location of a device or the location where the event happened.

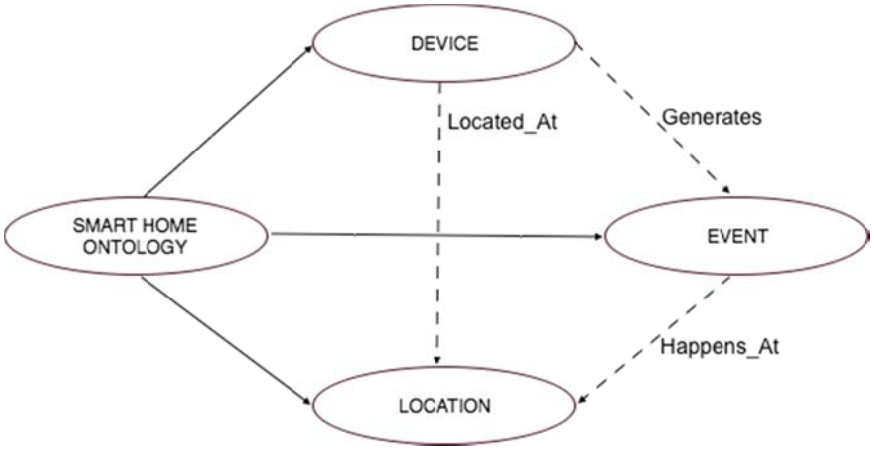


Figure 3-4: High-level ontology

Details of the classes are given below.

Device class The devices class (shown in Figure 3-5) has two sub-classes: appliances and sensors. These sub-classes are used to classify the type of device connected to a residential gateway. The devices that generate different sensor information are classified as sensors; for example, motion sensor, light sensor, temperature sensor, etc. The sensor class includes sensors from different domains including automation and Telecare. The other type of subclass of device class is an appliance. This class is used to define devices that do more than just sensing. For example, a washing machine or television. Such devices have extra properties or actions beyond simply returning sensor information. For example, with a television we can adjust volume or change channels and with a washing machine we can set different modes of operation.

The device class has two high level properties: Located_At and Generates. The Located_At property defines where the device is located and the Generates property defines the type of event the device generates. In general the devices class supports three basic operations: TurnOFF, TurnON, and GetCurrentStatus.

Event class The event class (shown in Figure 3-6) is used to define different types of events happening in the home. The event could be a low level event generated by a device or a high level event that results from reasoning with different types of sensor information. Each event class has two types of property defining the location where the event happened and another defining indicates which device generated the event. Depending upon the types of event or domain, the event class has three sub classes: security based event, health based event, or ambience based event. A security-based event is related to the security of the home. This subclass includes fire threats, burglary alerts, etc. Health-based event are related to the health of a homeowner and include high blood pressure, fall detected event, etc. Finally an ambience-based event relates to general home-related events and includes high room temperature, humidity change, etc.

Location class The location class (shown in Figure 3-7) is used in the smart home ontology to define the location of an entity. The location class could define a location of a device or where an event has happened. A

refrigerator device could have “kitchen” as its “Located_At” property. A motion sensor could generate an event “motion detected” at “living room” as its “Happens_At” property. In the smart home ontology, the location class has two subclasses (indoor and outdoor) depending upon where the device is located or an event happens. The indoor class includes the living room, bedroom, etc., while the outdoor class includes garage or garden.

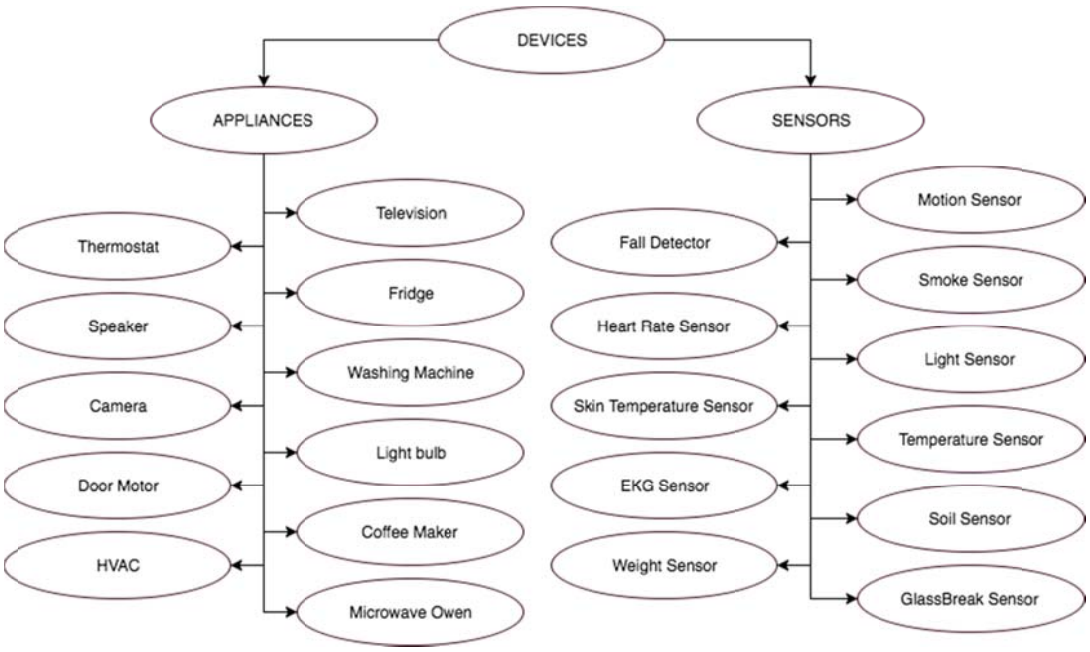


Figure 3-5: Device class

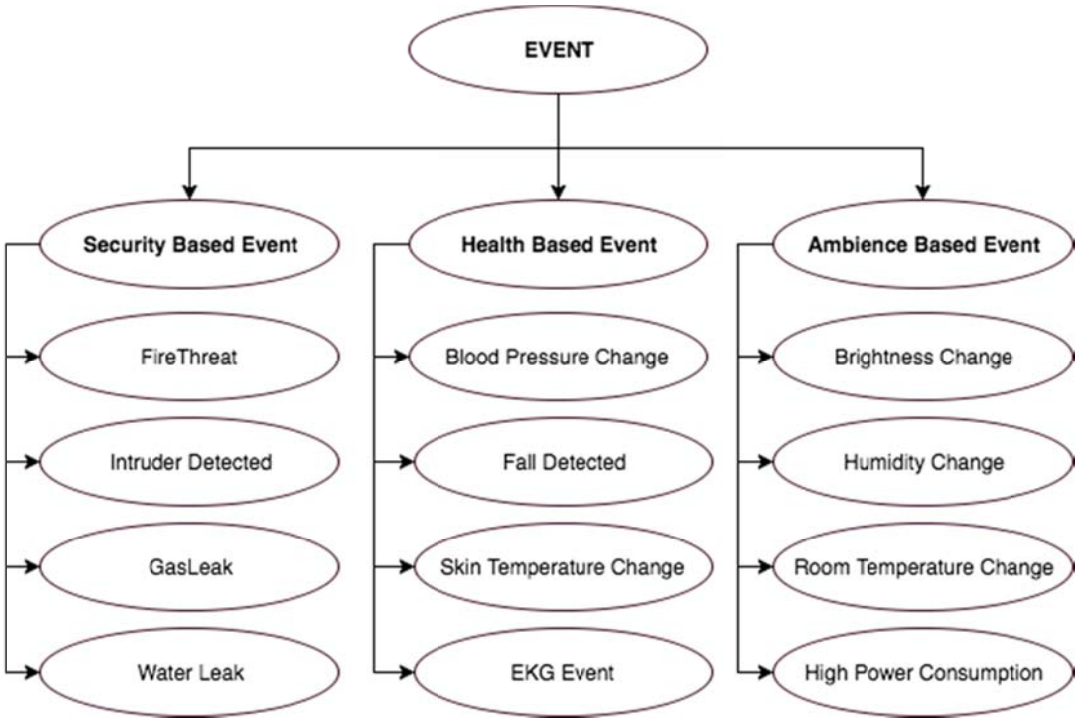


Figure 3-6: Event class

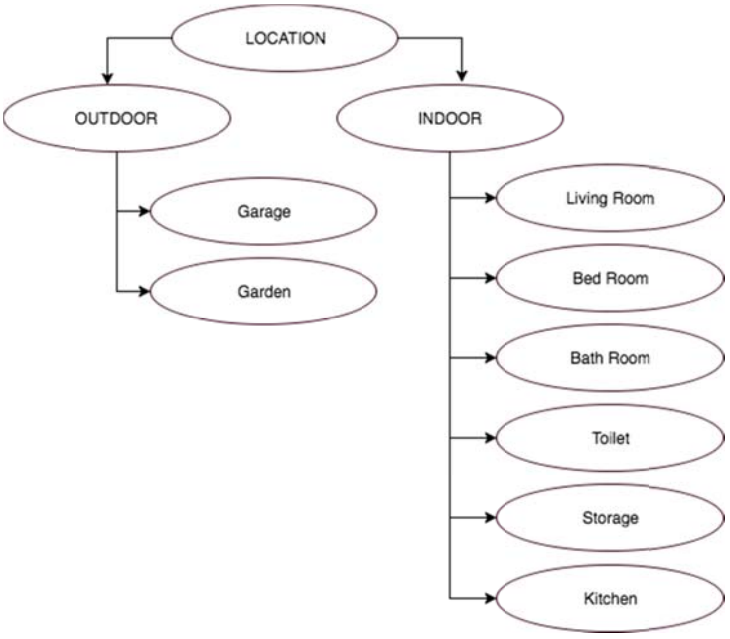


Figure 3-7: Location class

4 Experimental Gateway Design

This chapter contains the detailed explanation of the individual components in the proposed residential gateway design.

4.1 The Essence

Figure 4-1 briefly outlines the essence and key entities involved in the residential gateway design. We will assume that the homeowner receives the residential gateway from a gateway operator and installs this gateway in his/her home to provide different types of services. The gateway operator is assumed to have already configured the gateway with a few essential services before delivering it to the homeowner.

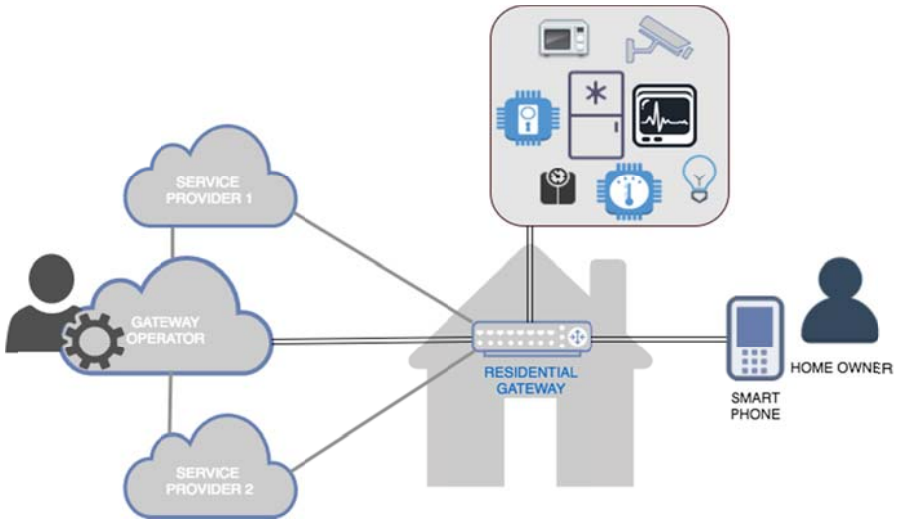


Figure 4-1: Overall view of a residential gateway and its context

Using the residential gateway, a homeowner could connect different types of home appliances via different communication medium. Additionally, a homeowner can install different types of services with or without connected appliances. These services will coexist together with other services in the residential gateway. The different types of services include device control, home automation, Telecare, home security, etc.

The homeowner is expected to install a mobile application developed by the gateway operator to securely communicate with the gateway. In this thesis project the communication with the residential gateway is achieved using a HTTP service running in the gateway. Using this application the homeowner can view the different services installed in the residential gateway. Additionally, the homeowner can open each service to perform some actions using this service or simply to view the status of the service. We will assume that the residential gateway can only be accessed via this application.

