PROMOTING PRIVACY IN
SMART SPACE

SECURE COMMUNICATION BETWEEN
SMART HOME AND SMART METERING
SYSTEM ARCHITECTURES

MOHAMMADHADI MISAGH

Master of Science Thesis
Stockholm, Sweden 2013
XR-EE-LCN 2013:018
Privacy in Smart Space
Secure Communication Between Smart Home and Smart Metering System Architectures

Mohammadhadi Misagh

Master of Science Thesis XR-EE-LCN 2013:018
KTH Electrical Engineering
Lab of Communication Networks (LCN)
SE-100 44 STOCKHOLM
Secure Communication Between Smart Home and Smart Metering System Architectures

Mohammadhadi Misagh

<table>
<thead>
<tr>
<th>Approved</th>
<th>Examiner</th>
<th>Supervisor</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-10-14</td>
<td>Panos Papadimitratos</td>
<td>Panos Papadimitratos</td>
</tr>
<tr>
<td>Commissioner</td>
<td>{Name}</td>
<td>Contact person</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Panos Papadimitratos</td>
</tr>
</tbody>
</table>
The Smart Metering Systems are state of the art technologies and modern services integrated with legacy metering systems. These systems are one of the most important parts of smart grids. Smart Grids are electrical power infrastructures that make intelligent decisions about the state of the electrical power devices. Smart Grids provide a stable electrical power environment. Smart Grids support all functionalities of traditional power systems in addition a Smart Grid bears new functionalities of intelligent power systems e.g. self-healing, resisting to attacks, increasing power quality, motivating consumers to wisely use the electricity, enabling new business to electrical markets, optimizing assets and operating efficiently and accommodating all generation and storages options. A smart home is usually a modern home that is equipped with some special structured wiring or wireless systems to enable inhabitants to remotely control or program any of home electronic devices supporting specific types of control functionalities by entering a single command at their remote computer or smart phones. Privacy in smart spaces/environments is not only confidentiality of data/connections but is the management of the complication of these environments due to decentralised and dynamic nature of these spaces. General model of smart home and smart metering systems have their specific considerations e.g. assumptions, threats, countermeasures and privacy preserving model. Both security architectures have their specific security policies, considerations, authorization and authentication authorities, and encryption and decryption strategies. Running two different security architectures in parallel may raise some issues, threats and considerations that are the main purpose of this thesis project. So, the principle goal of this research project is to define the secure model of collaboration between two different security architectures in the smart space in which general smart home network architecture is established in parallel to smart metering system architecture. Federated authentication systems e.g. SAML and OAuth, OpenID and federated authorization systems like XACML are most common distributed standard protocols to manage distributed security systems. This thesis investigates issues and problems of two parallel security architectures and proposes a secure communication protocol in addition to a proof-of-concept of the final solution. All of most famous available protocols for authentication and authorization will be discussed and finally will be implemented as a proof of concept. Encryption and decryption as basic security functions is not considered in this project and will be postponed as a future work around this research project.

Keywords
Smart Grid, Smart Home, XACML, SAML, OAuth, OpenID
FOREWORD

This master thesis report was written during the period of spring 2012 until summer 2013, under the supervision of Prof. Panos Papadimitratos in LCN (Lab of Communication Networks) in the school of Electrical Engineering at KTH. First of all I like to gratefully appreciate my supervisor who guides me through this research thesis. He has guided me all the way from the beginning towards the end. Then I would like to thank Ph.D. students who helped me to understand the problems better during the discussions we had, Nikolaos Alexiou and Stylianos Gisdakis for the excellent feedback and comments on the early versions of the design. Moreover, I would like to appreciate my friend, Anders Hansson from Cryptzone for all the interesting discussions we had in design and implementation of secure distributed authorization and authentication systems. Finally, I would also like to dedicate my heartfelt appreciation to my lovely parents, Zabi and Banoo, for all of their supports during my whole life.

Mohammadhadi Misagh

Stockholm, October 2013
**Abbreviations**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL</td>
<td>Access Control List</td>
</tr>
<tr>
<td>AP</td>
<td>Asserting Party</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial of Service</td>
</tr>
<tr>
<td>EPAL</td>
<td>Enterprise Privacy Authorization Language</td>
</tr>
<tr>
<td>HAN</td>
<td>Home Area Network</td>
</tr>
<tr>
<td>IdP</td>
<td>Identity Provider</td>
</tr>
<tr>
<td>IOI</td>
<td>Item of Interest</td>
</tr>
<tr>
<td>LMN</td>
<td>Local Metrological Network</td>
</tr>
<tr>
<td>MAC</td>
<td>Message Authentication Code</td>
</tr>
<tr>
<td>MITM Attack</td>
<td>Man-In-The-Middle Attack</td>
</tr>
<tr>
<td>OASIS</td>
<td>Organization for the Advancement of Structured Information Standards</td>
</tr>
<tr>
<td>OP</td>
<td>OpenID Provider</td>
</tr>
<tr>
<td>PAP</td>
<td>Policy Administration Point</td>
</tr>
<tr>
<td>PDP</td>
<td>Policy Decision Point</td>
</tr>
<tr>
<td>PEP</td>
<td>Policy Enforcement Point</td>
</tr>
<tr>
<td>PEV</td>
<td>Plug-in Electric Vehicle</td>
</tr>
<tr>
<td>Pip</td>
<td>Policy Information Point</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>RBAC</td>
<td>Role-based Access Control</td>
</tr>
<tr>
<td>RP</td>
<td>Relying Party</td>
</tr>
<tr>
<td>SAML</td>
<td>Security Assertion Markup Language</td>
</tr>
<tr>
<td>SHG</td>
<td>Secure Home Gateway</td>
</tr>
<tr>
<td>SMSG</td>
<td>Smart Metering System Gateway</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SP</td>
<td>Service Provider</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td>SSO</td>
<td>Single Sign on</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td><strong>WAN</strong></td>
<td>Wide Area Network</td>
</tr>
<tr>
<td><strong>XACML</strong></td>
<td>Extensible Access Control Markup Language</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>5</td>
</tr>
<tr>
<td>FOREWORD</td>
<td>6</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>8</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>10</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>18</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>18</td>
</tr>
<tr>
<td>1.1.1 Smart Homes</td>
<td>18</td>
</tr>
<tr>
<td>1.1.2 Smart Metering Systems</td>
<td>20</td>
</tr>
<tr>
<td>1.1.3 Privacy Terms and Definitions</td>
<td>22</td>
</tr>
<tr>
<td>1.2 Description of Research Area</td>
<td>24</td>
</tr>
<tr>
<td>1.3 Research Question and the purpose of the Research</td>
<td>25</td>
</tr>
<tr>
<td>1.4 Methodology</td>
<td>26</td>
</tr>
<tr>
<td>1.4.1 Explicate Problem</td>
<td>28</td>
</tr>
<tr>
<td>1.4.2 Outline Artefact and Define Requirements</td>
<td>28</td>
</tr>
<tr>
<td>1.4.3 Design and Develop Artefact</td>
<td>28</td>
</tr>
<tr>
<td>1.4.4 Demonstrate Artefact</td>
<td>28</td>
</tr>
<tr>
<td>1.4.5 Evaluate Artefact</td>
<td>29</td>
</tr>
<tr>
<td>1.5 Delimitations</td>
<td>29</td>
</tr>
<tr>
<td>2 FRAME OF REFERENCE</td>
<td>31</td>
</tr>
<tr>
<td>2.1 Smart Home Systems</td>
<td>31</td>
</tr>
<tr>
<td>2.1.1 Authentication in the Smart Home</td>
<td>31</td>
</tr>
<tr>
<td>2.1.2 Authentication Model for Remote Access</td>
<td>32</td>
</tr>
<tr>
<td>2.1.3 Authorization in the Smart Home</td>
<td>33</td>
</tr>
<tr>
<td>2.2 Smart Metering Systems</td>
<td>36</td>
</tr>
<tr>
<td>3 FEDERATED AUTHORIZATION SYSTEMS</td>
<td>38</td>
</tr>
<tr>
<td>3.1 eXtensible Access Control Markup Language (XACML)</td>
<td>38</td>
</tr>
<tr>
<td>3.1.1 XACML Data Flow Model</td>
<td>38</td>
</tr>
<tr>
<td>3.1.2 XACML Policy Structure</td>
<td>40</td>
</tr>
</tbody>
</table>
3.1.3 XACML Authorization Request and Response ...................................................... 41
3.1.4 XACML Security and Policy Threats ................................................................. 42
3.1.5 XACML Security and Policy Safeguards .......................................................... 43
3.1.6 XACML Pros and Cons ....................................................................................... 43
3.2 Enterprise Privacy Authorization Language (EPAL) .................................................. 44
  3.2.1 EPAL Policy Structure ....................................................................................... 44
  3.2.2 EPAL Authorization Request and Response ..................................................... 46
  3.2.3 EPAL Pros and Cons ....................................................................................... 47
3.3 Discussion about Federated Authorization Systems .................................................. 47
4 FEDERATED AUTHENTICATION SYSTEMS .................................................................. 49
  4.1 Security Assertion Markup Language (SAML) .................................................... 49
    4.1.1 SAML Profiles ................................................................................................ 51
    4.1.2 SAML Bindings ............................................................................................. 52
    4.1.3 Security in SAML ......................................................................................... 53
    4.1.4 Privacy in SAML ......................................................................................... 53
    4.1.5 SAML Pros and Cons .................................................................................... 54
  4.2 OpenID .................................................................................................................. 54
    4.2.1 OpenID Protocol Overview .......................................................................... 55
    4.2.2 OpenID Security Considerations ................................................................... 55
    4.2.3 OpenID Pros and Cons ................................................................................ 56
  4.3 Open Authorization (OAuth) ............................................................................... 56
    4.3.1 OAuth Protocol Overview ............................................................................ 57
    4.3.2 OAuth vs. OpenID ......................................................................................... 58
    4.3.3 OAuth Pros and Cons .................................................................................. 59
  4.4 Discussion about Federated Authentication Systems ............................................. 59
5 FUNCTIONAL REQUIREMENTS .................................................................................... 61
  5.1 Appliance/Device Categories .............................................................................. 61
  5.2 Use cases or Functional Requirements ............................................................... 62
    5.2.1 Use Case/Scenario #1: Smart Home Zone .................................................. 62
    5.2.2 Use Case/Scenario #2: Smart Metering Zone .............................................. 63
    5.2.3 Use Case/Scenario #3: Common Zone ......................................................... 64
  5.3 Actors or External Entities .................................................................................... 66
LIST OF FIGURES