During the initial set up, the homeowner registers him/herself as administrator of the residential gateway via the gateway operator. Next the gateway administrator in this home can add additional homeowners to the residential gateway and assign them different roles. These users also need to be registered with the gateway operator. Subsequently the different homeowners from the same home can access different services depending upon their assigned access rights. For example, perhaps only an adult homeowner can access the

thermostat to change the temperature. As another example, a child homeowner is not allowed to add another new homeowner. Using the client mobile application a homeowner with appropriate access rights can view the status of sensors, control home appliances, add home automation rules (with or without appliances), etc.

4.2 Homeowner Authentication Management

An authentication process is used to verify the identity of homeowner who wishes to communicate with the residential gateway. This user authentication can be based on "something you know", "something you have", or "something you are" [86]. Using a username and password combination is an example of "something you know" based authentication. This is a popular and low cost method. Using biometrics, such as fingerprints or iris scan, are examples of "something you are" authentication. This is a more secure but expensive method compared to a password. Finally, "something you have" based authentication involves using smartcards or hardware tokens that store some information to identify the owner. To implement homeowner authentication for this thesis project, the password method was selected for simplicity. However, in future work this could be extended with other authentication methods.

The gateway operator plays a vital role in this implementation of password-based homeowner authentication. The gateway operator implements an authentication server that stores the homeowner's authentication details such as homeowner ID, password, and residential gateway ID. When a homeowner connects to a residential gateway using the client application, a login page requesting the homeowner's ID and password is displayed. A login component implemented in the residential gateway takes care of this authentication process (as shown in Figure 4-2). The homeowner ID and password could also be stored in the residential gateway for direct authentication, but considering security and extensibility factors the authentication server of the gateway operator is used.

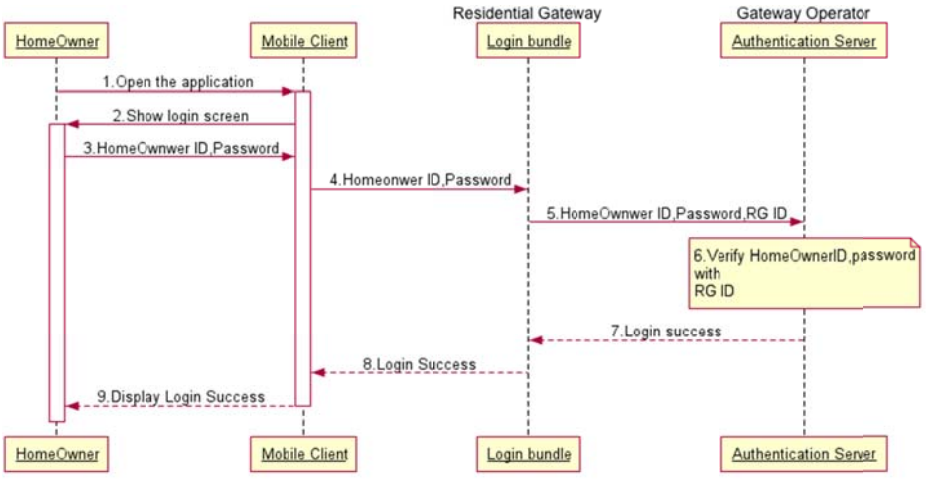


Figure 4-2: Authentication process

This authentication procedure relies to two components:

Authentication Server — The gateway operator implements the authentication server. This authentication server stores the homeowner's authentication details (as described above). When the server receives an authentication request, it searches in its database

to retrieve the credentials that will be used subsequently to authenticate a homeowner of a specific residential gateway. If the homeowner's credentials are correct, then the server replies back with "Login success" message. Otherwise, it replies with error message stating that the credentials were incorrect.

Login bundle The login bundle resides in the residential gateway. This login bundle works with the gateway's management component to require authentication of the homeowner. The gateway's login component is responsible for collecting the homeowner credentials from the client application and securely passing this information to the authentication server. Depending upon the response from the authentication server, the login bundle allows the homeowner to access the home gateway service or not.

4.3 Homeowner Authorization Management

The authentication process described in the previous section detailed how someone who is trying to access the residential gateway can be authenticated as a homeowner for his or her respective residential gateway. This section explains how an authenticated user is subsequently authorized to access different services running in their residential gateway. To reduce the burden on the homeowner given the relevant authorization, the RBAC method of access control was chosen for this thesis. Each homeowner is given a role and the services in the home are visible & accessible to the homeowner depending upon their assigned role.

The role manager bundle in the management component is responsible for managing user roles in the residential gateway. The role manager stores the homeowner's ID and the role of the respective homeowner. Once the homeowner is authenticated, the role manager fetches the role information and considers the request coming from the homeowner.

The following are examples of roles that could be created:

Administrator A homeowner who has access to almost the entire home services and also the gateway configuration related rights. The administrator is also responsible for:

1. Downloading services from gateway operator's service store
2. Adding new homeowner and assigning roles to them
3. Adding custom access control rules

Adult Next to the administrator, the adult homeowner has greater control over the installed services. Such an adult homeowner could access and configure complex home appliances such as HVAC appliances, security devices, etc.

Child A child has a homeowner role that only has restricted access to services.

There could additional roles, such as elder, visitor, … and the access control could be configured for each of these roles.

4.3.1 Access control component

Adopting the reference architecture of XACML the access control component (shown in Figure 4-3) that is defined for this thesis project contains the following bundles that execute the XACML access control method:

PEP The Policy Enforcement Point (PEP) is the front interface that receiving access requests from services running in the residential gateway. The PEP converts the service request to an XACML request and sends it to PDP to make an access control decision.

PDP The Policy Decision Point (PDP) is the brain of the access control component as it makes the decision of whether to authorize an access request or not. In order to make this decision the PDP searches the access rules already saved in the PRP. If necessary, the PDP also collects external information with the help of the PIP from the connected appliances and other services in order to make the authorization decision.

PRP The Policy Retrieval Point (PRP) stores all the access rules in XACML format. The PRP also identifies conflicts between different access control policies.

PIP The Policy Information Point (PIP) is used by PDP to collect the necessary information from the resource component that is needed to make the authorization decision.

PAP The Policy Administration Point (PAP) is an external database residing external to the residential gateway. The PAP has all the access control policies for the services provided by the gateway operator. Every time a new service is installed in the residential gateway the associated access control policies are downloaded to the PRP from the PAP.

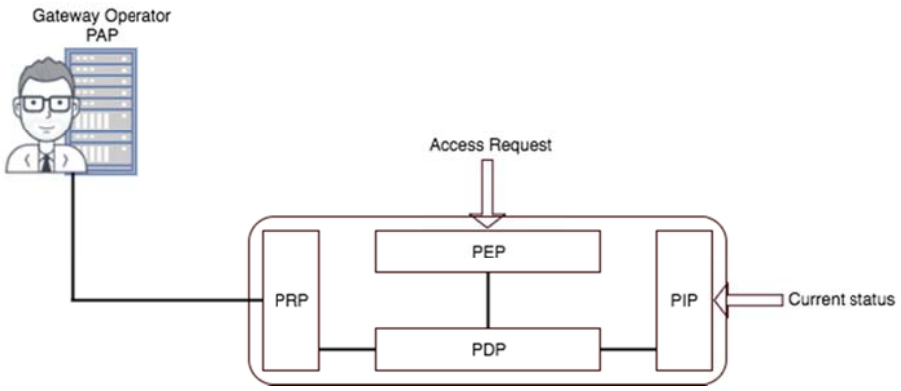


Figure 4-3: Access control component

4.3.2 Authorization procedure

As an example, consider an adult homeowner who would like to switch ON a lamp in the living room. Using the respective service bundle adult homeowner requests the action to switch ON the lamp. The service bundle sends an authorization request to the PEP inside the access control component. The PEP converts this request from the action request to XACML format and then passes it to PDP, where the PDP determines whether this action request should be approved or not. The request element contains information such as the subject attribute, resource attribute, action attribute, and an optional attribute. The

authorization components are shown in Figure 4-4, while the complete authorization process is shown in Figure 4-5.

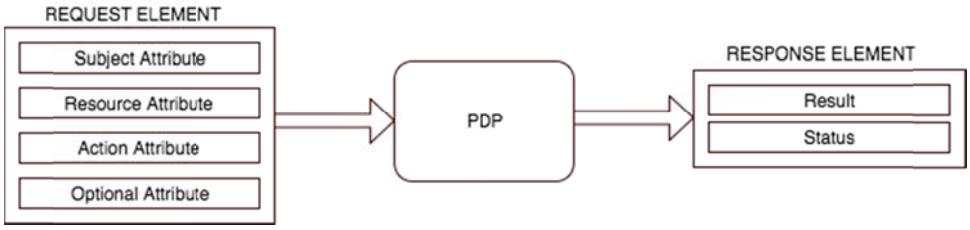


Figure 4-4: Authorization component

The subject attribute defines the source of the request (i.e., the Homeowner's ID and source service). The resource attribute contains information about the target resource for which access is requested. The action attribute contains information about the type of action that is going to be performed.

The PDP first searches the access control policies that are stored in the PRP and matches against the action request. If the stored access policy specifies that an adult homeowner can switch ON the lamp, then the PDP sends back a response element indicating that access has been approved in a message to the PEP. The PEP allows the service bundle to switch on the lamp. The response element contains information such as the result of the access request (i.e, Permit, Deny, Not Applicable, or indeterminate). The response element could also contain status information that is used when there is any error during the evaluation of the authorization request.

In some cases, the stored policy could also mention some external attribute information that is to be utilized in determining the authorization. For instance, in the example of switching ON the lamp, the access policy might have another condition that the bulb can only be switched ON if the light intensity level is below a certain value. In this case when the PDP receives the request, it collects the light intensity level from the appropriate sensor using the PIP and then uses this information to decide upon the authorization. In this way the XACML standard not only uses the RBAC information that is stored in the PRP, but also can use external information to make a fine-grained authorization decision.

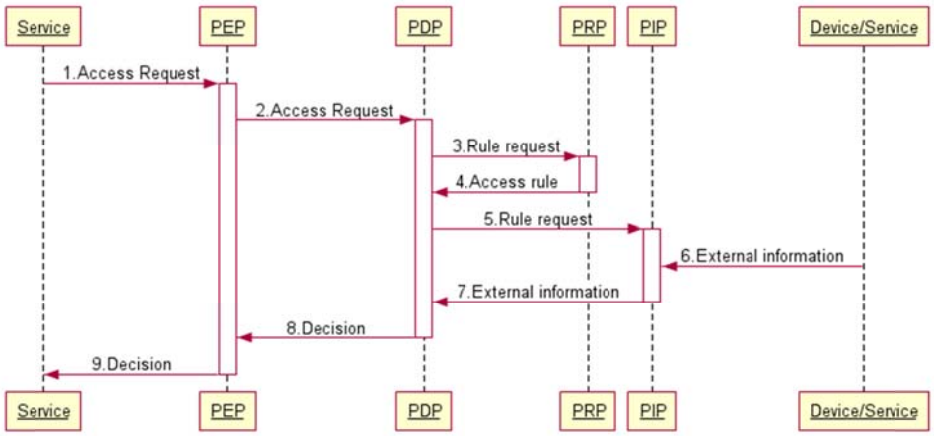


Figure 4-5: Authorization process

4.3.3 XACML Policy Structure

XACML policies are the primary information stored in PRP. These policies are used to determine whether an access request should be approved or not. At the top level, a policy set is a collection of policies or another policy set. A single XACML policy contains a target and rules. The entities and their relationships are shown in Figure 4-6. The target is an XML element that determines whether to execute the respective policy or not. The target is a collection of criteria conditions (for checking the subject, resource, and action) that is used to choose which policy to choose from a collection of polices. Once the PDP receives the request element, it search for the relevant policy in the PRP using the information from the request element (subject attribute, resource attribute, and action attribute).

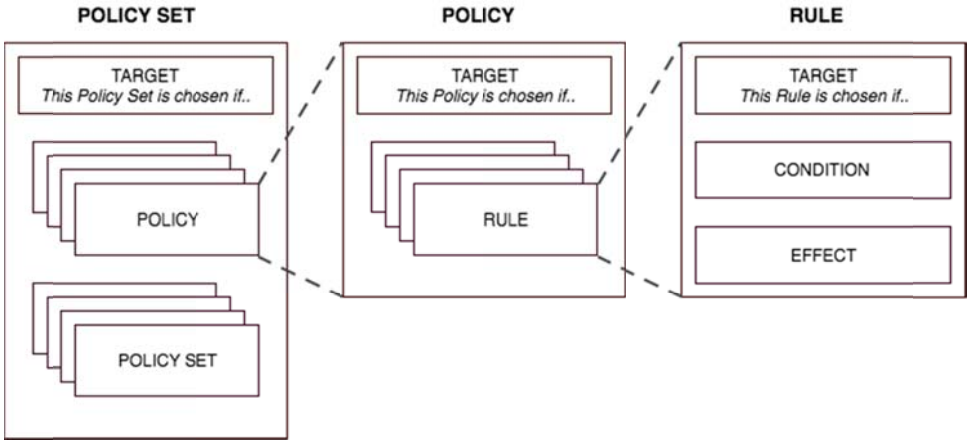


Figure 4-6: Policy set, Policy, and rule

Once the correct policy is chosen the rules in the policy are evaluated to determine the response to the authorization request. The rules are the core of XACML policies and each rule contains a target, condition, and effect. The target of the rule is used to select the rule and if the condition of the rule is satisfied, then the rule's effect will be sent back in a response.

Each policy set defines the list of policies corresponding to a service. A policy inside the policy set corresponds to the authorization policy.

4.4 Context Aware Resource Management

In order to perform context aware resource management the residential gateway uses a number of modules to realize a resource agent, a control agent, and a context agent. Their relationship is show in Figure 4-7.

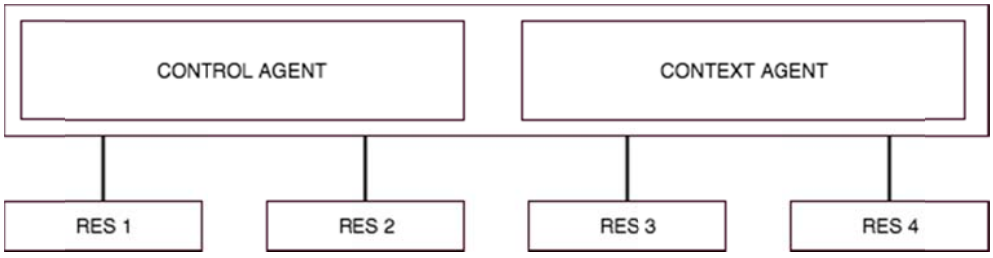


Figure 4-7: Resource modules

4.4.1 Resource Agent

The resource agent acts as a context provider in the context aware resource architecture. The resource agent is a bridge between the connected appliance(s) and the rest of the residential gateway. There are two primary tasks for the resource agent:

- To collect information from the connected appliance(s) or sensor(s) and send this information to the aggregation agent in the context agent. This could utilize either a push or pull based information retrieval mechanism.
- To execute the action request coming from the control agent.

4.4.2 Control Agent

The control agent bundle in the resource component is used by the service running in the residential gateway and the automation component to control the connected appliances and sensors. Once the access control module approves the device control request, the request is passed to the control agent in the resource module. The control agent passes the request to the respective resource bundle the device is connected to. Depending upon whether the requested action has been executed successfully or not, the control agent sends back an acknowledgement indicating the status of the request's execution,. If the incoming control request contains contextual information, then the control agent consults the context agent to collect the information necessary to process the request. For example, if the request is to switch ON all the lights in the living room, then the control agent receives information about the list of resource that have location context information tagged with "living room". Then the control agent passes the request to the respective resource bundles.

4.4.3 Context Agent

The context agent in the resource management component (shown in Figure 4-8) is in charge of providing context aware feature for the residential gateway. The following bundles are available in the context agent:

Aggregation Agent — The aggregation agent collects low-level context information from different resource agents and aggregates them to form a high level context. Usually the information coming from the resources agent is raw information and without context is meaningless, so the aggregation agent ensures the collected information is put together to form meaningful information which will be interesting for different services. For example, the information from a motion sensor is raw but when combined with location and time based context information produces more interesting information for services running in the residential gateway.

Context Engine — Once the aggregation agent converts the low level context to high-level context information, there could be some context information that should be filled in. The context engine acts as an inference engine using the domain knowledge and ontology information to complete the high level context. Once the high level context is complete, it is stored in the context repository.

Repository Interface As the name suggests, this bundle acts as an interface to the context repository. Service uses this interface to access control component, and the automation bundle to get contextual information from the repository. Services use this interface to read status or other information from a connected device in order to display this information to the homeowner. The access control component uses this interface to make access control decisions. Finally the automation component uses this information as a trigger to make automation decisions.

Subscription agent A service or automation component could register with the subscription agent for a specific change in the status of specific appliances. The subscription agent maintains a list of trigger events to watch out for and sends notifications back to the respective requestor that the event has happened. For example, the automation agent could subscribe with the subscription agent to receive a notification when a light in the living room is switched ON. Once the light is switched ON, the subscription agent sends a notification to automation agent and the automation agent could carry out a set of actions. The subscription process is fine grained in order to avoid accessing unauthorized information.

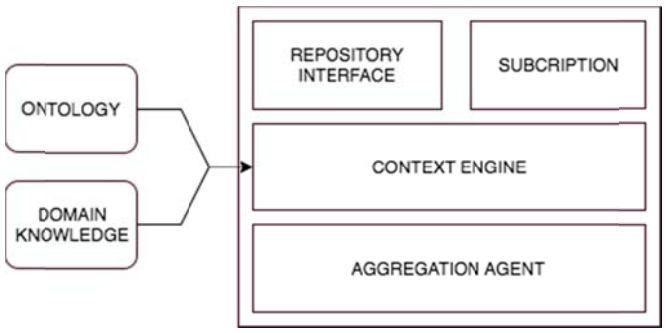


Figure 4-8: Context agent

4.5 Home Automation Management

The home automation management component in the residential gateway is the component responsible for managing automation in the residential gateway. It is responsible for accepting automation policies and executing them. Home automation using the residential gateway is carried out with the help of the automation policies.

4.5.1 Policy Format

To express automation policy in the residential gateway a simple ECA rule is followed. Each automation policy is expressed in the format:

WHEN <<trigger>>IF…<<Condition>>THEN…<<action>>

In order to execute an automation policy the trigger condition in the “WHEN” should be satisfied. The WHEN section could hold one or more trigger components to execute the

automation policy. Once the trigger condition is satisfied, then the optional “IF” condition is checked to determine of the automation action should be executed. To make the automation more accurate the IF section is used. Finally, once the trigger conditions and optional conditions are satisfied the action mentioned in the “THEN” section will be executed.

The following is an example simple security policy:

WHEN motion sensor detects motion
IF the home is in vacation mode
THEN send notification to home owner **and** take picture with living room camera

The following is another automation policy for controlling lights in the home:

WHEN light intensity goes below 60% **and** the current time is greater than evening time
IF the lights in the living room is switched OFF
THEN switch ON the living room lights

4.5.2 Automation component

The following are the individual bundles in the automation component:

Policy Server The policy server stores all the home automation policies configured by homeowner using a service.

Register The policy register bundle in the automation component acts as an interface to the policy server to write automation policies. The homeowner should use this register to save the automation policy in the policy server. Every time when a new automation policy is sent to register to save in the policy server, it checks for possible conflicts with the existing policies. If the register finds any possible conflicts it sends back error message back to the requested service. The process of storing a policy is shown in Figure 4-9.

Executor The executor bundle in the automation component executes the automation policies (as shown in Figure 4-10). The executor waits for triggers from different sources and executes the respective actions that are stored in the policy server and associated with these triggers. The executor can receive triggers from the subscription bundle in the resource context component, and external service, or an internal clock. If the respective automation action is to control a device, then the executor sends the action request to the control agent in the resource context component to execute the action.

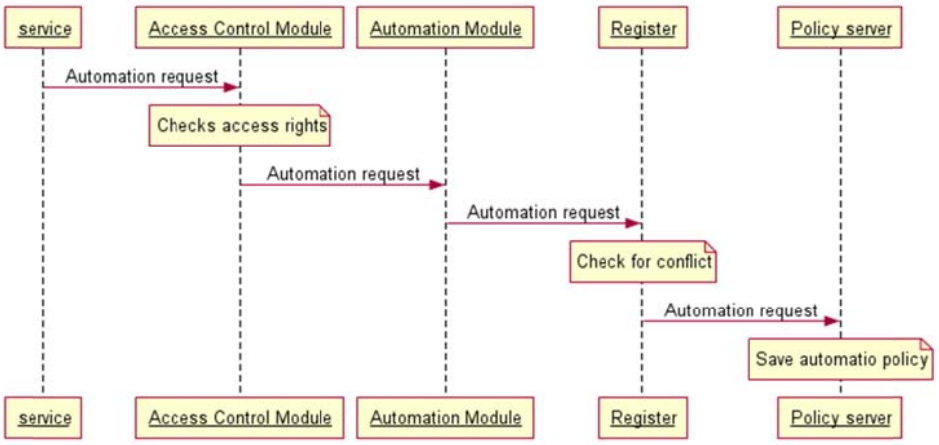


Figure 4-9: Saving a new policy

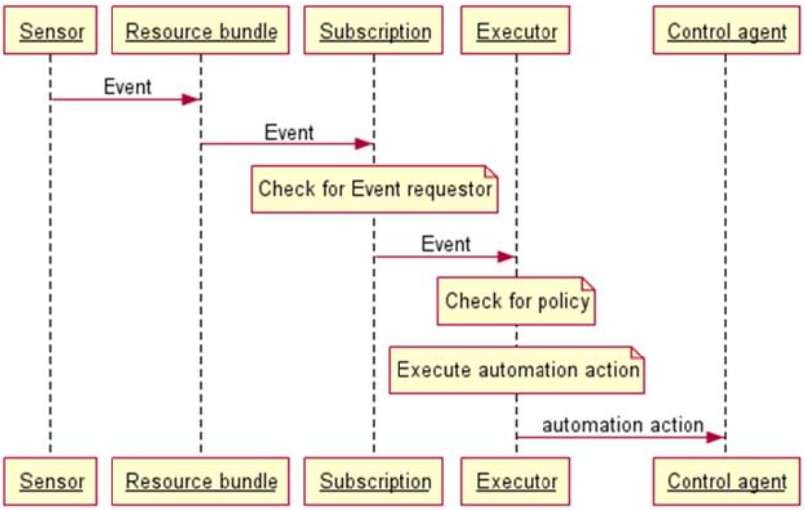


Figure 4-10: Executing a policy

5 Implementation and Analysis

The following section details the details the implementation of the proposed design. Starting with the generation of access control policies, followed by ontology construction and finally the implementation of login bundle in the residential gateway. This section also includes the analysis of the design with the use cases and non-functional requirements discussed in chapter 2.

5.1 XACML policyset generation

To make the process of authoring and development of XACML polices easier Axiomatics introduced their Abbreviated Language For Authorization (ALFA) a pseudo-code language. ALFA was then donated to the OASIS consortium in order to standardize the language and to make it available for everyone. ALFA reduces the complexity of writing access policies in XACML that uses XML as its main encoding language and the pseudo code written using ALFA can be directly mapped to XACML policies.

The ALFA plugin installed in Eclipse neon3 (release 4.6.3) was used to generate policies in ALFA. Once the pseudo-code policy is written in a ".alfa" extension file the plugin automatically generates an ".xml" file with an equivalent XACML policy in it and this file can be used for defining the access control using an XACML implementation. When a service provider develops a service, this .xml file with policy set specific to the service is also generated by the service provider and sent to the gateway operator. The gateway operator in turn validates the .xml file for security and privacy conflicts. Finally when the homeowner installs the service the residential gateway receives the .xml file and adds the policyset to the PRP.