Figure 1. General home network model for security (ITU-T Study Group 17, 2007) ............... 19
Figure 2. Smart Metering system (Federal Office for Information Security, 2011)............. 21
Figure 3. SHG and SMSG Security Architectures............................................................. 25
Figure 4. Overview of the Design Science Method (Johannesson & Perjons, 2012)........... 27
Figure 5. Authentication Model for Remote Access ....................................................... 32
Figure 6. Authorization Entities for Home Network ......................................................... 33
Figure 7. Conceptual policy model for authorization (ITU-T Study Group 17(Authorization), 2008) ................................................................................................................. 34
Figure 8. XACML Data Flow Diagram (Moses, 2005)...................................................... 39
Figure 9. Policy Language Model (Moses, 2005)............................................................. 41
Figure 10. XACML Context (Moses, 2005)...................................................................... 42
Figure 11. A Sample of EPAL Authorization Rule ............................................................ 45
Figure 12. Non-normative High-level UML Overview of an EPAL policy (Ashley, Hada, Karjoth, Powers, & Schunter, 2003) .......................................................................... 46
Figure 13. EPAL Authorization Request Message ............................................................ 46
Figure 14. EPAL Authorization Response Message ......................................................... 47
Figure 15. General Single Sign-On Use Case (N.Ragouzis & al, 2008)............................ 50
Figure 16. General Identity Federation Use Case (N.Ragouzis & al, 2008)....................... 50
Figure 17. Basic SAML Concepts (N.Ragouzis & al, 2008) .............................................. 50
Figure 18. Differences in Initiation of Web Browser SSO (N.Ragouzis & al, 2008)......... 52
Figure 19. SP-Initiated SSO with Redirect and POST Bindings (N.Ragouzis & al, 2008) 53
Figure 20. OpenID 2.0 Protocol Flow ............................................................................ 55
Figure 21. OAuth 2.0 Authentication Protocol Flow (D. Hardt, 2012)............................. 57
Figure 22. OpenID vs. OAuth Authentication (Wikipedia, OAuth, 2013) ......................... 59
Figure 23. Abstract Model of SHG ................................................................................. 63
Figure 24. Abstract Model of SMSG ............................................................................. 64
Figure 25. General Model of Collaboration of SHG and SMSG ...................................... 65
Figure 26. WSO2-based Gateway Architecture (User Stories #1, 2)............................... 72
Figure 27. WSO2-based Two Gateways Architecture (Use case #3)............................. 72
Figure 28. SAML Identity Provider Settings ................................................................... 74
Figure 29. SAML Authentication Request .................................................................... 74
Table 1. Communication Flows between Devices within Networks (Federal Office for Information Security, 2011) ................................................................. 22
Table 2. Design-Science Guidelines (Alan R. Hevner, 2004) .................................................. 26
Table 3. Relationship between Threat Resistance and SALs .................................................. 32
Table 4. OAuth Authorization Grant Phase ........................................................................... 58
Table 5. Identity Types and Their Attributes ........................................................................ 60
Table 6. Home Appliance Catagorization ............................................................................. 61
Table 7. Actors in Smart Metering System .......................................................................... 66
Table 8. SHG Turn on Policy ............................................................................................... 77
Table 9. Secure Home Gateway Web Application Settings .................................................. 91
Table 10. Roles and Users .................................................................................................... 92
Table 11. SAML SSO Settings ......................................................................................... 92
Table 12. OpenID Providers (Wikipedia, OpenID, 2013) .................................................... 93
Table 13. OAuth Application Settings .................................................................................. 95
Table 14. OAuth Identity Server Auto-generated Settings .................................................... 95
Table 15. Secure Home Gateway Policy Set (Turn on) ........................................................ 96
Table 16. Secure Home Gateway Policy Set (Turn off) ....................................................... 96
Table 17. Smart Metering Gateway Policy Set ...................................................................... 97
1 INTRODUCTION

This chapter describes background of the research project including smart home systems and smart grid/metering systems, description of the research, research question, the goal of the research, methodology of the research and finally the limitations of the thesis project.

1.1 Background

In the first part of the report, Smart Homes and Smart Metering Systems will be introduced briefly. In addition Privacy related terms and definitions will be presented. Description of Research area and research question will be developed at the end of this section.

1.1.1 Smart Homes

“A smart home or building is a home or building, usually a new one that is equipped with special structured wiring to enable occupants to remotely control or program an array of automated home electronic devices by entering a single command. For example, a homeowner on vacation can use a touchtone phone to arm a home security system, control temperature gauges, switch appliances on or off, control lighting, program a home theatre or entertainment system, and perform many other tasks”. (Redriksson, n.d.)

Electronic communication is coming to our houses day by day. And Networking of devices is rapidly growing, and today management and controlling devices is faster, easier and more straightforward. Network of communication devices, entertainment devices, kitchen appliances and so on is making our life more comfortable. Network of communication is based on different technologies, either wired or wireless technologies.

Wired communication usually is based on a technology known as Power line Carrier Systems (PCS). PCS is using the current power lines in the home to send some specific signals to devices. Any command comprises address of the device and a request or command. For example, you can send turn on or turn off command to the devices or get the status of devices in reverse. X10 is most important protocol in this technology area using the short radio frequency (RF) bursts that represent digital information, enable communication between transmitters and receivers.

Wireless technology is another way to control home appliances. Wireless communication is everywhere and is growing drastically. Wi-Fi, ZigBee and Z-Wave are three of most common wireless protocols in smart homes. These protocols are supporting two way communications so not only you can send command messages to devices or appliances but you can get the status, data and any other requested information from them.

A general home network model for security is shown in Figure 2. This general model defines all collaborative components in the smart home, their relationships and all details of their position in the environment.
There are seven entities in this model: Remote user, Remote terminal, Application server, secure home gateway, Home application server, Home user and Home devices.

Some of the most important terms of smart home technology are as defined below (ITU-T Study Group 17, 2007):

**Secure Home Gateway (SHG):** “A secure home gateway is a kind of residential gateway seen from the point of view of security, and a point or an entity which forwards data packets from open network to internal home network or vice versa, changes security parameter or communication protocol from home network to open network or vice versa, and can perform security-related functions, such as packet filtering, intrusion detection, and policy management function and so on, according to a given security policy. That is, a secure home gateway comprises more than only firewall”.

**Remote terminal:** “Remote terminal is an entity that has network access function and an Internet interface to connect or control the home devices in the home network”.

**Remote user:** “Remote user is an entity (person) outside the home network that uses and operates the remote terminal for accessing the devices in the home network”.

Most important characteristics of a home network are listed below (ITU-T Study Group 17, 2007):

1. Various transmission mediums can be used for the home network.
2. Home network is a combination of a wireless network and a wired network.
3. There are many environments from the security point of view.
4. Remote terminals are carried around by remote users.
5. There are various types of home network devices requiring different levels of security.

Some of the most important security requirements for home network model related to the scope of this research are (ITU-T Study Group 17, 2007):

1. All the network elements e.g. a remote terminal, a secure home gateway and home devices should preserve their sensitive data in a secure manner, and should prevent their assets against unauthorized access, unauthorized modification, or unauthorized deletion.
2. The remote terminal should have an appropriate capability for user authentication like user name/password and/or biometric and key management.
3. The secure home gateway should have security functions, including user authentication, key management and so on.

1.1.2 Smart Metering Systems

“Nowadays, Smart Grid is as a growing infrastructure of state of the art technologies, services and entities integrating with legacy organization and systems. Smart Grid is an electrical power infrastructure that makes intelligent decision about the state of the electrical power system to maintain a stable environment. Most important characteristic of a Smart Grid is not only supporting the functionality of current power systems but bearing new functionalities of intelligent power systems e.g. self-healing, resists attack, increase power quality, motivates and includes consumers, enables electrical markets, optimizes assets and operates efficiently and accommodates all generation and storages options” (SGIP-CSWG, 2010).

“Smart Grid considerably increases the amount of data that can be monitored, collected and aggregated and analysed” (SGIP-CSWG, 2010). This information of smart grid is not only comes from the power consumption but is related to the consumers, so it raises the privacy concerns of the Smart Grids. Smart Grids comprises smart meters, smart appliances and other smart devices. The privacy risks are broadening in all sides.

One of the most important components in smart grid is Smart Metering System (SMS). This Component is responsible to gather power consumption information in the customer’s home or buildings and send it to the utility provider administration end points, all of this information make the basis for the customer billing or payments.

An abstract functional model of smart metering system is shown in Figure 2:
The Gateway connects three different network areas to each other:

1. A Wide Area Network (WAN) or external world to a network of devices including one or more smart metering devices (Local Metrological Network, LMN).
2. The consumer’s Home Area Network (HAN), which hosts Controllable Local Systems (CLS) to outside of the home network (WAN).

The Gateway is not only the communication handler between the components in the consumer’s LAN and the outside world but it can be seen as a special kind of firewall dedicated to the smart metering functionality. The gateway oversees the requests from inside of home to the outside and outside of home to internal part of home and decides to give the access permissions to the requestor. In addition, the gateway collects, processes and stores the records from Smart Meter(s) and ensures that only authorised parties have access to them. “The Meter itself records the consumption or production of one or more commodities (e.g. electricity, gas, water, heat) in defined intervals and submits those records to the Gateway. Controllable Local Systems (CLS) may range from local power generation plants, controllable loads such as air condition and intelligent household appliances “white goods” to applications in home automation. CLS may use the services of the Gateway for communication services. However, CLS are not part of the Smart Metering System” (Federal Office for Information Security, 2011).

The communication flows that are enforced by the Gateway between parties in the HAN, LMN and WAN are summarized in Table 1:
Table 1. Communication Flows between Devices within Networks (Federal Office for Information Security, 2011)

<table>
<thead>
<tr>
<th></th>
<th>WAN</th>
<th>LMN</th>
<th>HAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAN</td>
<td>see following list</td>
<td>No connection establishment allowed</td>
<td>No connection establishment allowed</td>
</tr>
<tr>
<td>LMN</td>
<td>No connection establishment allowed</td>
<td>see following list</td>
<td>No connection establishment allowed</td>
</tr>
<tr>
<td>HAN</td>
<td>Connection establishment is allowed to trustworthy, pre-configured endpoints and via an encrypted channel only</td>
<td>No connection establishment allowed</td>
<td>see following list</td>
</tr>
</tbody>
</table>

The Gateway shall separate devices in the LAN of the consumer from the WAN and shall enforce the following information flow control to control the communication between the networks that the Gateway is attached to (Federal Office for Information Security, 2011):

1. Only the Gateway or devices in the HAN may establish a connection to an external entity. Connection establishment by an external entity in the WAN or a Meter in the LMN is not possible.
2. The Gateway can establish connections to devices in the LMN or in the HAN.
3. Meters in the LMN are only allowed to establish a connection to the Gateway.

For communications within the different networks the following assumptions are defined (Federal Office for Information Security, 2011):

1. Communications within the WAN are not restricted. However, the Gateway is not involved in this communication,
2. No communications between devices in the LMN are assumed. Devices in the LMN may only communicate to the Gateway and shall not be connected to any other network,
3. Devices in the HAN may communicate with each other. However, the Gateway is not involved in this communication. If devices in the HAN have a separate connection to parties in the WAN (beside the Gateway) this connection is assumed to be appropriately protected.

1.1.3 Privacy Terms and Definitions

Privacy has different definition in the computer security area, Warren and Brandies say the privacy is “the right to be left alone” (Waren & Brandies, 1890), Alan Westin defines privacy as “Privacy is the claim of individuals, groups and institutions to determine for themselves, when, how and to what extent information about them is communicated to others” (Westin, 1967).

Roger Clark believes that privacy considerations have four dimensions (SGIP-CSWG, 2010):

1. Privacy of personal information
2. Privacy of the person
3. Privacy of personal behaviour
4. Privacy of personal communication
Anonymity is a vague term in the world of security. Many definitions deal with the simple case of a sender and a message, and discuss “anonymity” in terms of not being able to link a given sender to a sent message, or a message back to a sender (Pfitzmann.Ed, 2010). Related terms to the anonymity are linkability, detectability, observability, and pseudonymity. At the rest of this section, some of the most acceptable definition of privacy terms will be defined (Pfitzmann.Ed, 2010):

1. “Anonymity of a subject means that the subject is not identifiable within a set of subjects, the anonymity set”.
2. “Unlinkability of two or more items of interest (IOIs, e.g., subjects, messages, actions...) from an attacker’s perspective means that within the system (comprising these and possibly other items), the attacker cannot sufficiently distinguish whether these IOIs are related or not”.
3. “Undetectability of an item of interest (IOI) from an attacker's perspective means that the attacker cannot sufficiently distinguish whether it exists or not”.
4. “Unobservability ensures that a user may use a resource or service without others, especially third parties, being able to observe that the resource or service is being used”.
5. “A pseudonym is an identifier of a subject other than one of the subject’s real names. Pseudonymity is the use of pseudonyms as identifiers”. There are different types of Pseudonymity, e.g., sender pseudonymity and receiver pseudonymity. Regarding the context of usage, there are different type of pseudonyms including: Person pseudonym, Role pseudonym, Relationship pseudonym and Role-Relationship pseudonym.

**Person pseudonym:** “A person pseudonym is a substitute for the holder's name which is regarded as representation for the holder's civil identity. It may be used in many different contexts, e.g., a number of an identity card, the social security number, DNA, a nickname, the pseudonym of an actor, or a mobile phone number” (Pfitzmann.Ed, 2010).

**Role pseudonym:** “The use of role pseudonyms is limited to specific roles, e.g., a customer pseudonym or an Internet account used for many instantiations of the same role "Internet user". The same role pseudonym may be used with different communication partners. Roles might be assigned by other parties, e.g., a company, but they might be chosen by the subject himself/herself as well” (Pfitzmann.Ed, 2010).

**Relationship pseudonym:** “For each communication partner, a different relationship pseudonym is used. The same relationship pseudonym may be used in different roles for communicating with the same partner. Examples are distinct nicknames for each communication partner. In case of group communication, the relationship pseudonyms may be used between more than two partners (Pfitzmann.Ed, 2010)”.

**Role-relationship pseudonym:** “For each role and for each communication partner, a different role-relationship pseudonym is used. This means that the communication partner does not necessarily know, whether two pseudonyms used in different roles belong to the same holder. On the other hand, two different communication partners who interact with a user in the same role, do not know from the pseudonym alone whether it is the same user. As with relationship pseudonyms, in case of group communication, the role-relationship pseudonyms may be used between more than two partners” (Pfitzmann.Ed, 2010).
1.2 Description of Research Area

Security in smart space e.g. home, buildings, organizations is considered in different standard security models like Framework of security technologies for home network (ITU-T Study Group 17, 2007) and protection profile for the gateway of smart metering (Federal Office for Information Security, 2011). These standards and frameworks propose some security models preserving privacy of customers and any other individuals in the environment. These models generally divide internal space from external environment and proposed a security gateway as a mediator between internal smart space and external environment. Privacy in smart spaces or environments is not only confidentiality of data and/or connections but is the management of the complexity of these environments due to decentralised and dynamic nature of these spaces. General model of home network and smart metering system model have their specific considerations e.g. assumptions, threats, countermeasures and privacy preserving models.

In any smart space e.g. smart building or smart house is being protected from attackers using these general models of home network and smart metering system, there are two operational infrastructures running in parallel. Any infrastructure includes a security gateway, authentication and authorization system, encryption/decryption system and any other security-related systems. Access permission to the devices or home appliance is one of the most critical issues when two different infrastructures are running simultaneously. Furthermore, it is highly required to establish a good communication system between two architectures to have a consistent operational environment. As an example, home owner could have a contract to electricity utility provider to turn on a specific device during the night when the price of electricity is minimum. On the other hand, there are some possible cases when one of inhabitants may wish to turn off that specific device in the night remotely via the smart home services, so clearly this kind of collisions may results an unsafe and unsecure case in these security models. Having consistent integrated communication may mitigate these security risks. Policies, authentication and authorization request and response messages are most important security data exchange between these two securities architectures. So, having a well-defined security communication to carry security data is a must for these inter-related security architectures.

This thesis investigates these two security architectures, i.e., smart home network security architecture and smart metering system security architecture models. Both security architectures have their specific security policy, authorization and authentication authorities, and encryption and decryption strategies. Two different security architecture are running in parallel has some potential security issues, threats and considerations that should be considered in which privacy and security of the smart space is a mandatory requirement. Policy collisions, security data (e.g. authorization and authentication data) exchange are most important issues and potential threats between these security architectures. Authentication and authorization is two major aspects of security in any security architecture, so it is necessary to define a good process of collaboration or framework to exchange this information between two running architectures. Defining a standard format of request and response messages for authentication and authorization and describing a well-defined process of collaboration is the most important security requirement for this research. Distributed authentication and authorization and federated authentication are most common solutions in this area that will be discussed at the rest of this report.
1.3 Research Question and the purpose of the Research

The abstract model of collaboration in smart space in which two different security architectures i.e. Secure Home Gateway (SHG) and Smart Metering System Gateway (SMSG) are running is shown in Figure 3. Each gateway is a part of whole security architecture that support different security functions including authentication, authorization and encryption systems. Remote user or internet users could be desktop user, mobile user and so on. Devices are any home appliance with different functional services like turn on/off, reload, restart, standby and so on. These functions of home devices could be categorized in three types:

1. Type one: these functions are controlled only by secure home gateway (SHG) in SHG’s control domain (green devices, these devices are completely supervised by SHG but in real case any function of device could be controlled by the gateway), for example low-power devices may not be considered in smart grid but it is necessary to have a granular control for each function of devices.

2. Type two: these functions are controlled only by smart metering gateway in the smart grid. These functions are controlled as a subsection of smart metering gateway control domain (Blue devices). For example reload, restart, hibernate and standby functions are controlling function that are controlled only by smart metering gateway.

3. Type three: these functions are controlled by two gateways. The intersection of control domains of two gateways (Yellow devices). As an example turn on/off major devices or high-power devices that have a high rate of power consumption. These functions are considered by two gateways. Usually these devices (functions) should be listed in the contract of power consumers and power utility providers.

Regarding to security perspective, having a safe and secure model of collaboration in which two security architectures are running simultaneously in the environment is a critical requirement that should be considered in smart space. In this research, the security of smart space e.g. smart home, smart building where two security architectures i.e. smart metering system and secure
home gateways are running in parallel is the key research area. The main purpose of this thesis is to define the a secure model of collaboration between two different security architectures in smart spaces in which the general smart home architecture is established in parallel to smart metering system architecture. In the other word, how the security information like security policies, authentication and authorization data can be exchanged between two different securities architectures. The research question is how to exchange confidential data like policies, authentication and authorization data of two security architectures e.g. general home network and smart metering system in the smart spaces securely?

The purpose of research is providing a general secure model/protocol of collaboration between two security architectures i.e. smart home and smart metering systems. This model (protocols set) is mainly dealing with how to exchange the authentication and authorization data between two gateways. The output of this research could be of interest of home owners who are interested in a secure and privacy preserving smart home, and also power utility providers that are responsible to provide privacy preserving environment for their consumers. Final goal of the research is to provide a proof of concept (POC) demonstration of this model supporting secure communication of two gateways.

1.4 Methodology

In this section, the methodology of this research project will be described. This research is a kind of problem solving project and the final target of the project is finding a solution. “Two paradigms characterize much of the research in the Information Systems discipline: behavioural science and design science. The behavioural science paradigm seeks to develop and verify theories that explain or predict human or organizational behaviour. The design-science paradigm seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artefacts”. (Alan R. Hevner, 2004)

Design-science is a problem solving paradigm, which is the root of innovative artefacts in information systems based on current ideas, practices, products and technical capabilities in an effective and efficient design, implementation and management process (Alan R. Hevner, 2004). In the area of communication and collaboration of smart home and smart grid areas, there is not a well-designed model of interaction. It is a great problem for future of gas, electricity and water ecosystems. Design-science paradigm is applied to develop a privacy-security preserving model of communication between smart home and smart metering system gateways.