Figure 5-1 shows a simple example of ALFA code that can be used to generate a policy file for a thermostat application.

```
/**
* PolicySet for Thermostat Service
*/
policyset thermostatServicee {
        target clause service.serviceID=="Thermostat service"
        apply firstApplicable
        /**
        * Policy-1: Policy for adult user with respect to thermostat service
        */
        policy adultAccess{
                target clause user.role=="adult"
                apply firstApplicable
                /**
                * Rule-1: Adult user can turn ON temperature for the thermostat
                */
                rule control{
                                target clause action.actionID=="TurnON"
                                permit
                        }
                }
        }
```

Figure 5-1: Policy set example in alpha

The structure of ALFA code is simple and straightforward as it explains the different components of a policy. Traversing through different policyset, policies, and rules with their respective target clause condition, results in making the access control decision. The code in Figure 5-1 is a simple policyset for a thermostat service that contains one policy and one rule inside the policy. The keywords policyset, policy, and rule are used to define the policyset for the thermostat service, the policy this applicable for adult users in the home, and a specific rule for allowing the adult user to perform a specific action. The target clause in each section is used to specify the condition for choosing the specific path during the decision-making and the keyword permit in the rule section specifies the final decision to allow the user access. Finally the keyword firstApplicable is used to inform the PDP to choose the first matching policyset or policy. Once the ALFA code is finished, the plugin automatically generates the .xml file and the content is displayed in Figure 5-2.

```
<?xml version="1.0" encoding="UTF-8"?>
<xacml3:PolicySet xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"

PolicySetId="http://axiomatics.com/alfa/identifier/com.smarthome.project.thermostatService"
    PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:first-
applicable"
    Version="1.0">
    <xacml3:Description>Policy Set for Thermostat Service</xacml3:Description>
    <xacml3:PolicySetDefaults>
        <xacml3:XPathVersion>http://www.w3.org/TR/1999/REC-xpath-
19991116</xacml3:XPathVersion>
    </xacml3:PolicySetDefaults>
    <xacml3:Target>
        <xacml3:AnyOf>
            <xacml3:AllOf>
                <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <xacml3:AttributeValue
                        DataType="http://www.w3.org/2001/XMLSchema#string">Thermostat
service</xacml3:AttributeValue>
                    <xacml3:AttributeDesignator
                        AttributeId="serviceID"
                        DataType="http://www.w3.org/2001/XMLSchema#string"
                        Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
                        MustBePresent="false"
                    />
                </xacml3:Match>
            </xacml3:AllOf>
        </xacml3:AnyOf>
    </xacml3:Target>
    <xacml3:Policy xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"

PolicyId="http://axiomatics.com/alfa/identifier/com.smarthome.project.thermostatService.adul
tAccess"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-
applicable"
        Version="1.0">
        <xacml3:Description>Policy-1: Adult user can turn ON/OFF/Set temperature for the
thermostat</xacml3:Description>
        <xacml3:PolicyDefaults>
            <xacml3:XPathVersion>http://www.w3.org/TR/1999/REC-xpath-
19991116</xacml3:XPathVersion>
        </xacml3:PolicyDefaults>
        <xacml3:Target>
            <xacml3:AnyOf>
                <xacml3:AllOf>
                    <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
equal">
                        <xacml3:AttributeValue

DataType="http://www.w3.org/2001/XMLSchema#string">adult</xacml3:AttributeValue>
```

```
                        <xacml3:AttributeDesignator
                            AttributeId="role"
                            DataType="http://www.w3.org/2001/XMLSchema#string"
                            Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-
subject"
                            MustBePresent="false"
                        />
                    </xacml3:Match>
                </xacml3:AllOf>
            </xacml3:AnyOf>
        </xacml3:Target>
        <xacml3:Rule
                Effect="Permit"

RuleId="http://axiomatics.com/alfa/identifier/com.smarthome.project.thermostatService.adultA
ccess.control">
            <xacml3:Description />
            <xacml3:Target>
                <xacml3:AnyOf>
                    <xacml3:AllOf>
                        <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
equal">
                            <xacml3:AttributeValue

DataType="http://www.w3.org/2001/XMLSchema#string">TurnON</xacml3:AttributeValue>
                            <xacml3:AttributeDesignator
                                AttributeId="acionID"
                                DataType="http://www.w3.org/2001/XMLSchema#string"
                                Category="urn:oasis:names:tc:xacml:3.0:attribute-
category:action"
                                MustBePresent="false"
                            />
                        </xacml3:Match>
                    </xacml3:AllOf>
                </xacml3:AnyOf>
            </xacml3:Target>
        </xacml3:Rule>
    </xacml3:Policy>
</xacml3:PolicySet>
```

Figure 5-2: Policy set in XACML

Developing XACML policy using ALFA plugin in eclipse was straightforward. The ALFA code was directly translated to an XACML policy set in an XML file. With XACML it is possible to write very granular access control policies. The residential gateway using the XACML method of access control could also perform dynamic access control. For example, once the access control module receives a request, it could check the current status of a device and make a decision depending upon the device's status. This provides flexibility for the access control. For example, if a rouge service is asking to switch on a microwave oven when there is no food inside it, then the access control component could check if there is any food inside and dynamically reject the access request.

5.2 Ontology Implementation

Protégé version 5.1.0 was used to develop the smart home ontology. Protégé is an open-source and free tool developed at Stanford University for designing knowledge-based system with ontologies. It was very simple to create the smart home ontology using the graphical user interface provided by the tool.

The OWL ontology basically contains three types of resources: classes, individuals, and properties. Classes are the fundamental blocks of the OWL ontology and define the domain

specific concepts. In our case the primary classes to define the ontology are: Device, Event, Location, and User. These are shown in Figure 5-3. Each of the primary classes are created by adding them as subclass of the owl:Thing class.

Figure 5-3: Class hierarchy view in protégé

Once the primary classes were created, the subclasses for individual primary classes were created with respect to the ontology design as explained in the previous section. For example the diagram in Figure 5-4 shows the subclass created for the location class. The location primary class contains two subclass Indoor and outdoor. In turn the indoor class contains other subclasses such as bathroom, bedroom, etc. Examples of these are shown in the right-hand part of Figure 5-3.

Figure 5-4: Object Property and Datatype property

The next step is to define OWL properties for the previously defined class and it defines the relationship. Using Protégé it is possible to define an *object property* and *datatype property* for a class. The *object property* defines the link between two individuals. The *datatype property* defines the link between an individual and a data value. The relationship between the classes Location and Device is defined by the object property Located_At. The User class could have a string datatype property UserID. See Figure 5-5.

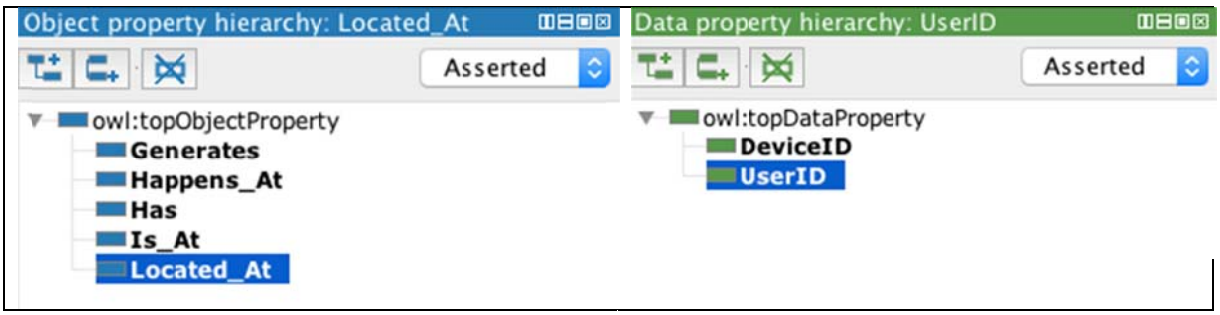


Figure 5-5: Object Property and Datatype property view in protégé

5.3 Bundle Implementation

A user authentication mechanism for the homeowner was developed using OSGi bundles. Using this mechanism a homeowner can register with the gateway operator and then can login to the gateway itself with correct credentials. To realize this mechanism, two different bundles were created. One bundle in the residential gateway collects the homeowner's credentials for authentication when the homeowner use a HTTP web client to connect to the residential gateway. The other bundle is use as the gateway operator's authenticator bundle.

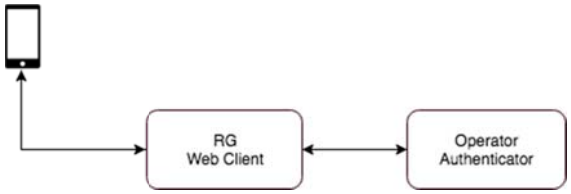


Figure 5-6: OSGi bundles for HTTP service

Netbeans 8.2 was used to create in the Java language the bundles with the help of the Apache Felix Maven OSGI Bundle Plugin [87]. Once the bundle Jar files are ready, there were implemented in Apache Felix. Apache Felix from the Apache software foundation is a open-source container for OSGi. All the dependencies were added in a pom.xml (Project Object Model) file in each bundle and then the maven plugin takes care of properly configuring the bundle for execution.

As a first step a **common bundle** was created to include a UAService interface with two basic methods login and register. This public interface is shown in Figure 5-7.

```
public interface UAService
{
   boolean user_login(String uname, String upassword);
   boolean user_register(String uname, String upassword);
}
```

Figure 5-7: UAService interface

The authenticator bundle was created in the next step with an implementation of the previously mentioned interface. The primary purpose of this bundle is to store the user's credentials on the gateway operator's side. To register the Authenticator service the bundle contains a Java file called *AuthenticatorServiceActivator*. This file is called as the Bundle activator and is also an interface to invoke when the bundle is started or stopped. The activator class is shown in Figure 5-8 and the authenticator implementation is shown in Figure 5-9.

```
public class AuthenticatorServiceActivator implements BundleActivator {
   public void start(BundleContext context) throws Exception {
```

```
        context.registerService(UAService.class.getName(), new UAServiceImpl(), null);
        System.out.println("AuthenticatorServiceActivator started");
    }

    public void stop(BundleContext context) throws Exception {
        System.out.println("AuthenticatorServiceActivator stopped");
    }
}
```

Figure 5-8: Activator class

```
public class UAServiceImpl implements UAService {
    private Map<String, String> credentials = new HashMap<String, String>();

    public synchronized boolean login(String name, String password) {
        return credentials.containsKey(name) && credentials.get(name).equals(password);
    }
```

Figure 5-9: Authenticator implementation

```
public class LoginServlet extends HttpServlet
{
    public void service(HttpServletRequest req, HttpServletResponse resp)
            throws ServletException, java.io.IOException
    {
        resp.setContentType("text/html");
        PrintWriter out = resp.getWriter();
        out.println("<HTML> <HEAD> <TITLE> Login " +
                "</TITLE> </HEAD> <BODY BGCOLOR=white>");

        String name = req.getParameter("name");
        String password = req.getParameter("password");
        try
        {
            ServiceLocator locator = new ServiceLocator(getBundleContext());
            try {
                if (locator.getUAService(-1).login(name, password)) {
                    out.println("You are now loggeed in")
                    out.println("Click on the following options");
                    out.println("To access the list of service");
                    out.println("Logout");
                } else {
                    out.println("Incorrect user name or password. Try again");
                }
            } catch (ServiceLocator.ServiceUnavailableException e){
                out.println("Service is not yet available");
            }
```

```
    }
    catch (Exception e)
    {
       e.printStackTrace(out);
    }
    out.println("</BODY> </HTML> ");

  }

  private BundleContext getBundleContext() {
    return BundleContext.class.cast(getServletConfig().getServletContext().getAttribute("osgi-
bundlecontext"));
  }
```

Figure 5-10: LoginServelet

Once the authenticator is ready, the residential gateway side web application bundle (wab) was created to collect the homeowner credentials to send it to the operator bundle for validation. Two different servlets were created for the login screen and registration screen. The login servlet takes care of collecting the user's credential for validation and the registration servlet is responsible for creating a new homeowner account. When the homeowner tries to login using the login screen the login servlet collects the login information and sends the information to the authenticator bundle using the code locator.getUAService(-1).login(name, password). Once the authenticator bundle receives this information, it will check if the user credential is available in the memory to validate. If the credentials matched it returns a true back to the login servlet. Figure 5-11 shows screenshots of the simple authentication process using the above-mentioned program.