The design-science framework defines a combination of processes, artefacts and guidelines. There are two processes in this framework, build and evaluate. And also, four design artefacts e.g. of constructs, models, methods, and instantiations. In addition seven guidelines as described in the following section:

<table>
<thead>
<tr>
<th>Guideline</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guideline 1: Design as an Artefact</td>
<td>Design-science research must produce a viable artefact in the form of a construct, a model, a method, or an instantiation.</td>
</tr>
<tr>
<td>Guideline 2: Problem Relevance</td>
<td>The objective of design-science research is to develop technology-based solutions to important and relevant business</td>
</tr>
</tbody>
</table>
Guideline 3: Design Evaluation

The utility, quality, and efficiency of a design artefact must be rigorously demonstrated via well-executed evaluation methods.

Guideline 4: Research Contributions

Effective design-science research must provide clear and verifiable contributions in the areas of the design artefact, design foundations, and/or design methodologies.

Guideline 5: Research Rigor

Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artefact.

Guideline 6: Design as a Search Process

The search for an effective artefact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.

Guideline 7: Communication of Research

Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

According to (Johannesson & Perjons, 2012), design science method includes five activities from problem investigation to artefact demonstration and evaluation. At the rest of this section all of the activities will be described in more detail. Activities of the design science method are shown in Figure 4:

One of the most important aspect of the design science method is the back and forth behaviour of this paradigm. Although design science is a sequential process but it is actually a back and forth process between different phases. For example in the design and development phase, the developers can request a new change in the requirement based on technological, technical, social issues arises in the organization. Arrows in the figure 4 is not the temporal sequence but is the input relationship.

Data generation is the next important part of the research methodology. Five of the most important data generation methods for any scientific research are interviews, group discussions, questionnaire, observation studies and document studies.
1.4.1 Explicate Problem
This activity is intended to investigate and elaborate the problem. The problem must be of general interest not a specific case or scenario, because the design science is about general problem solving. This activity starts with literature review on the target domain. At the beginning, the target of this research was privacy preserving solutions for smart grid and smart home. The output of this activity is a problem statement that is described in previous section. In this research the group discussion and document study are the method of data generation at the explicate problem phase. Reading several articles, scientific papers and standards on secure home network and smart metering system architecture in addition to group discussion with my supervisor and his assistants about the current situation and potential gap in these areas guided me to a bold ambiguity in the communication model of these gateways, in the other words, it was not clear how these two architectures could collaborate to each other and how these architectures manage information security data exchange. This problem is not a specific case in the world and this scenario could be generalized to any other inter-gateway communication. The result of this activity is summarized in the section 1.

Based on the studies and several meeting with security experts in this domain, the general problem appeared. How you can exchange data between two gateways in a privacy preserving approach especially for security related data including:

- Authentication and authorization in distributed environment and,
- Policy enforcement, Policy exchange and inter-gateway negotiation.

1.4.2 Outline Artefact and Define Requirements
Requirement elicitation will be considered in this phase, so the problem statement should be transform to requests and requirement specification. The requirements could be functional and nonfunctional and also should be prioritized at the first step of requirement analysis and development decision making. In this research the functional requirements are authorization and authentication data exchange between two security architectures and security policy negotiation to prevent security policy enforcement collision in two gateways. Non-functional requirement is having a model of collaboration based on open source implementation, open standard protocol and well-structured and good documented product in the security market. In addition, performance acceptance and privacy preserving considerations in the collaboration model could be part of non-functional requirements. Requirements specification and artefact outline is defined in section 5.

1.4.3 Design and Develop Artefact
In this step proposed artefact in the requirement phase will be design and constructed. Sections 3, 4 are pre-requisite of the designing final solution and sections 6, 7 are the output of this activity, this section covers design and implementation of the project. In this step different authorization and authentication standards will be introduced and then most appropriate standards and protocols will be chosen for implementation. And finally the design and implementation will be explained with adequate details.

1.4.4 Demonstrate Artefact
This activity aims to bring the artefact in the real-life; it is a Proof of Concept (POC). In this phase at least one instance of real life will be illustrated by the developed artefact. In this project,
there is a POC demonstration for the communication of two different security gateways. This communication model consists of authentication and authorization data exchange in addition to functional demonstration of home device. Fourth step of design science process as demonstrate artefact activity requires setting up and configuring environment including all servers and database and developed applications. This step is intended to bring the artefact in a real life case as a proof-of-concept implementation; Appendix A is a full description of installation and configuration environment for demonstration of this research project.

1.4.5 Evaluate Artefact

This activity concentrates on the evaluation of artefact with defined requirements. The purpose is to specify how extent the artefact can solve different problem cases. Evaluate artefact is the process of examining the artefact with the requirement and determines how well the final artefact solved the detailed problem and fulfil the specified requirements. There are two way of evaluation, ex-ante and ex-post evaluation (Johannesson & Perjons, 2012). Ex-ante evaluation is the process of evaluation without using or running and ex-post evaluation is running or executing the artefact. The former method is based on interviews with expert and investigate their point of views on the artefact, and also the alternative way is researcher’s argument in the artefact documentations. This form of evaluation called informed argument (Johannesson & Perjons, 2012). The latter evaluation strategy is dependent to performing artefact to provide evaluators by results of the artefact. This method is stronger and more expensive rather than ex-ante evaluation. In this research ex-ante evaluation will be carried out in the section 8 and also by supervisors of the thesis project as domain experts.

1.5 Delimitations

In this research project, there are some limitations that should be considered in advance:

1. The first limitation for this research is, the topology of internal home network, communication between appliances and gateways are not the point of consideration. So it doesn’t matter which protocol is used in the home network e.g. X10, Wi-Fi, Zig Bee and Z-Wave.
2. The second limitation is on remote access protocols to the home services either smart home services or smart metering services. For example mobile devices, internet desktop devices and business to customer technologies do not care in this research.
3. For the purpose of implementation and POC demonstration, it is necessary to use not only open standard, but open source software architecture and implementations in the market for the final solution of the research.
4. The complexity of smart metering infrastructure is not in the scope of this research. For example there are different components beside smart metering system in the home/building including collectors, distributers and switches. So, network communications and smart grid’s internal protocols to provide different functionalities for smart grid are not in the scope of this research.
5. Home appliance/device identification standard, and their actions are not in the domain of the research. Furthermore, device identification standards or device categorization standards are not considered in this research. So, only an abstract core functionalities for home devices e.g. turn on/off, reload, standby and hibernate will be applied for this research.
2 FRAME OF REFERENCE

This chapter looks on the available researches in security areas of the secure home gateways and smart metering systems in detail. Having familiar with the current standards, protocols and security system in these areas are the essential purpose of this chapter.

2.1 Smart Home Systems

Smart Home systems have the same security requirements to other information systems including Authentication, Authorization, Encryption/Decryption, and so on. These functions are most common required security functions in the smart home. There are different standard and documentations about the security in smart home in different aspects and about any of security functions. This section summarizes current situation of security research in the smart home.

2.1.1 Authentication in the Smart Home

There are three models of user authentication in the home network (ITU-T Study Group 17(Authentication), 2007):

1. Remote access model.
2. Authentication model for communication within the home.
3. Access model from home to open network.

The first authentication model is in the scope of this research but second and third authentication models are not in the scope of this research, so they will not be considered at this research anymore.

On the other hand, there are three levels of authentication in the smart home (ITU-T Study Group 17(Authentication), 2007):

Level 1: This level of authentication is lightest and simplest model of authentication. This Level should be used only in wired network. This level of authentication is minimum level of security and vulnerable to man in the middle (MITM), eavesdropping and session hijacking.

Level 2: This Level of authentication is average level of security. This level is used for authentication in wireless network. Even though it is protected for eavesdropping and server impersonation but still is vulnerable for MITM and session hijacking. More cryptographic protection may mitigate the risk of MITM and Session hijacking.

Level 3: This is the highest level of secure authentication model. This model is intended to protect the communication of smart home against open network. This level is protected against any kind of attacks including eavesdropping, MITM and session hijacking. This level requires more level of cryptographic mechanisms rather than Level 1 and Level 2.

Regarding the resistance to threats of each level of authentication, Table 3 summarizes authentication levels and their resistances.
Table 3. Relationship between Threat Resistance and SALs  
(ITU-T Study Group 17(Authentication), 2007)

<table>
<thead>
<tr>
<th>Type of Resistance</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-line guessing resistance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Replay resistance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Eavesdropping resistance</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Server impersonation resistance</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>MITM resistance</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Session hijacking resistance</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Since the scope of this thesis research is limited to remote access of home resources via open network, so Level 3 authentication is required to fulfill the basic security requirement of authentication in the smart home.

2.1.2 Authentication Model for Remote Access

Remote access authentication model is shown in Figure 5. The Remote Terminal (RT) attempts to connect to back end services in the home e.g. Home Device (HD) or Home Application Server (HAS) via the Secure Home Gateway (SHG). RT is a client device including desktop computer, mobile device used to connect user to the home network and is used in authentication and authorization phases.

Figure 5. Authentication Model for Remote Access  
(ITU-T Study Group 17(Authentication), 2007)
The authentication process is divided into two stages. The first stage is authentication to secure the home gateway, and the second stage is authentication to backend services, either home device or home application server. First step is preparing a security key for the next stage. Next stage could be one of Level 1 (Wired network) or Level 2 (Wireless network).

There are different researches in the authentication of smart home, either within the home or remote access to the home. Some of the most impressive researches in this area are:

1. “User Authentication Using Neural Network in Smart Home network” (Shahbaz Zahr Reyhani, 2007): This article proposed two stage authentication, first stage is registration, and the second stage is authentication by user name and password. This research does not cover remote access authentication.