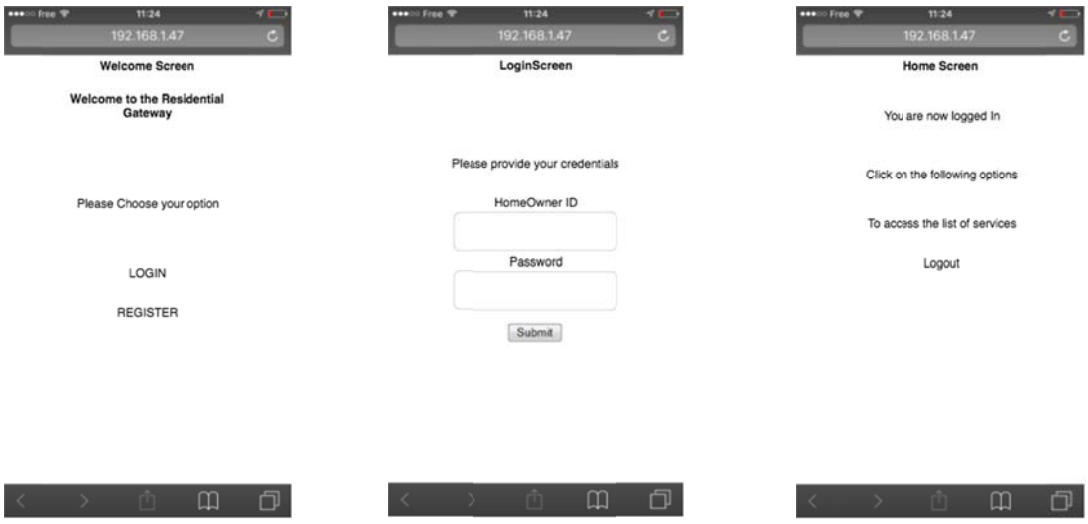


Figure 5-11: Login Screen using OSGi bundles

Inference: With regard to the implementation, it was quite clear that modularity and heterogeneity could be achieved using OSGi. The bundles were easy to install, remove, or modify at run-time without stopping any other service. Developing a web service bundle with Java servlets was easy with the help of maven and net beans. Online resources and documentations available for OSGi development played an important role in understanding the concepts and requirements. Given this implementation experience, it is obvious that developing an OSGi bundle is simple and that the online support for bundle development was definitely significant. As the next steps in the process of implementation, the access control modules, automation modules and the context aware resource modules should be implemented. The first step could be implementing a automation bundle for controlling a simple connected devices like light bulb and then extend this automation bundle for supporting multiple device. Once the automation service bundles are implemented, the Telecare services could be added later.

5.4 Functional Use case analysis

In order to analyse how the proposed residential gateway design function, the use cases described in Section 2.9.2 were taken into consideration. The data and control flow during each use case provide us with information regarding the functional properties of the residential gateway. The three functional use cases are described in the following section that covers most of other functions that could be performed by a homeowner.

5.4.1 Use Case 1: In home or remote control of devices

This simple use case requires the residential gateway to act as a bridge between the homeowner and the device connected to the residential gateway. For this use case we will assume that the homeowner would like to increase the temperature of a room in the home. We will assume that the user has already authenticated using the authentication procedure explained earlier. Figure 5-12 shows the overall sequence diagram for this use case.

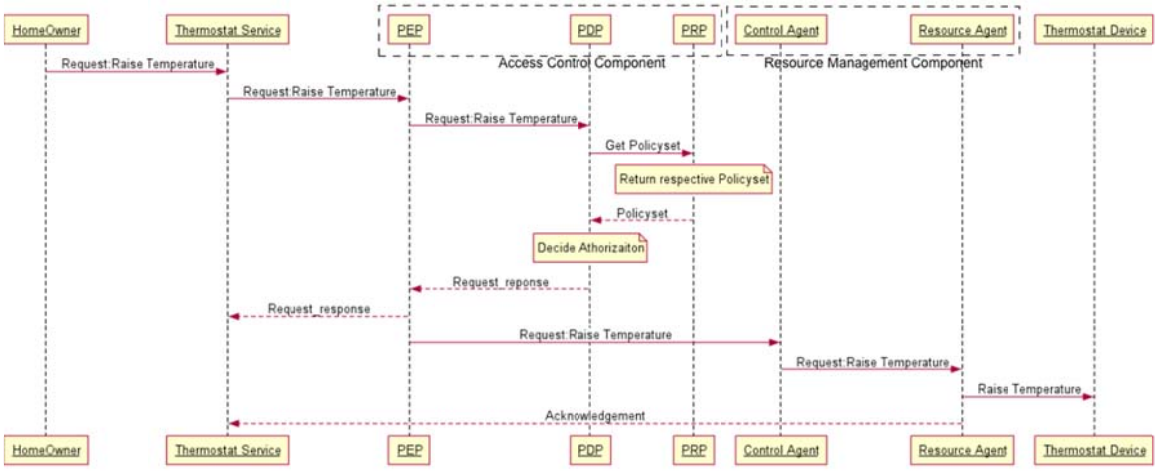


Figure 5-12: Sequence diagram for controlling a device

Step 1: The homeowner uses the already installed thermostat service to send a request to increase the room's temperature. The thermostat service sends an action request to the

access control component with the necessary information (i.e., service name, target device, and action). This request is shown in Figure 5-13. The service name is used to identify the service that initiates the request, the target device is used to identify the device to be controlled, and the action is used to identify which action is supposed to perform on the device.

Homeowner Role	Service Name	Target device	Action
Adult	Thermostat	Thermostat_1	Increase Temperature

Figure 5-13: Action request

Step 2: The PEP is the first bundle in the access control component to receive the request from installed services. The PEP receives the request from the thermostat service in order to deciding **if** the action should be approved or not. The PEP converts the request into XACML format and sends it to the PDP to make the decision. Finally, the PDP checks its database for a matching policy. The PDP sends a decision back to PEP based upon the result of the policy. For the purpose of this description, we will assume that the policy approves the action request.

Step 3: Once the decision is made, the PEP forwards the action request to the control agent bundle in the resource management component. The control agent in turn forwards the request to the resource agent that is connected with the thermostat to execute the action. The resource agent in turn executes the requested action to increase the temperature.

Step 4: Once the action is executed, the resource agent responsible for the thermostat sends a status message back to the thermostat service that requested the action.

Inference: The access control component and resource components are actively involved in this use case. Because of the access control only an *authenticated* user is allowed to access the services installed in the residential gateway and only an *authorized* user for this specific service can access information or control the device's specific function.

The access control policies (written in XACML format) allow the gateway operator, service provider, or homeowner to write fine-grained policies. In this use case, the XACML access policy that allowed the thermostat service to access the thermostat device to increase the room temperature could have been expressed as follows:

IF the request origin service is "**thermostat service**" and
IF the requesting homeowner role is "**adult**" and
IF the target device is "**thermostat_1**" and
IF the requested action is "**Increase temperature**".
THEN permit.

By providing this fine level of granularity in the access rules an unauthorized access of the device's actions can be avoided and the installed service is restricted to accessing only those actions allowed for the specific service and user. For example, even if the thermostat service has an access rights to increase the temperature, the same service could not decrease the temperature or perform other actions offered by the thermostat device. In

order to decrease the temperature would require the addition of another access policy for this service and device.

The need for two different policies (for increasing and decreasing) and the need to write policies for specific devices both suggest that it would be helpful to be able to write policies that have variables in them. For example one could generalize the policy to handle any of the N instances of thermostats as well as permit both an increase or decrease in temperature:

> IF the request origin service is "**thermostat sevice**" and
> IF the requesting homeowner role is "**adult**" and
> IF the target device is "**thermostat_$N**" and
> IF the requested action is "**Increase temperature**" or "**Decrease temperature**".
> THEN permit.

However, the extension of the system to handle such generalizations is left for future work.

As this use case just deals with sending a simple action request for one device, the control agent directly sends the command to the respective resource agent bundle. However, if a service requests for an action is to be done for set of device based upon location context information; for example, switching OFF all the light bulbs in the living room. Then the control module would use the help of the context engine to find all the relevant devices that belong to the location "living room" and then send each of them an individual request. The access control component will then do the same access control processing for each of these requests.

A potential extension for future work is to be able to make the access control decision on the high level request, rather than repeatedly making the decision of each of the resulting low level requests.

5.4.2 Use Case 2: Home Automation

This specific use case enables a homeowner to add automated actions in their home based upon triggers initiating the automated action. For this case we will split the analysis in two parts: saving the automation policy and details of the automation process.

The automation policy will be: "when there is a motion detected in the living room and If it is after 5 PM then switch ON all the lights in the living room".

To facilitate the understanding of this use case we will assume that the motion sensor and the light bulb services provided by the same service provider, hence they can be used by a single "lightsaver" service.

5.4.2.1 Use Case 2: saving the automation policy

In order for the authenticated homeowner to save an automation policy in the residential gateway the following steps (see Figure 5-15) will take place:

Step 1: To prepare the home automation policy, the homeowner uses the installed service that is dedicated to preparing automation policies or the service that is responsible for the connected device. In this use case, the authenticated homeowner prepares the automation policy and sends the automation request to the access control component. The request (shown in Figure 5-14) contains the requesting service, requesting homeowner's role, the automation policy.

Homeowner Role	Service Name	Automation policy
Adult	lightsaver	AutomationPolicy_1

Figure 5-14: Automation request

Step 2: The PEP receives the automation request from the lightsaver service and must decide if the automation request should be approved or not. The PEP converts the request to XACML format and sends it to PDP to make the decision. The PDP checks its database for a matching policy. The PDP sends its decision back to PEP. For understanding let us consider that policy approves the action request. As this automation request involves contextual information like light bulbs located in the living room, the PDP requests the context engine in the resource management component to get the list of light bulbs in the living room. Then the PDP checks the access control policies with respect to the action and the list of bulbs in the living room. Depending upon the policy the PDP sends the decision back to PEP. For the purpose of this description, we will assume that the policy approves the action request.

Step 3: Once the decision is made, the PEP forwards the automation request to the register in the automation component. The register first checks if there are any conflicts with existing automation policies and if not, then it saves the automation policy. Then the register bundle then subscribes to the trigger stated in the automation policy by using the subscription bundle in the resource management component.

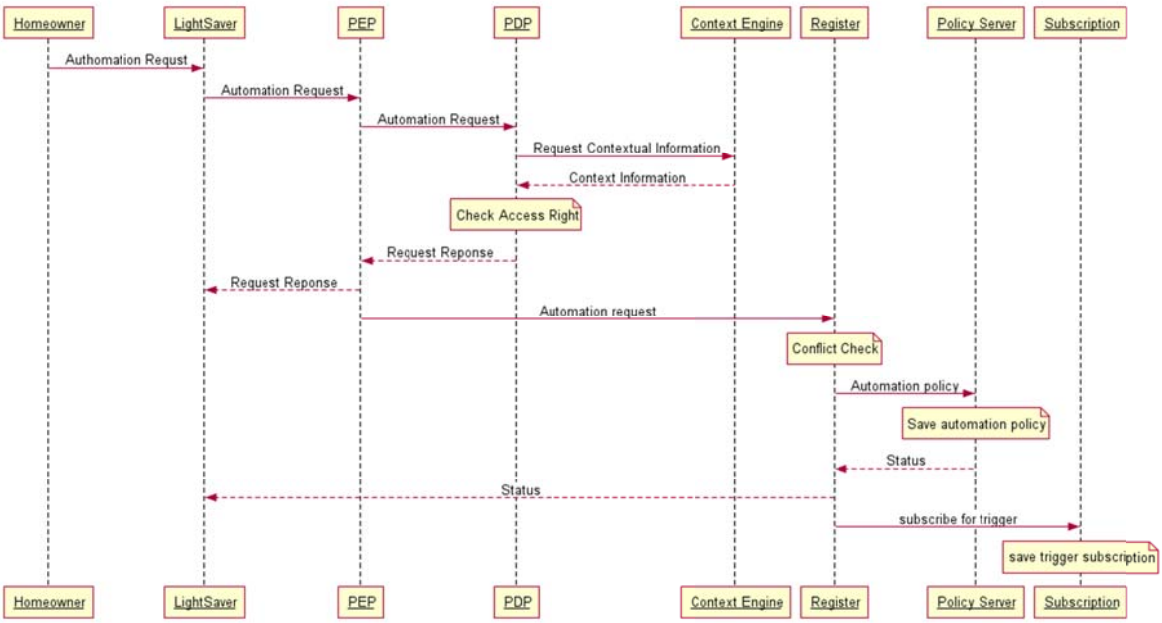


Figure 5-15: Sequence diagram for saving an automation policy

5.4.2.2 Use case 2: details of the automation process

The following steps will be following by the components in the residential gateway to execute an automation policy. Figure 5-16 shows a sequence diagram of these steps.

Step 1: The subscription bundle in the resource management component is responsible for sending information to the requestor when a specific action has happened. The executor bundle in the automation component executes the respective automation policy. In this specific use case the subscription bundles waits for a motion-detected event from the motion detector connected in the living room. Once the subscription bundle detects the motion-detected event, it sends the event information to the executor bundle in the automation component.

Step 2: When the executor bundle receives the event information from the subscription, it checks the list of automation policies stored in the policy register. When the executor finds a match policy for the received event trigger, then the executor checks the other conditions associated with the automation policy. In our example, the other condition is based on time so the executor checks whether the time related condition is also satisfied. If all the conditions are satisfied, then the executor passes the action to be executed to the control agent in the resource management component.

Step 3: When the action to be executed is received, then the control agent executes the requested action. If the target devices are simple, the control agent directly forwards the action request to the respective resource agents; while if the target devices are given in a contextual format; the control agent sends a request to the context agent to learn the specific devices. In this specific example, the control agent requests the list of light bulbs located in the living room and the control agent sends an automation action to the respective resource agents of each of the light bulbs located in the living room.

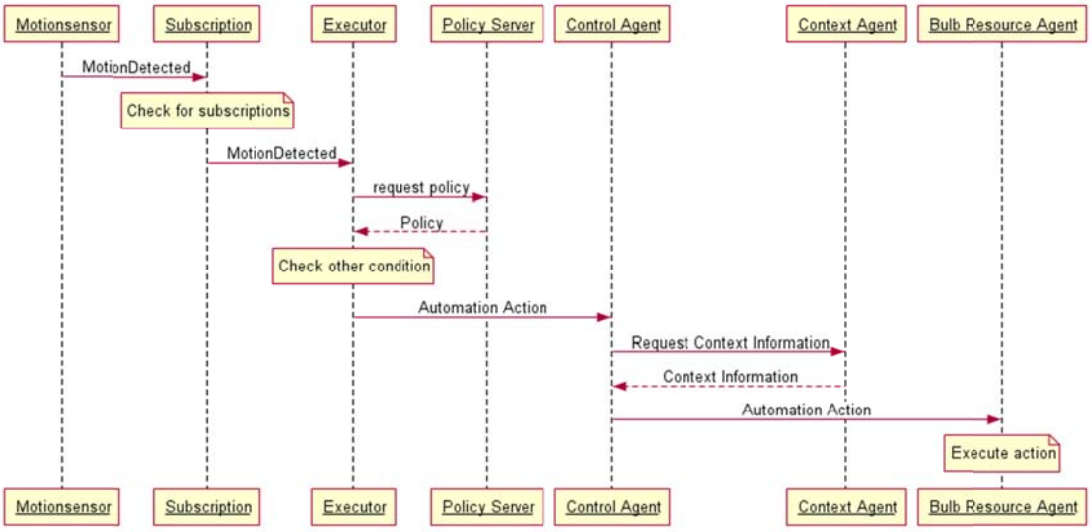


Figure 5-16: Sequence diagram for automation rule execution

Inference: A homeowner using dedicated services could an add automation policy to perform automated actions in their home. This home automation feature is achieved using the automation component and resource management component. In order to avoid unauthorized access of resources, the access control module is also involved. As in the previous use case, only an *authenticated* homeowner with an *authorized* service is allowed to add a home automation policy.

The residential gateway design provides different types of triggers for creating an automation policy. Moreover, the trigger for the automation is not restricted to events

coming from the connected device, but these triggers could come from another service. For example, an email service running in the residential gateway could provide an email received event and this could be used to trigger the creation or configuring of an automation policy. In addition to triggers, the design provides the ability to create automation with heterogeneous devices. For example, a trigger coming from a gas leak sensor could be used with thermostat device or other electrical devices to provide a safety service.

One of the primary advantages of storing the automation policy in the residential gateway is that even if there is no Internet connectivity the residential gateway can execute the (stored) automation rules with the connected devices.

5.4.3 Use Case 3: Remote Patient Monitoring

In this use case we assume that a health care provider (such as a hospital) is interested in monitoring their outpatients. The hospital develops its own healthcare system that includes a server to manage patient data along with a patient side OSGi service. The healthcare provider makes the patient side OSGi service available in their service store so that the patient & homeowner can install it in their residential gateway. The patient side OSGi service installed in the residential gateway acts as a communication proxy to securely transfer the data collected by a medical device in the home to the hospital's server. Once the patient's medical information is available in the server, then a professional from hospital side can view this data using a dedicated portal designed by hospital. For this example we will consider the case where the patient is interested in taking a blood pressure measurement and sending this measurement to the hospital's server.

As in the previous use cases we will assume that the patient & homeowner is already authenticated, that they were authorized to add the patient side OSGi service to their residential gateway, and that they have already added this service to their residential gateway. Figure 5-17 shows the sequence diagram for this use case. Details of the individual steps in this process are given following the figure.

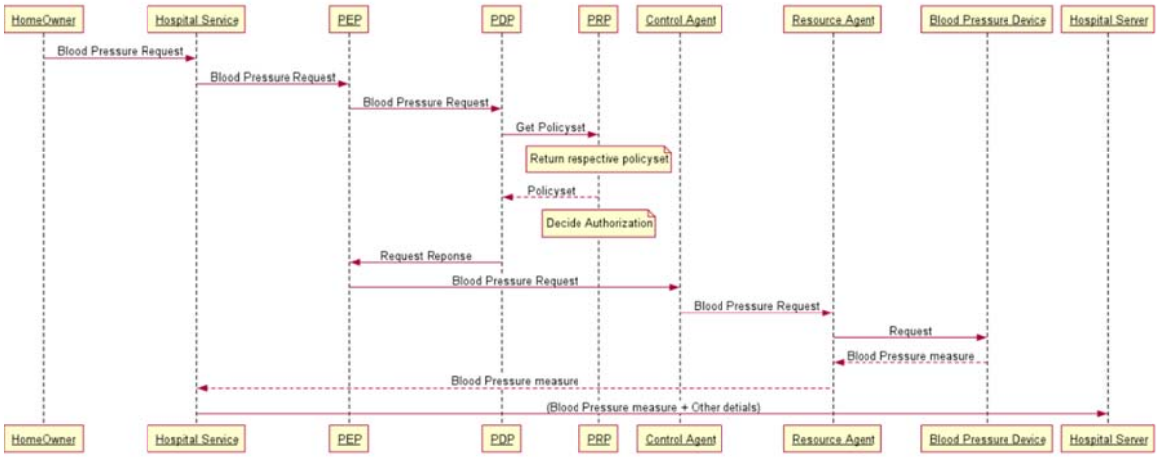


Figure 5-17: Sequence diagram for remote patient monitoring

Step 1: The homeowner uses the patient side OSGi service installed in the residential gateway to take their blood pressure measurement using a connected blood pressure

monitor. The service sends an action request to the access control component with information such as the origin service name, target blood pressure device, and the intended action to perform, i.e. to take a blood pressure measurement.

Step 2: The PEP receives the request from the installed service. The PEP decides whether the action should be approved or not. To do this PEP converts the request to XACML format and sends it to the PDP to make the decision. Finally, the PDP checks the database for a matching and sends the decision back to the PEP. For the purpose of this description, we will assume that the policy approves the action request.

Step 3: Once the decision is made, the PEP forwards the action request to the control agent bundle in the resource management component. The control agent in turn forwards the request to the respective resource agent that is connected with the blood pressure monitor to execute the action. The resource agent in turn executes the requested action to take a blood pressure measurement.

Step 4: Once the action is executed, the resource agent responsible for the blood pressure monitor sends the blood pressure value back to the patient side OSGi service (or an error message if any). The patient side OSGi service in turn displays the blood pressure value to the patient via the client application.

Step 5: Once the homeowner approves the measure, the patient side OSGi service establishes secure communication with the hospital's server to send the blood pressure measurement in XML format to the server using the HTTPS protocol. When the patient side OSGi service sends the medical data, it also adds extra information for management purposes such as the patient's identifiers, other medical data related information, etc.

Step 5: The hospital's server receives the medical data and stores it. Using the patient identifiers the healthcare professionals can access the relevant patient data to visualize and monitor their patient. It is assumed that the healthcare provider develops this part of implementation and it is outside the score of this thesis.

Inference: This use case detailed how a patient & homeowner could measure their blood pressure and utilize their residential gateway to send the measurement to the hospital's server. As security is an important and basic requirement for any health care system, the residential gateway provides a secure way for collecting the medical data and securely transmitting this data to the hospital's server. Only an *authenticated* homeowner using an *authorized* service can access the medical device to take a measurement. Therefore, even another service installed in the same residential gateway cannot access this medical device or the data without having permission to do so via an access control policy. As the access control policies for the services are composed by the service providers and validated by the gateway operator we will assume that domain experts verify the policies. By using HTTPS for the communication between the residential gateway and the hospital's server the security of the medical data is also guaranteed.

The Telecare service provided by the residential gateway are not restricted to simply taking measurements, but could also provide healthcare automation services. An automation use case similar to that described in the previous use case could be implemented for the Telecare domain to provide an automated Telecare service. For example, a homeowner using an automation service could add a policy to adapt the room temperature depending upon the skin temperature of the homeowner. Collecting the homeowner's temperature measure from a skin temperature monitor and changing the temperature for the relevant thermostat could achieve this.

5.5 Non Functional Use case analysis

The following subsection give a non-functional requirement analysis of the proposed residential gateway according to the home automation requirements given in Section 2.3. The table 5-1 show the overall comparison of the proposed design with the other similar works discussed in the background section.

5.5.1 Heterogeneity

Providing syntactic heterogeneity was one the primary requirement for the designed residential gateway. The resource management bundle exposes a device from device from a home automation domain in the same way as a device from a Telecare domain. The service provider who manufactured the connected device should ensure that the resource agent bundle they create for the device exposes interface for communication. If so, then any service installed in the residential gateway could use (subject to the access control policies) any device connected to the residential gateway, thereby providing an opportunity for service providers to develop heterogeneous services. The access control management component and the resource management component in the residential gateway facilitate (but control) this device level heterogeneity. To show the possible heterogeneity, a Telecare service provider could develop a service that uses a motion sensor together with other medical sensors to provide a remote patient monitoring service.

5.5.2 Extensibility

One of the main reasons for using OSGi in designing the residential gateway is the extensibility that the OSGi framework provides. Each service bundle installed in the residential gateway acts independently and could be installed or removed any point of time without disturbing other service bundles running in the gateway. In this way a homeowner could install any service bundle in the residential gateway and make it available to the other services installed in the gateway. As a first step to achieve heterogeneity, this extensibility provides an opportunity for the homeowner to install different types of services and different types of connected devices in a single residential gateway. Unlike the 2net hub and Insteon home automation hub the proposed residential gateway provides the possibility to install different types of services in the residential gateway.

5.5.3 Security and privacy

Providing heterogeneity and extensibility comes with the side effect of security and privacy concerns. The issues faced by the other systems described in Sections 2.4 to 2.8 were taken into consideration when designing the proposed residential gateway. In order to provide a basic level of security all communication with the residential gateway were made secure. Therefore, the homeowner client application communicates with the residential gateway using SSL/TLS to secure this communication. Similarly, every service provider must ensure that they implement a secure communication protocol when they communicate with the connected device and when they communicate with any external services or servers.

As a next step the access control management component was introduce to prevent access by unauthenticated and unauthorized users to services installed in the residential gateway. This component supports the ability to add fine-grained access control policies, thereby avoiding the coarse-grained access control problem faced by Samsung's

SmartThings system. Unlike the SmartThings system where once a service is granted access to one device function it gains complete control over the device, the proposed design allows for access control on per function level. So if a service is given access to switch OFF a device it does not necessarily provide access rights to switch it ON. With the help of the subscription component the level of granularity is also extended to the subscription process. As a result an installed service can only subscribed for a specific event from a device and will only get notifications about that specific event and not other notifications from the same device.

Service providers are given greater responsibility for managing the system. Moreover, these service providers are given the freedom to configure the policy set for their specific service so that domain experts write the policies. In turn to validate the access control policies the gateway operator should take the responsibility for verifying the policyset in order to address global security issues. The gateway operator is also responsible for verifying and distributing the service bundle before the homeowner can installs it in their residential gateway. In a similar way to Apple's appstore or Android's play store the gateway operator could establish a online service distribution centre from which homeowners can download a verified and validated service bundle directly to their residential gateway and install it. By verifying the bundle *before* using installing it in their residential gateway homeowner can avoid installing malicious service bundles developed by rouge service providers.

At present there are no gateway operators who provide a service distribution centre containing validated and verified service bundles. This remains an area for future work.