2. “Sentry@Home - Leveraging the Smart Home for Privacy in Pervasive Computing” (Susana Alcalde Bagüés, 2007): This article formulates a seamless embedded solution in smart home. This is a privacy preserving access to smart home. It is a context-aware and privacy-aware infrastructure in smart home. This article is only covering the authentication in smart home and has not any recommendation for remote access to the smart home.

3. “Security technologies based on a home gateway for making smart homes secure” (Geon Woo Kim, 2009): This article investigates authentication and authorization in smart homes. For authentication in smart home suggests two different ways for authentication in smart home for inter-domain and intra-domain environments. For single domain of smart home, most of common authentication methods e.g. user name-password based, biometric, and certificate based work properly but for multi domain smart home, it suggests an authentication mapping in smart home gateway.

### 2.1.3 Authorization in the Smart Home

The purpose of the authorization system in home network is to manage the access privilege to home devices when the user is authenticated at the first step of request. So the authorization policy is a combination of users, resources, or home devices and operation accesses or device functions access privilege. General model of authorization for smart home network is shown in Figure 6. This model consists of three major elements, Authorization Server (AS), Home Subject (HS) and Home Resource (HR) (ITU-T Study Group 17(Authorization), 2008).

![Figure 6. Authorization Entities for Home Network](image-url)
**Home Resource (HR):** is any kind of home appliance or home devices accessible for home users via wired or wireless communication. In the smart home this devices must provide some extra functions to be accessible for remote controls; for example, having some transmitter in wireless network or switches and controllers in wired home power network.

**Home Subject (HS):** is any inhabitant in the home or outside the home accessing to the HR via public network. For example landlords, home occupants, guests, home appliance service providers, maintenance services and so on.

**Authorization Server (AS):** AS is an entity that enforces authorization policies and manages the authorization policies information repository. It is the only authorization authority in the smart home. AS receives the access request to HR i.e. home devices from HS i.e. home users and decides to grant or deny access permission to the resources. The AS is responsible to perform two different functions, Policy Decision and Policy Enforcement. And usually there is a physical independent module for each function in the system. A conceptual policy model of authorization is shown in figure 8. Below is a brief description of authorization policy modules (ITU-T Study Group 17(Authorization), 2008):

- **Policy Decision:** Policy Decision Point (PDP) provides an authorization decision based the authorization policy. The authorization policy should cover every possible access within the service domain and have no conflicts in its policy.
- **Policy Enforcement:** Upon receiving a request from HS, Policy Enforcement Point (PEP) controls access based on the decision by PDP. PEP may be executed separately from PDP, where an additional way for securing communication between the two entities is required.

The conceptual policy model for authorization is:

![Conceptual policy model for authorization](image)

Figure 7. Conceptual policy model for authorization (ITU-T Study Group 17(Authorization), 2008)

“The authorization policy model applies to authorization methods based on both Role-based Access Control (RBAC) and Access Control List (ACL). One important consideration is the Identifier of HS not only includes Unique ID of user, device or other types of HS, URI or URL of HS and Address of HS, but it should contain the role of the subject in the system and in the role of subject in the other system” (ITU-T Study Group 17(Authorization), 2008).

Authorization server usually installed in the smart home gateway but it is possible to be installed in any other places, for example home devices. There are two types of authorization model in smart home network, centralized and distributed.

**Centralized Authorization:** Centralized Authorization model is a kind of legacy authorization system that a centralized authority decides all decisions about access privileges to the resources.
This model is relatively simpler and more manageable rather than distributed authorization model. AS is the single node in the home network that manage the authorization requests and responses. Usually centralized authorization systems are located in the home network gateways.

**Distributed Authorization**: in this model of authorization there is no central authority in the system. So any device can let the users to access resources in the home. This model is less manageable, maintainable and more difficult to implement. For example it is a hard task to manage distributed policy databases on different devices in the home network.

On the other hand, there are two type of authorization mode: Static authorization and Dynamic Authorization mode.

**Static Authorization mode**: in this mode one administrator is responsible to define all authorization policies. She defines all resources, users and authorization policies in the system. Role-based Access Control (RBAC) and Access Control List (ACL) are two types of static authorization systems.

**Dynamic Authorization Mode**: generally speaking, this mode is supporting authorization for temporary users in the home network. When the users are not registered in the system, this mode permits them to access the privilege of devices in the home. Different factors are considered to decide about the access privileges such as rage of access, time of access, period of validity and so on.

There are several potential threats on the authorization data that should be considered and mitigated in any smart home application. Possible threats of authorization system could be summarizes in below list:

- Eavesdropping/ Interception or disclosure of authorization data,
- Injection and modification of data,
- Unauthorized access,
- Shoulder surfing,
- Lost remote terminal and stolen remote terminal.

To protect or mitigate above threats in the smart home authorization system, there are some mandatory security requirements that should be considered in the implementation of authorization system. These requirements overlay are listed below:

- Authorization data confidentiality and integrity,
- User authentication,
- Server availability and consistency.

These security requirements must be fully satisfied by any implementation of home network system.

There are not many researches particularly in the field of authorization of smart home, but below article is a good attempt to define a new security policy language as extensible home security description language (xHDL), based on XML.

“**Security technologies based on a home gateway for making smart homes secure**” (Geon Woo Kim, 2009): This article define xHDL as an alternative to Extensible access control markup language (XACML) since they believe that XACML is a general policy definition language and
there are redundant syntax notation in this standard protocol. Furthermore for the purposes of smart home system a simplified security policy specification like xHDL is required.

2.2 Smart Metering Systems

Authentication, authorization systems are similar to smart home and the same mechanism and protocols can be used in the smart metering systems security gateway. There are some considerations that proposed to ensure about some aspects of security like confidentiality in the smart metering system network as follows (Federal Office for Information Security, 2011):

1. External entities are categorized to Consumer, Grid Operator, Supplier, Producer, Meter Operator, Gateway Operator, Meter Admin, Gateway Administrator, Gateway Developer, Profile Provider and User.
2. As a general rule all processed Meter Data that is prepared to be submitted to external entities is encrypted on a content level using PKCS#7. Further, all communication with external entities is enforced to happen via encrypted, integrity protected and mutually authenticated channels.
3. AES and TLS is default encryption algorithm in the smart metering system.

In this research, the protection profile of smart metering system produced by German’s Federal office of Information security (Federal Office for Information Security, 2011) is the main resource of study for security and privacy considerations in smart metering system. This document defines the security architecture and in particular TOE (Target of Evaluation) and its environment. This security architecture is shown in Figure 1. Below is a summary of authentication requirements in this specification:

1. User authentication before any action
2. Maintaining user information including User Identity, user status, Role membership and so on.
3. Cryptographic enabled authentication for external users in WAN
4. Handling authentication validity in a limit period of time and support transparent re-authentication.

This specification defines a number of access control profiles based on “encryption” and “pseudonymisation”. Each access control profile specifies (Federal Office for Information Security, 2011):

1. How Meter Data must be processed,
2. Which processed Meter Data must be sent in which intervals,
3. To which component or external entity,
4. Signed using which key material,
5. Encrypted using which key material,
6. Whether processed Meter Data shall be pseudonymised or not,
7. Which pseudonym shall be used to send the data?

On the other hand there are some other research on the privacy and security of smart grid that is not directly related to the topic of this research but could be helpful to be considered as well:
1. **Toshiba’s ElecPrivacy** (Georgios Kalogridis, Costas Efthymiou, Stojan Z. Denic, Tim A. Lewis and Rafael Cepeda, 2010) : This article proposes a hardware base solution as a power management system using rechargeable batteries to conceal real power consumption in the home. This solution provide an acceptable privacy protection level in the home/building with smart meters.

2. **Toshiba’s Escrow architecture** (Costas Efthymiou and Georgios Kalogridis, 2010): this research proposes a frequent anonymizing data sent to utility provider by smart meters to hide the actual power consumption.

Both researches concentrate on the privacy of customer data specifically actual power consumption as a symptom of consumer presence in the home. None of available papers focuses on data interchange between different architectures e.g. smart grid and smart home systems. There are two different threat group in the smart grid system:

1. Threat on the network communication between smart meters and power provider.
2. Threat on power provider and third systems or smart meters and third party systems.

All of the articles discuss about the first category threats and there are not many resources available on the second category threats. This research tries to fill this gap in the security and privacy of customer information in smart homes and smart grids.
This chapter is intended to investigate some of the most famous and applicable Authorization protocols like eXtensible Access Control Mark-up Language (XACML) and Enterprise Privacy Authorization Language (EPAL) in the distributed open environment. At the rest of this section each protocol will be described in detail.

3.1 eXtensible Access Control Markup Language (XACML)

XACML stands for eXtensible Access Control Markup Language. XACML is a distributed authorization standard of OASIS (Organization for the Advancement of Structured Information Standards). XACML describes a policy language and an access control decision-making system (XML-based request/response messages). XACML is a general-purpose language that lets a specific subject (e.g., human user, workstation, and so on) access a specific object (resource) in particular conditions. XACML supports both role-based access control (RBAC) and access control list (ACL) authorization systems. In this section, XACML will be introduced briefly by different aspects, e.g., data flow model, policy structure, and request/response format. XACML as an access control language has several points to be considered in distributed authorization systems as listed below:

1. First of all, it is a standard language, a large number of expert communities are agreed upon it and different organizations support this standard model.
2. Second, it is a generic language. Furthermore, XACML is not limited to a specific environment. XACML supports a variety of subjects, objects, and action types.
3. Third, it is a distributed authorization language. So policy definition, policy enforcement, policy decisions will be done in different components. Different people are responsible for different components. It supports one of the most important security design principles as separation of duties (SoD). And also it is a powerful language in order you may define any type of conditions, for example, time, date, count, and so on. It supports different data types, functions, rules, and policies. And finally, it is highly extensible, so you can define any customized conditions and policy as well.

3.1.1 XACML Data Flow Model

XACML specification defines four basic components in the XACML Architectures as defined briefly in the rest of this section:

1. **Policy Administration Point (PAP)**: PAP is the repository of the policies and provides the policies to the Policy Decision Point (PDP), in addition it provides user interfaces for policy administrators to define new policies or edit any predefined policies.
2. **Policy Enforcement Point (PEP)**: PEP is the interface of the whole environment to the outside world. It receives access requests and evaluates them with the help of the other actors and issues access permits or access denies to the requester of resources. PEP is only interface to the external part of authorization system and has no authority to define policy, making decision or any other functions in authorization system.
3. **Policy Decision Point (PDP):** PDP is the main decision point for the access requests. It collects all the necessary information from other actors and concludes a decision. Actually, PDP is the brain of the authorization system body.

4. **Policy Information Point (PIP):** PIP is the point where the necessary attributes for the policy evaluation are retrieved from several external or internal actors. The attributes can be retrieved from the resource to be accessed, environment (e.g., time), subjects, and so forth.

Data flow of the XACML architecture from Policy definition to the XACML request/ responses is shown in Figure 9.

![Figure 8. XACML Data Flow Diagram (Moses, 2005)](image)

XACML Data Model works as below steps:

1. In the PAP, policies and policy sets will be defined by policy administrator(s). These policies are available to the PDP. Any changes in the authorization policies will be defined in the PAP for any specified target.

2. Communication of PEP and context handler is in the native form, so the context handler is responsible to manage different request and response formats in 3 and 12. All requests may have different attributes of the subjects, resource, action and environment.

3. The context handler sends XACML request to the PDP.

4. The PDP may request more attributes of subject, resource, action and environment from context handler through steps 5 and 10. Context handler connects to the PIP and gets all required information via steps 6 and 8. PIP gathers all required attributes in step 7.

5. The PDP provides the response context consisting of the authorization decision to the context handler.

6. The context handler converts the response context to the native response format of the PEP. And sends it to the PEP.

7. The PEP runs all of the obligations. If the authorization is permitted, then the PEP allows requester to access the resource; otherwise, it denies access.
3.1.2 XACML Policy Structure

The main elements of the XACML policy language are Rule, Policy, and Policy Set. In this section, definition of each element will be presented briefly.

**Rule**: rule is elementary part of policy language. Each rule may consist of three parts, one or more Target(s), one Effect and one or more Condition(s). Target is specification of Resources, Subjects, Actions and Environment. The Effect could be a value of “Permit” or “Deny” and Condition is optional logical expression to refine the applicability of target.

**Policy**: policy is a combination of a Target, a set of Rules, Rule combining algorithm and Obligations. Rule combining algorithm is important part to final decision of PDP and is the way to combine the results of different policies and making final decision on the XACML request. It could be one of the values listed below:

1. **Deny-overrides**: if any of the rules in the Policy is “Deny” the final result shall be “Deny”.
2. **Ordered-deny-overrides**: this algorithm is identical with Deny-Override with an exception; the order of rule evaluation shall be the same as order of rules in the policy set.
3. **Permit-overrides**: The result of the Policy will be “Permit” if at least one of the rule results is Permit.
4. **Ordered-permit-overrides**: the behaviour of this algorithm is similar to the Permit-override with an exception; the order of rule evaluation shall be the same as rule definition in the policy.
5. **First-applicable**: the result of the policy is the same as first rule in the policy that its target matches the request and the conditions is evaluated by True value.

Obligations could be part of policies and policy sets and their values should be return to upper level of evaluation only if the evaluation result of the policy or policy set matches the “FulfillOn” attribute of the obligation.

**Policy Set**: the Policy Set is a combination of one Target, a policy combining algorithm, a set of policies and optional obligations. The evaluation result of the Policy set is based on policy combination algorithm as described above for the rule combination algorithm, the same analogical definition is for Policy combination algorithm in which policy is evaluated instead of rule. And also there are one more combination algorithm as Only-one-applicable that means the result of XACML Authorization Request is “NotApplicable” if and only more than one Policy match the Target and condition of the XACML Request, Otherwise is equal to the Policy evaluation that matches to the request both Target and Condition criteria.

XACML Policy language model is shown briefly in Figure 10:
There are some samples of XACML Policy in section 7(Implementation). Please refer to this section for more details.

3.1.3 XACML Authorization Request and Response

XACML specification isolates core language from different application environments by a new concept as XACML Context. XACML context is shown in Figure 11. XACML Request is a combination of Subject, Resource, Action and Environment Attributes in a XML format. XACML request is sent by PEP to the PDP and PDP returns XACML Responses to the PEP. Each XACML Response is consisting of three parts as Result, Status and Obligations. Attributes are in the form of XPath Expressions or identified by attribute designators. Some typical sample of XACML request and response will be presented in the section 7(Implementation) of this report. Please refer to this section for more information.

XPath: XPath or XML Path Language is a query language on the XML files. In the other words, XPath is a language to evaluate values like strings, integers, strings from XML files. Path specification was defined by the World Wide Web Consortium (W3C). For more information please refer to XPath 2.0 specification (Anders Berglund, Scott Boag, et al, 2013).
Security and privacy are not the point of concern in the XACML specification but it has some informative suggestions for whom implementing any XACML based application. Not only all messages transmitted between PDP and PEP are susceptible to attack by malicious attackers but all other components indirectly related to XACML environment including operating systems and domain name systems (DNSs) could be compromised by attackers but the latter case is not in the scope of security and privacy for the XACML based applications and are general security and privacy concerns that should be considered properly by any security architect.

General threats over any message transmission system and safeguarding mechanisms against them are as listed below. These threats must be considered in any implementation of XACML system as well:

1. **Unauthorized disclosure**: disclosure of Subject attribute, decision request and response is a point of concern that should be safeguarded by confidentiality support features.
2. **Message replay**: adversary can record the legitimate messages and replay back to the components of the XACML architecture. Messages should be protected against replay attacks scenarios.
3. **Message insertion**: attackers can insert some messages in the scenarios of XACML, so it is necessary to have message sequence integrity and mutual authentication. It should be mentioned SSL mutual authentication is not enough and authentication should be established before message passing.
4. **Message deletion**: this threat will be realized when an attacker delete some messages in the sequence of XACM use-stories and could be a point of Denial of service attack. All components have to ignore all incorrect decision requests. To safeguard this threat message sequence integrity mechanisms are applicable.
5. **Message modification**: Message integrity mechanisms are the defence mechanism to this kind of attacks.

One more specific threat on the XACML is “Not Applicable results”, not applicable results means PDP could not find any related policy in which its target match’s the decision request and condition is consistent. It is so recommended in this case the Deny result will be returns instead of not applicable. For any reason like incorrect request format, Network issues it is possible to have some requests that are not controlled by decision policies so the PDP returns Not applicable, for the sake of security considerations it is highly recommended to return Deny
results in this cases so PEP grant access to the request only and only of PDP explicitly return Permit results.

### 3.1.5 XACML Security and Policy Safeguards

1. **Authentication**: Before doing any authorization transaction, it is required to authenticate all parties. Authentication could be one direction or bilateral authentication. So authentication is prerequisite to any authorization decision request/response between PDP and PEP.
2. **Confidentiality**: this countermeasure mechanism ensures that only desired participants are able to read data. Confidentiality is for both Policy information and message in transit.
3. **Policy Integrity**: Digital signatures are a well-designed mechanism to ensure about data integrity. Integrity is important not only for authorization response but for Policy stored in the PDP. Therefore it should be guaranteed that policies are not changed after definition in the PAP anytime authorization decision making phase is demanded.
4. **Trust Model**: PDP collects data from PEP like authentication data and verifies them, so the correctness of data is not in charge of PDP. Also the correct results of the total system are dependent to the PEP’s policy decisions enforcement. And finally the PDP results are based on the data passed by PEP, so PEP is responsible to provide correct data to PDP.
5. **Privacy**: any Authorization request/response message may reveal private information of subjects, so any privacy preserving mechanisms like encryption, authentication, integrity checking and verification shall employ to protect all subject attributes in the environment.

### 3.1.6 XACML Pros and Cons

XACML has many benefits over other access control policy languages (Sun Microsystems, 2006):

1. It is a generic access control policy language, so easily can replace any other of application-specific access control policy languages.
2. XACML saves time and money of administrators because they don't need to rewrite their access control policies in many different application domain languages.
3. XACML saves time and money of developers, since they don't require developing any other new policy languages and writing code to support them. They can reuse existing code.
4. There is a variety of good tools for writing and managing XACML policies. So they can be used with many applications.
5. XACML is flexible and extensible enough to support most access control policy requirements in different application domains.
6. One XACML policy can cover many resources. This helps avoid inconsistent policies on different resources.
7. XACML support inter-policy relationship. It permits one policy to refer to another policy. This is important for large organizations. For instance, a site-specific policy may refer to a company-wide policy and a country-specific policy.
3.2 Enterprise Privacy Authorization Language (EPAL)

Enterprise Privacy Authorization Language (EPAL) is a standard language for enterprise access control systems or privacy policies to manage data access in the enterprise organizations based on fine-grained positive and negative authorization rights.

IBM is the creator and most supporter of this access control policy language. IBM submitted EPAL 1.2 to World Wide Web Consortium (W3C) to be considered for recommendation on November 2003. EPAL is a request/response message authorization system. The rest of this section is describing Policy definition, Request and response format of EPAL messages and finally a summary of advantages of EPAL as an Access Control policy language or authorization language.

3.2.1 EPAL Policy Structure

Before focusing on the EPAL’s policy, it is required to define two related terms of EPAL as Vocabulary and Container.

**Vocabulary:** Vocabulary defines the language to specify domain-specific privacy policy. Every vocabulary may contain seven different parts including one vocabulary info, zero or more of user-category, data-category, purpose, action, container and obligation.

**Vocabulary info:** is some information about the privacy policy like issuer, expiration time and date, version number and so on.

**Container:** actually the container is a set of attributes values. Attributes may be not ordered and multiple values. Containers are used to evaluate the conditions, all of containers belong to a condition must be available in evaluation time.