5.5.4 Context aware

A context management bundle installed in the resource management component provides the possibility for a service provider to develop context aware solutions. This component can use different types of component information, such as user based, location based, device based, and event based means to construct the context ontology. The context management bundle provides the ability to detect changes in the home environment and reason about them using the context engine. A service could use low level contextual information based on the information from sensors or other sources or use high level contextual information produced from other services based on low-level context information. It remains for future work to further develop the functions for context aware services.

5.5.5 Mobility

In order to connect with the residential gateway and use the services installed in it the homeowner can use a mobile application or web browser. This is made possible by running a web server OSGi bundle in the residential gateway. This web server is also used to display details of the services provided by the individual services running in the gateway. Additionally, the web browser or app could acts a communication medium for the homeowner when using these services. . However, there is not yet any standardization for what this communication or APIs for the interfaces, hence this is left for future work.

5.5.6 Usability

By using the RBAC method of access control, the administrator homeowner simply assigns roles to the other homeowners, thus avoiding the need for detailed configuration of access

control in the residential gateway. In this way, the task for the administrator homeowner is reduced.

The design makes it possible to install an interface bundle to control the services in the residential gateway. Each service provider has the freedom to develop this interface so that the homeowner can use the interface to use the installed service. However, there is not yet

Table 5-1: Overall summary with the proposed deisgn

	Samsung SmartThings	Insteon	2net Hub	Homer	Virtualized Residential Gateway	Proposed design
Heterogeneity	No	No	No	Yes	Yes	Service interoperability
Mobility	Yes	Yes	Yes	Partially implemented	Partially implemented	Could be connected remotely
Extensibility	No	No	No	Yes	Yes	Multiple services could be installed
Security and privacy	Authorization issues	Access issues	No known issues	No known issues	No known issues	Fine grained authorization, Secure communication
Usability	Using mobile Application	Using mobile application	Using mobile application	Using mobile application	Using mobile application	Using mobile application or web browser
Context aware	Location based Context	No	No	Location Based Context	No	Location, Event, User, Device based context

6 Conclusions and Future work

The following section discusses the conclusion for the thesis project. This is followed by a section that suggests possible future work that could be considered as an extension or a new line of research work.

6.1 Conclusions

Designing a residential gateway that supports a suitable set of requirements was the goal of this thesis project. The first step in this was to carefully define the requirements and evaluate existing efforts in terms of these requirements. This first step is given in Chapter 2. The proposed residential gateway design in the Chapter 4, specifically the components: access control component, automation component, and resource management component try to address the identified short coming from the requirement analysis of other systems.

The design suggests a list of components that would be necessary for a residential gateway to support different types of services and different types of users. The access control component using the XACML implementation allows for a fine-grained authorization mechanism for services from different service providers and for multiple types of homeowners. The automation component tries to provide a mechanism by which a homeowner can configure home automation rules for different types of connected devices. In addition, this component is also capable of providing automation using heterogeneous devices. Finally the resource component with its context aware bundles could provide an opportunity for service providers to develop context aware intelligent services. Although the complete design was not implemented or tested we believe that the proposed design should be considered a good base design for Acreo's home automation system.

Due to this thesis project, I was able to study different concepts in the home automation domain and details of a residential gateway. The deep study of currently available devices gave me a good idea about the existing technologies and their characteristics. Additionally, this thesis enabled me to explore different domains and understand their key concepts and how some of these ideas were implemented.

6.2 Future work

The proposed residential gateway design can manage home-related services. Such a residential gateway could be expanded to support additional features, such as building automation. The residential gateways of apartments of the same building could communicate with each other in order to provide building level functionality. This would be possible simply by installing a service bundle in each residential gateway to provide the building automation functionality or the residential gateway design could be changed to directly support building automation. To further extend the service provided by the residential gateway, the design could be extended to support the smart city concept. Residential gateways communicating to provide city level service could be an interesting research topic. Although the gateway could be used for building level or city level or higher-level service, the security and privacy of the individual homeowner should be given higher importance.

6.3 Limitations of this work

One of the primary limitations for the thesis work is the implementation of different modules for achieving the goal. Due to this limitation it is difficult to prove practically the advantages of the proposed home automation model compared to existing ones. The scope of the thesis work focused on providing home automation and remote monitoring services with example use cases. In order to extend the thesis work, different other services areas could be explored with respective use cases and changes could be made to the gateway design.

6.4 Required reflections

The thesis work tried to find the drawbacks of the existing commercially available gateways & research projects to propose a design that overcomes the drawbacks and satisfy the basic requirements for constructing a home automation based residential gateway. Improper implementation of the services areas discussed in this thesis could lead to privacy invasive solution for the homeowner, so extra effort should be taken while constructing the services and the gateway operator could validate the service before making it available to the homeowner. Finally, even if the technology permits for different services to be executed in one residential gateway, the service providers or device manufacturers could come forward and take more initiatives thereby the possibility of interoperability or service collaboration can be achieved.

References

[1] LG Electronics Australia Pty Ltd., “LG Split System Air conditioners.” LG Electronics Australia Pty Ltd., 15-Sep-2015.

[2] Home Gateway Initiative, “HG Requirements for HGI Open Platform 2.1,” Home Gateway Initiative, Report HGI-RD048v2, Mar. 2016.

[3] Department of Energy & Climate Chang, “Smart Meters: Quarterly Report to end March 2016, Great Britain,” U.K. Department of Energy & Climate Change, 3 Whitehall Place, London SW1A 2A, Statistical Release: Experimental National Statistic, Jun. 2016.

[4] “EDF Energy -Your smart meter display. Step by Step user guide,” *Your smart meter display - EDF Energy*. [Online]. Available: https://www.edfenergy.com/sites/default/files/smart_meter_data_guide.pdf. [Accessed: 03-Apr-2017].

[5] Qualcomm Life, Inc., “A Deployment-Ready Cellular Gateway Solution for Transmitting Medical Device Data to the Cloud.” Qualcomm Life, Inc., 2011.

[6] “Carenet-SE,” *Carenet-SE*, 2010. [Online]. Available: http://www.carenet-se.se/. [Accessed: 09-Feb-2017].

[7] ISO/IEC JTC 1/SC 25/WG 1, Interconnection of Information Technology Equipment, and Home Electronic System, “CD1 15045-01: Information technology — Interconnection of information technology equipment — Architecture for HomeGate, the residential gateway (AHRG),” ISO/IEC, Committee Draft 1 ISO/IEC JTC 1/SC 25/WG 1 N 912, Apr. 2000.

[8] F. Ding, A. Song, E. Tong, and J. Li, “A Smart Gateway Architecture for Improving Efficiency of Home Network Applications,” *Journal of Sensors*, vol. 2016, no. Article number 2197237, pp. 1–10, 2016.

[9] EU Member States and commented within the i2010 Subgroup on eHealth, “The Prague Declaration: eHealth 2009 Conference Declaration - eHealth for Individuals, Society and Economy.” EU Member States, 20-Feb-2009.

[10] Aaron Tilley, “Samsung-Owned SmartThings Launches New Smart Home Hub And Monitoring Service,” *Forbes*, 03-Sep-2015.

[11] “INSTEON Home Page,” *Insteon*. [Online]. Available: http://www.insteon.com/. [Accessed: 24-Apr-2017].

[12] INSTEON, “Whitepaper: The Details,” INSTEON, Version 2.0, 2013.

[13] Dag Spicer, “The ECHO IV Home Computer: 50 Years Later,” *Computer History Museum*, 31-May-2016. [Online]. Available: http://www.computerhistory.org/atchm/the-echo-iv-home-computer-50-years-later/. [Accessed: 09-Feb-2017].

[14] “Turn a home sweet home into a smart house.” [Online]. Available: http://www.atarimagazines.com/compute/issue134/98_Turn_a_home_sweet_ho.php. [Accessed: 03-Apr-2017].

[15] Siemens, “Building automation and control systems: System Catelog 2016,” Siemens Switzerland Ltd., Zug, Switzerland, Order number A55995-Q101, 2015.

[16] R. McDowall, *Fundamentals of HVAC systems*. Amsterdam: Elsevier, 2006.

[17] Siemens Industry, Inc., “Improving Performance with Integrated Smart Buildings,” Siemens Industry Inc., Buffalo Grove, IL, USA, White paper Order No. 153-BAU-053, 2012.

[18] A. Kailas, V. Cecchi, and A. Mukherjee, “A Survey of Communications and Networking Technologies for Energy Management in Buildings and Home Automation,” *Journal of Computer Networks and Communications*, vol. 2012, p. e932181, Mar. 2012.

[19] R. Carbou, M. Diaz, E. Exposito, and R. Roman, Eds., “Network Technologies,” in *Digital Home Networking*, John Wiley & Sons, Inc, 2011, pp. 17–57.

[20] G. O'Driscoll, "Moving IPTV Around the House," in *Next Generation IPTV Services and Technologies*, John Wiley & Sons, Inc., 2007, pp. 285–331.
[21] H. A. Latchman, S. Katar, L. Yonge, and S. Gavette, "Introduction," in *Homeplug AV and IEEE 1901:A Handbook for PLC Designers and Users*, Wiley-IEEE Press, 2013, p. 384-.
[22] "HomePNA Alliance." [Online]. Available: http://www.homepna.org/. [Accessed: 24-Apr-2017].
[23] M. over C. Alliance, "MoCA Technology for Your Home Network." [Online]. Available: http://www.mocainyourhouse.com/. [Accessed: 24-Apr-2017].
[24] F. Bouhafs, M. Mackay, and M. Merabti, "Communication Technologies for Smart Energy Management Systems," pp. 69–81, 2014.
[25] W. Lumpkins, "Home Automation: Insteon (X10 Meets Powerline) [Product Reviews]," *IEEE Consumer Electronics Magazine*, vol. 4, no. 4, pp. 140–144, Oct. 2015.
[26] V. Oksman and J. Egan, "Applications of ITU-T G.9960, ITU-T G.9961 transceivers for Smart Grid applications: Advanced metering infrastructure, energy management in the home and electric vehicles," International Telecommunication Union, Technical paper, Jun. 2010.
[27] C. Buratti, M. Martalo, R. Verdone, and G. Ferrari, *Sensor Networks with IEEE 802.15.4 Systems - Distributed*, 1st ed. Springer-Verlag Berlin Heidelberg, 2011.
[28] "Z-Wave Smart Home." [Online]. Available: http://www.z-wave.com/. [Accessed: 24-Apr-2017].
[29] O. Hersent, D. Boswarthick, and O. Elloumi, "Z-Wave," in *The Internet of Things*, John Wiley & Sons, Ltd, 2011, pp. 139–151.
[30] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," *Internet Request for Comments*, vol. RFC 4944 (Proposed Standard), Sep. 2007.
[31] J. Suhonen, M. Kohvakka, V. Kaseva, T. D. Hämäläinen, and M. Hännikäinen, *Low-Power Wireless Sensor Networks*. Boston, MA: Springer US, 2012.
[32] "Energy Harvesting Wireless Sensor Solutions and Networks from EnOcean." [Online]. Available: https://www.enocean.com/en/. [Accessed: 24-Apr-2017].
[33] J. Ploennigs, U. Ryssel, and K. Kabitzsch, "Performance analysis of the EnOcean wireless sensor network protocol," in *2010 IEEE 15th Conference on Emerging Technologies Factory Automation (ETFA 2010)*, 2010, pp. 1–9.
[34] C. Grimm, P. Neumann, and S. Mahlknecht, Eds., *Embedded Systems for Smart Appliances and Energy Management*, vol. 3. New York, NY: Springer New York, 2013.
[35] B. Jai, M. Ogg, and A. Ricciardi, "Effortless software interoperability with Jini #x2217; connection technology," *Bell Labs Technical Journal*, vol. 5, no. 2, pp. 88–101, Apr. 2000.
[36] "Apache River." [Online]. Available: https://river.apache.org/. [Accessed: 24-Apr-2017].
[37] S. Helal, "Standards for service discovery and delivery," *IEEE Pervasive Computing*, vol. 1, no. 3, pp. 95–100, Jul. 2002.
[38] "Universal Plug and Play Device Architecture - UPnP-arch-DeviceArchitecture-v1.0-20081015.pdf." [Online]. Available: http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0-20081015.pdf. [Accessed: 24-Apr-2017].
[39] "Bonjour protocol." [Online]. Available: http://hes-standards.org/doc/SC25_WG1_N1164.pdf. [Accessed: 24-Apr-2017].
[40] A. P. Robert, "Salutation Architecture: Enabling Applications and Services - A White paper," Salutation Consortium, Inc., White Paper, Aug. 1998.
[41] "The Salutation Consortium." [Online]. Available: http://salutation.org/. [Accessed: 24-Apr-2017].