On the Other hand, EPAL specification defines policy rule (Authorization rule) as a combination of six hierarchical elements of data-categories, user-categories, purposes, sets of actions, obligations, and conditions. Any rule specifies certain user-categories are allowed or denied to access a certain data-categories to perform some actions with specified purposes if the conditions are satisfied and finally what obligations that environment should execute after performing the actions. A sample rule file is shown in Figure 11.
Below is a short description of each element (Ashley, Hada, Karjoth, Powers, & Schunter, 2003):

1. **User-category**: user-category is the entity either users or groups that use collected data (e.g., travel expense department or tax auditor).
2. **Data-category**: data-category defines different types of collected data that are managed differently from a privacy perspective (e.g., medical-record vs. contact-data).
3. **Purpose**: the purpose models the intended service for which data is used (e.g., processing a travel expense reimbursement or auditing purposes).
4. **Action**: action specifies how the data is used (e.g., disclose vs. read).
5. **Obligation**: obligation defines actions that must be taken by the environment of EPAL when a specific action performs (e.g., delete after 30 days or get consent).
6. **Condition**: is a Boolean expression that evaluates the context (e.g., “the user-category must be an adult” or “the user-category must be the primary care physician of the data-subject”).

A Unified Modeling Language (UML) model of EPAL Policy structure is shown in Figure 12:
3.2.2 EPAL Authorization Request and Response

EPAL is a request and response authorization system, so any requester should submit an EPAL request to the authorization server and authorization server returns an EPAL authorization response message. Any EPAL Request message includes a user-category (U), a data-category (T), a purpose (P) and an action (A). These four items make the tuple (U, T, P, A) called as authorization quadruple. A sample of EPAL authorization request is shown in Figure 13:

```xml
<xml version="1.0" encoding="UTF-8">
<epal-query
 xmlns="http://www.research.ibm.com/privacy/epal/interface"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
<user-category refid="SalesDepartment"/>
<data-category refid="CustomerRecord"/>
<purpose refid="OrderProcessing"/>
<action refid="Store"/>
<container refid="CustomerRecord">
<attribute refid="CustomerID">
  <value>0123456789</value>
</attribute>
</container>
</epal-query>
```

Figure 13. EPAL Authorization Request Message

The EPAL authorization response message is a set of three parts: a result e.g. "allow", "deny", or "not-applicable", a rule id that mandates the result and a set of obligations. An example of EPAL response message is shown in Figure 14:
3.2.3 EPAL Pros and Cons

EPAL is a general purposes authorization protocol but it is not a good option for some situations listed in the following list:

1. “Directly encoding or enforcing specific privacy legislation in a generic and completely application and enterprise independent way.
2. Manipulation of EPAL by data subjects to set their preferences: E.g., providing information that enables the creation of user-preferences forms based on the information given in an EPAL policy.
3. Creation of new mechanisms or syntax for data representation: EPAL should abstract from the data model and should not provide its own syntax for describing schemas for data.
4. The limitation of features in EPAL for the sake of conformance to the assumptions, data structures, or features of any particular pre-existing product or tool” (Ashley, Hada, Karjoth, Powers, & Schunter, 2003).
5. In addition, there is not any open source implementation of EPAL for any platform or programming languages.

3.3 Discussion about Federated Authorization Systems

While EPAL and XACML are very analogous in different aspects, Both of them are XML-based request/response model of authorization, but actually are different in functionality, capability and applicability. Generally speaking, the functionality of XACML is a superset of EPAL. Furthermore, the EPAL has less functionality than XACM. In many cases, the EPAL is not flexible or scalable enough to construct privacy policies for a certain domain. Even though in some cases it sounds impossible or very difficult, There are major features contained in XACML that are not in EPAL (Anderson, 2006):

1. Combination of multiple policies results issued by different independent issuers.
2. Inter-policy reference in the XACML, and making complicated conditions and principles upon an authorization request.
3. Multiple authorization decisions for each node when authorization request is on a hierarchical resource such as an XML document.
4. Conditions with multiple principals, independent hierarchical roles or groups.
5. Handling of error conditions and missing policy attributes based on policy definition.
7. Support for additional primitive data types including X.500 Distinguished Names, RFC822 names, and IP addresses.
8. XCACML is OASIS Standard supported by different organizations, mature access control language.
9. Several open source XACML implementation is available. On the other hand there is not any open source implementation for EPAL.

The only significant functions added by EPAL that are not in XACML are (Anderson, 2006):

1. The concept of a policy vocabulary. Although, XACML supports all of the major functionalities of EPAL but in complex authorization policies, policy vocabulary facilitates the policy definition.
2. Hierarchical categories of user and data. Although it is not a big advantage of a policy definition language and XACML is also able to support these through its context handler component with no change in the syntax of the language.

Considering these differences, we could conclude that it is more appropriate for future research and standards efforts related to privacy policy languages to focus on XACML. Enterprises and organizations requiring a privacy policy language should establish XACML systems. However XACML is a large, generic and complex language, so it is hard to have a formal analysis. But it is recommended to use a subset of XACML features for the certain domain in the organization.
4 FEDERATED AUTHENTICATION SYSTEMS

There are different standards and mechanisms for authentication in the decentralized, distributed and inter-domain environments. Most important and well-known standards and mechanisms are Security Assertion Markup Language (SAML), OpenID and Open Authorization (OAuth). In the rest of this section, these protocols will be described and discussed in different aspects like security, privacy and also their pros and cons.

4.1 Security Assertion Markup Language (SAML)

Security Assertion Markup Language (SAML) is an open standard, XML-based framework to exchange security information e.g. Authentication, Authorization and Attributes between online business domains. OASIS announced the SAML v1.0 specification in November 2002 and the second edition in November 2003. Security information expressed in SAML Assertions has been defined in precise specification including data structure, communication protocols, and request and response operations. The major requirements driven to the SAML specifications are:

1. Single Sign on (SSO)
2. Federated Identity
3. Web Services standards.

**Single Sign on:** Single Sign on is sharing authentication data between different domains. Having same identity provider or SAML Asserting Party (AP) and several service provider or SAML Relying Party (RP), each service provider uses the identity information provided by identity provider, so any user authenticates itself to identity provider and identity provider sends all required identity attributes to service providers.

**Federated Identity:** is collaborative authentication between different systems in which the same identity or user has different local authentication information in each system but these systems are able to agree, share data based on a common identity name.

**Web Services:** SAML specification allows non-native SAML context to carry the SAML assertions in their frameworks, WS-Security is one of the most applications of assertion token in Web service areas.

One of the most common scenarios is using SAML authentication between different shopping sites. When you are authenticated in a web site for example an Airline web site then you try to access to other link in the site e.g. to rent a car in other city or booking a room in a hotel. If these sites don't have any agreement in their business layer to support each other it is cumbersome for customers to authenticate themselves for all of the web sites. SAML SSO or Federated Identity use cases solve this issue and provide a comfortable shopping for customers. A general model of SAML Web SSO and SAML Federated Identity servers are shown in Figures 15, 16:
SAML specification consists of numerous requirement documents. In this report some of most important documents will be presented like Assertions and Protocols, Binding, Profiles and Security Privacy Considerations documents. The relationship of different components of SAML standard is shown in Figure 17, at the rest of this section these document will be introduced from bottom to top, i.e. Assertions, Protocols, Binding and Profiles:

**Assertions**: assertions are request and response messages in the form of statements of subjects. Assertions contains required and optional fields. SAML Specification defines three kinds of Statements, e.g. Authentication, Attribute and Authorization Decision statements. Authentication statements will be considered in this report in more details.

**Protocols**: Protocols are some contract consisting of requests and responses. According to SAML specification, there are six category of protocols including:

1. Authentication Request Protocol
2. Single Logout Protocol
3. Assertion Query and Request Protocol
4. Artefact Resolution Protocol
5. Name Identifier Management Protocol
6. Name Identifier Mapping Protocol

Two first protocols is required in the implementation of Web SSO scenario, so it will be discussed in more details in this report.

**Authentication Request Protocol:** This protocol defines the process of security context establishment to exchange authentication info between Identity Provider (IdP) and Service Provider (SP).

**Single Logout Protocol:** this protocol defines a mechanism to sign out from different sessions (service providers) associated with a principle simultaneously.

**Binding:** Binding is the next document of SAML specification documents that are considered as a major part of this protocol. Binding is a protocol to define how the SAML protocol could be carried over the underlying transport protocols.

SAML Binding protocol (Scott Cantor, Frederick Hirsch, et al, 2005) specifies six Binding Protocols including:

1. HTTP Redirect Binding
2. HTTP POST Binding
3. HTTP Artefact Binding
4. SAML SOAP Binding
5. Reverse SOAP Binding
6. SAML URI Binding

“HTTP Redirect binding” is one of the most applied binding protocol in the real use cases and it will be used in the implementation of this research project. This protocol will be described with more details in the following sections.

**SAML Profiles:** profiles define how the SAML Assertions, Protocols and Binding are combined together for a particular use case. SAMLV2.0 specifications (John Hughes, Scott Cantor, et al, 2005) defines eight profiles as follows:

1. Web Browser SSO Profile
2. Enhanced Client and Proxy (ECP) Profile
3. Identity Provider Discovery Profile
4. Single Logout Profile
5. Assertion Query/Request Profile
6. Artefact Resolution Profile
7. Name Identifier Management Profile
8. Name Identifier Mapping Profile

In the implementation of this project, the first and most important Profile i.e. Web Browser SSO Profile is used. In the rest of this section we discuss about further details of SAML components briefly, SAML Profiles, Binding in addition to SAML Security and Privacy considerations.

**4.1.1 SAML Profiles**

As previously mentioned, there are eight different SAML profiles described in SAML specification V2.0, however I am using only SAML Web Browser SSO Profile in the implementation of this project. Furthermore, this particular profile will be elaborated in detail. SAML support two options for Web browser SSO, Sp-Initialted and IdP-Initiated. The most
The common option is Sp-Initiated when the user is not logged in to the service provider, so when she requests a new service the SP sends her to the IdP to be authenticated. Finally the IdP makes a new authentication assertion and sends user information back to the SP. SP processes the assertion and accepts or denies the user’s request. Second option occurs when the user is already logged to IdP and click to a SP link. In this case the IdP starts an IdP-Initiated SSO process. Two SAML SSO Options is shown in Figure 15:

![Figure 15: Two SAML SSO Options](image)

For further information about other SAML profiles please refer to the SAML specification V2.0 documentation (N.Ragouzis & al, 2008).