[42] A. P. Robert, “Market Trends and Salutation Opportunities Review,” Salutation Consortium, Inc, Aug. 1988.
[43] E. Guttman, C. Perkins, J. Veizades, and M. Day, ‘Service Location Protocol, Version 2’, *Internet Request for Comments*, vol. RFC 2608 (Proposed Standard), Jun. 1999 [Online]. Available: http://www.rfc-editor.org/rfc/rfc2608.txt [Accessed: 24-Apr-2017].
[44] “Lifx smart bulb,” *LIFX*. [Online]. Available: https://www.lifx.com/. [Accessed: 24-Apr-2017].
[45] “Philips Hue - Hue motion sensor.” [Online]. Available: http://www2.meethue.com/en-us/productdetail/philips-hue-motion-sensor. [Accessed: 24-Apr-2017].
[46] “Meet the Nest Learning Thermostat | Nest.” [Online]. Available: https://nest.com/thermostat/meet-nest-thermostat/. [Accessed: 24-Apr-2017].
[47] “WeMo Crock-Pot® Slow Cooker | Crock-Pot®.” [Online]. Available: http://www.crock-pot.com/slow-cookers/wemo-enabled-smart-slow-cooker/crock-pot-6-quart.-smart-slow-cooker-with-wemo/SCCPWM600-V2.html. [Accessed: 24-Apr-2017].
[48] “Fiche descriptive de l’offre de fourniture d’électricité au tarif réglementé pour les particuliers,” Aug-2016. [Online]. Available: https://www.vialis.tm.fr/sites/default/files/pdf/energie/fiche_standard_elec_particuliers.pdf. [Accessed: 24-Apr-2017].
[49] “Wireless Blood Pressure Monitor | Blood Pressure Cuff.” [Online]. Available: http://www.withings.com/us/en/products/body-scale. [Accessed: 24-Apr-2017].
[50] T. K. L. Hui, R. S. Sherratt, and D. D. Sánchez, “Major requirements for building Smart Homes in Smart Cities based on Internet of Things technologies,” *Future Generation Computer Systems*.
[51] T. Perumal, A. R. Ramli, C. Y. Leong, S. Mansor, and K. Samsudin, “Interoperability among Heterogeneous Systems in Smart Home Environment,” in *2008 IEEE International Conference on Signal Image Technology and Internet Based Systems*, 2008, pp. 177–186.
[52] S. Moeller, K. P. Engelbrecht, S. Hillmann, and P. Ehrenbrink, “New ITG Guideline for the Usability Evaluation of Smart Home Environments,” in *Speech Communication; 11. ITG Symposium*, 2014, pp. 1–4.
[53] B. Schilit, N. Adams, and R. Want, “Context-Aware Computing Applications,” in *1994 First Workshop on Mobile Computing Systems and Applications*, 1994, pp. 85–90.
[54] “SmartThings.,” *SmartThings.com*. [Online]. Available: https://www.smartthings.com/. [Accessed: 24-Apr-2017].
[55] “SmartThings Developer Documentation - Release Latest.” Samsung, 22-Mar-2017.
[56] “List of all Officially Published Apps from the MORE category of Smart Setup in the Mobile App,” *SmartThings Community*. [Online]. Available: https://community.smartthings.com/t/list-of-all-officially-published-apps-from-the-more-category-of-smart-setup-in-the-mobile-app/13673. [Accessed: 25-Mar-2017].
[57] E. Fernandes, J. Jung, and A. Prakash, “Security Analysis of Emerging Smart Home Applications,” in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 636–654.
[58] “INSTEON Details Whitepaper Version 2 2005-2013,” White Paper.
[59] “Wireshark · Go Deep.” [Online]. Available: https://www.wireshark.org/. [Accessed: 26-Mar-2017].
[60] “INSTEON Hub Security Bypass Vulnerability.” [Online]. Available: https://tools.cisco.com/security/center/viewAlert.x?alertId=33393. [Accessed: 26-Mar-2017].

[61] Debra Gordon, “Telemedicine: Using Remote Monitoring to Reduce Hospital Readmissions,” *Milken Institute School of Public Health, The George Washington University*, 30-Oct-2015. [Online]. Available: https://mha.gwu.edu/blog/telemedicine-reduce-hospital-readmissions/. [Accessed: 09-Feb-2017].

[62] “Entra Health systems MyGlucoseHealth Wireless,” *Case Study Myglucosehealth*. [Online]. Available: http://www.qualcommlife.com/images/pdf/CaseStudy_MyGlucoHealth.pdf. [Accessed: 27-Mar-2017].

[63] Claire Maternaghan, “The Homer Home Automation System - Technical report.” Dec-2010.

[64] C. Maternaghan and K. J. Turner, “A Configurable Telecare System,” in *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, New York, NY, USA, 2011, p. 14:1–14:8.

[65] Y. Royon and S. Frenot, “Multiservice home gateways: business model, execution environment, management infrastructure,” *IEEE Communications Magazine*, vol. 45, no. 10, pp. 122–128, Oct. 2007.

[66] “Web Services Glossary.” [Online]. Available: https://www.w3.org/TR/ws-gloss/. [Accessed: 01-Apr-2017].

[67] M. P. Papazoglou and W.-J. van den Heuvel, “Service oriented architectures: approaches, technologies and research issues,” *The VLDB Journal*, vol. 16, no. 3, pp. 389–415, Jul. 2007.

[68] R. S. Hall and H. Cervantes, “Challenges in building service-oriented applications for OSGi,” *IEEE Communications Magazine*, vol. 42, no. 5, pp. 144–149, May 2004.

[69] “OSGi in Practice - osgibook_preview_20091217.pdf.” [Online]. Available: http://njbartlett.name/files/osgibook_preview_20091217.pdf. [Accessed: 01-Apr-2017].

[70] “IBM Knowledge Center - OSGi bundles.” [Online]. Available: https://www.ibm.com/support/knowledgecenter/was_beta/com.ibm.websphere.wdt.doc/topics/cbundles.htm. [Accessed: 01-Apr-2017].

[71] “Architecture – OSGi™ Alliance.” [Online]. Available: https://www.osgi.org/developer/architecture/. [Accessed: 01-Apr-2017].

[72] D. Marples and P. Kriens, “The Open Services Gateway Initiative: an introductory overview,” *IEEE Communications Magazine*, vol. 39, no. 12, pp. 110–114, Dec. 2001.

[73] G. O’Driscoll, *The Essential Guide to Home Networking*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.

[74] “Benefits of Using OSGi – OSGi™ Alliance.” [Online]. Available: https://www.osgi.org/developer/benefits-of-using-osgi/. [Accessed: 02-Apr-2017].

[75] D. M. Benantar, *Access Control Systems - Mandatory-Access-Control Model*. 2006.

[76] D. M. Benantar, *Access Control Systems - Discretionary-Access Control and the Access-Matrix Model*. 2006.

[77] D. M. Benantar, *Access Control Systems - Role-Based Access Control*. Springer US, 2006.

[78] “OASIS | Advancing open standards for the information society.” [Online]. Available: https://www.oasis-open.org/. [Accessed: 14-Apr-2017].

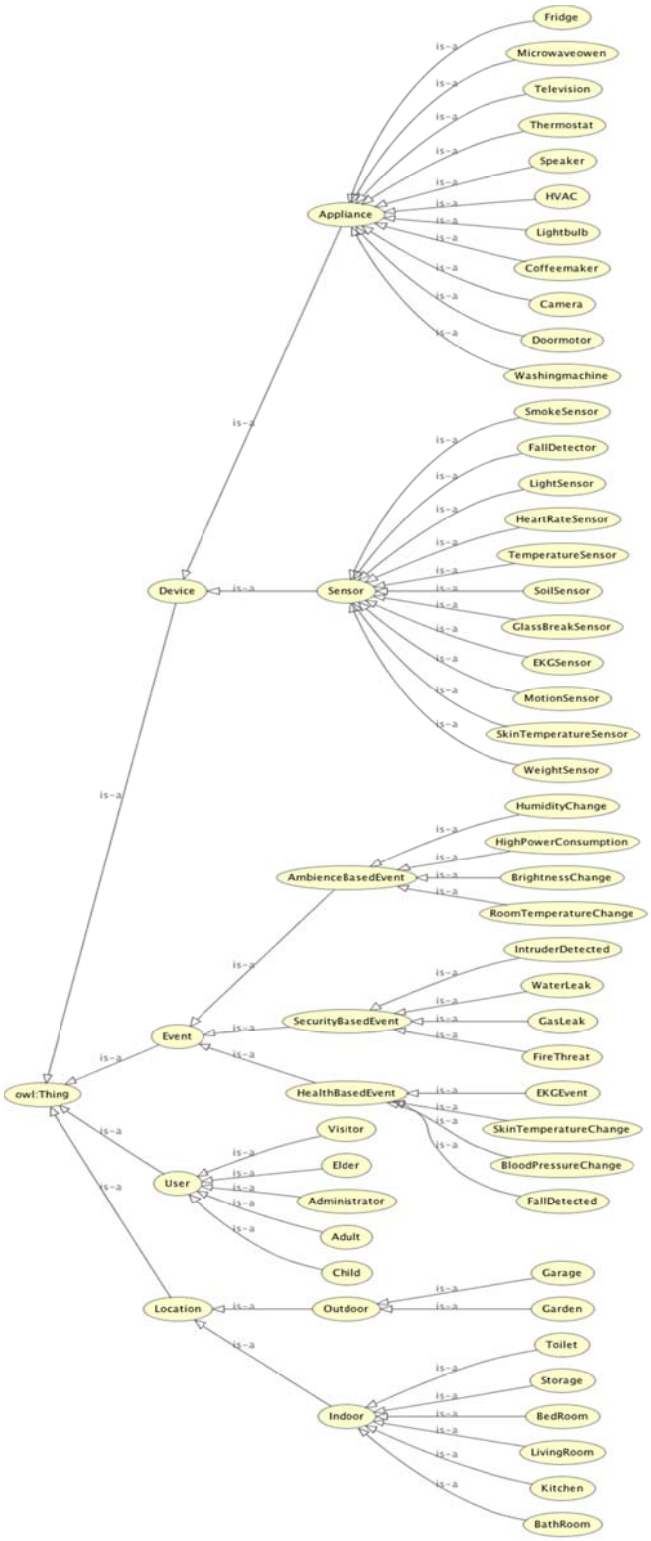
[79] E. Rissanen, “eXtensible Access Control Markup Language (XACML) Version 3.0 Committee Specification 01,” Aug. 2010.

[80] “XACML Reference Architecture.” [Online]. Available: https://www.axiomatics.com/blog/entry/xacml-reference-architecture.html. [Accessed: 14-Apr-2017].

[81] S. A. Brown, *Implementing Virtual Private Networks*. New York: McGraw-Hill Professional, 1999.
[82] P. Belimpasakis, S. Moloney, V. Stirbu, and J. Costa-Requena, “Home media atomizer: remote sharing of home content - without semi-trusted proxies,” *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 1114–1122, Aug. 2008.
[83] Z. Wei, J. Li, Y. Yang, and D. Jia, “A residential gateway architecture based on Cloud computing,” in *2010 IEEE International Conference on Software Engineering and Service Sciences*, 2010, pp. 245–248.
[84] P. Belimpasakis and V. Stirbu, “A survey of techniques for remote access to home networks and resources,” *Multimed Tools Appl*, vol. 70, no. 3, pp. 1899–1939, Jun. 2014.
[85] W. Colitti, K. Steenhaut, N. D. Caro, B. Buta, and V. Dobrota, “REST Enabled Wireless Sensor Networks for Seamless Integration with Web Applications,” in *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, 2011, pp. 867–872.
[86] M. Stamp, “Authentication,” in *Information Security*, John Wiley & Sons, Inc., 2011, pp. 227–264.
[87] “Apache Felix - Apache Felix Maven OSGi Plugin.” [Online]. Available: http://felix.apache.org/documentation/subprojects/apache-felix-maven-osgi-plugin.html. [Accessed: 26-May-2017].

Appendix

Complete Home automation ontology created using protégé



TRITA-ICT-EX-2017:48