### 4.1.2 SAML Bindings

SAML specification defines three binding models for authentication Request in Web browser SSO profile when SP initiates the process:

1. HTTP Redirect Binding
2. HTTP POST Binding
3. HTTP Artefact Binding

The Authentication response from IdP to SP will be available either through HTTP POST Binding or HTTP Artefact Binding. In the Implementation of this project, SP-initiated SSO process uses a Redirect Binding for the SP-to-IdP <AuthnRequest> message and a POST binding for the IdP-to-SP <Response> message. Post Binding model of collaboration between IdP and SP is shown in Figure 19. According to this model, the authentication process will be established upon seven steps (messages).

1. Authentication process starts by user request in the browser.
2. SP redirects user request to IdP via HTTP Redirect Binding message.
3. IdP exchanges the authentication data (user name and password) via browser with user through step 3 and 4.
4. IdP sends back signed user attributes (authentication response) to the browser.
5. Browser posts signed authentication response to the SP.
6. SP decides to accept or deny the user authentication request and finally continues or stops process.
4.1.3 Security in SAML

When the Relying Party (SP) and Asserting Party (IdP) are working together, several security threats could be possible to occur like replay attack, Man-in-the-middle attack. SAML Specification defines some security and privacy considerations in addition to some solutions to take over these threats. SAML specification defines a variety of security mechanisms to secure the SAML operations. These solutions are as follows (N.Ragouzis & al, 2008):

1. Establishment of a Public Key infrastructure (PKI) to make a trust relationship between Asserting Party and Relying party.
2. Using SSL3.0 or TLS 1.0 when the message integrity and message confidentiality is required.
3. Applying XML signature to digitally sign the authentication responses to be assured of authentication response integrity.

4.1.4 Privacy in SAML

SAML specification V2.0 determines a certain definition for privacy, privacy is user’s ability to control how the Identity data is shared or used. However, the privacy is not only the confidentiality of information but availability of information could be a breach of privacy in some use cases. For example turn on or turn off an appliance in the smart metering system or secure home area could be a sign of human presence in the house. SAML specification recommends some mechanisms to support the user’s privacy in the regular scenarios that are described in the following list (N.Ragouzis & al, 2008):

1. SAML supports pseudonyms negotiation between an identity provider and a service provider. Pseudonyms may protect the identities among the communication of different prevents service providers.
2. SAML supports one-time or transient identifiers. Although, service providers have ability to single sign on but no service providers is able to identify identities that have been visited their services recently.
3. SAML supports different assurance level, so based on the resource type it could be defined which service provider is allowed to access that resource.
4. SAML supports user confirmation on claims while certain operations are established between providers. The process of user consent is not specified in the SAML specification, and it is out of the scope of this specification.

### 4.1.5 SAML Pros and Cons

SAML has several advantages that could be summarized as follows:

1. SAML is Open standard, XML-based protocol. So, SAML is very extensible and flexible authentication standard.
2. The same software module could be implemented for both Identity Provider (IdP) and Service Provider (SP). Nonstandard and proprietary SSO mechanism requires each new provider new certain software implementation. However, a single SAML implementation can support SSO connections with many different federation partners.
3. Although SAML has a broad scope, but it has robust security and privacy model.
4. SAML enhances security, mitigates security risks and improves compliance by eliminating Web application passwords.
5. SAML reduces costs and increases productivity for the entire enterprise.

On the other hand there are some disadvantages or lack of functionalities for SAML that could be summarized as listed below:

1. SAML is a heavy protocol. It has a complex set of specifications in addition supports complicated constraints on security rules. The complexity of security rules is a result of SAML’s development for sophisticated business scenarios where SAML has consequently experienced significant success.
2. SAML was never optimized for SSO support in mobile applications. On the other hand, OAuth 2.0 as a competitor to SAML has emerged to satisfy all mobile application requirements, so it provides more value and use case flexibility.

### 4.2 OpenID

OpenID is an open standard that describes how users can be authenticated in a decentralized environment. It provides a way to prove user identity. OPENID 2.0 specification defines two components for its authentication, Relying Party (Service Provider) and Asserting Party (Identity Provider). OpenID uses general HTTP(S) requests and responses. So it is not dependent to any special capabilities for User agent of client software. OpenID is not relying to Client session managements like cookies. OpenID is a framework to enable portable, user-centric digital identity in decentralized environments. OpenID Foundation (OPIF) is most important community to support, promote and develop new features of the OpenID standard. Several companies incorporate in the OpenID foundation including Google, Microsoft, Yahoo, Facebook, and Symantec and so on.
4.2.1 OpenID Protocol Overview

OpenID Authentication 2.0 specification defines the process of communication between Relying Party and OpenID Provider or asserting Party as it is shown in Figure 20. The process of communication or protocol workflow is as follows:

1. OpenID authentication use case starts by user request to login the server the end user’s User-Agent is responsible to provide and OpenID Provider Identifier. This OP Identifier could be an URL either HTTP or HTTPS or XRI format.
2. Normalization is the process of rendering the user request URL and changes it to canonical form. Relying party is responsible not only for normalizing but for discovery the OpenID providers based on normalized user-entered OpenID Identifier.
3. Next step is establishing an association between Relying Party and OpenID Provider (OP). Association is a shared secret e.g. Diffie-Hellman Key Exchange between RP and OP, this association is a preventive mechanism for subsequent messages round trips. It is optional but it is highly recommended in the OpenID specification.
4. In the next Step RP redirects the user-agent to OP with an OpenID Authentication request.
5. OP verifies the user authentication information and returns results to RP.
6. RP verifies the results back from OP either authentication approved or failed. RP should checks some other information like nonce and signature that is generated during the association establishment.

Figure 20. OpenID 2.0 Protocol Flow

4.2.2 OpenID Security Considerations

OpenID Specification enumerates most important attacks over OpenID authentication protocol as listed in the following list, in addition it introduces some safeguarding or preventive mechanism for any of them (Barry Ferg, Brad Fitzpatrick, et al, 2007):
1. **Eavesdropping Attacks**: transport layer encryption e.g. TLS and nonce are used to prevent this kind of attacks.

2. **Man-in-the-Middle Attacks**: association is a countermeasure to this type of attacks. It is not possible to change data without changing the MAC that is established in the Association phase. Association is vulnerable if Transport Layer encryption or signatures will be compromised. Using SSL/TLS with certificate signed by certified authority prevent this kind of vulnerabilities. It is strongly recommended to use SSL/TLS in all communication path.

3. **Rogue Relying Party Proxying**: This is a special kind of MITM attack where the User-Agent is a rouge party. So in the authentication phase, rough RP captures the user credential like user name and password. There are several ways to prevent this attack i.e. Creating secure channel with End User.

4. **User-Agents**: User-Agents or common web browsers like Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, and so on are the target of spywares and malwares. There are two point of vulnerability in the User-agent side, first in the RP, second in the OP.
   - In the RP, it is very much possible that user’s credential will be disclosed to malwares. On the other hand cross site scripting (CSS) is another way to attack the OP. It is highly recommended to not rely on scripting in the OP.
   - Since the RP redirects users to OP sites, phishing attacks are another potential vulnerability. User education is most powerful defensive mechanism against Phishing attacks.

5. **Denial of services attacks (DoS)**: Rouge RP may initiate DoS attacks to OP by requesting frequently association, authentication and verification of signatures. The effective way of protection is using generic IP base rate-limiting and banning techniques.

### 4.2.3 OpenID Pros and Cons

Most important advantages of the OpenID protocol are:

- The OpenID 2.0 base protocol is very simple, but can be improved by additional extensions to pass registration attributes or to classify the strength of authentication method used at the Identity Provider.
- OpenID does not require any pre-registration in the Asserting Party (AP). Pre-registration of Relying party is mandatory in SAML. Also the RP is able to dynamically ask some user attributes in the authentication request but in the SAML it is required to register the attribute profiles in the SAML server.

On the other hand, OpenID emphasizes on SSO rather than attribute sharing. That is a greatest weakness or limitation of OpenID standard.

### 4.3 Open Authorization (OAuth)

OAuth is an open standard for Authorization that enables third party applications to access the resources on the resource server on behalf of the resource owner over HTTP service. Traditional client-server architecture allows users to enter their credential like username and password to access their resources in the resource server. So any third party application desires to access the resources in the resource server as a representative of the clients should be aware of the client credentials. Having access to the user credentials as a third party application is the source of
some security and privacy problems. Most important security issues are as listed in the following:

1. Third party applications are responsible to keep the credential information for future uses.
2. Password as a weak authentication mechanism should be supported by servers.
3. Third party applications are able to access unlimited to protected resources. There is no way to limit the duration time or scope of access for the third party application.
4. There is no way to revoke the access grant to a specific third party application except change the password or credential that affects all of third party applications.
5. Any compromised third party is a compromise to the resource server and all shared protected information.

OAuth introduces a new way of access method to the resources in the servers. Instead of using resource owner’s credential the client uses access token and present to resource server to access the protected resources. This standard is based on bearer tokens over HTTP/1.1 and TLS in the transport layer.

### 4.3.1 OAuth Protocol Overview

OAuth 2.0 specification defines four roles or elements in its authorization process. These roles are: Client, Resource Owner, Resource Server, and Authorization Server. The data flow model of OAuth is shown in Figure 21:

![Figure 21. OAuth 2.0 Authentication Protocol Flow (D. Hardt, 2012)](image)

There are three steps in OAuth Authentication model that will be described briefly in the following paragraph:

1. In the first step, client requests a resource from resource owner (A and B). At this step the resource Owner redirects the user agent or client to authorization server. Resource owner provides client with an authorization grant to deliver the authorization server.
2. At the second step, client delivers authorization grant to authorization server and authorization server provide an access token to be delivered to resource server. The client is intended to carry on the access token and delivers it to the resource server (C and D).
3. In the third step Client delivers the access token to the resource owner and the resource owner returns the protected resource to the client if the access token will be verified properly in the resource owner site (E and F).

Authorization grant is a resource owner’s credential in one of four defined types:

1. Authorization code: is consists of Client Id and Authorize URL.
2. Implicit: Contains Client Id and Authorize URL.
3. Resource owner password credentials: is a combination of Client Id, Client Secret and Access Token URL, User name and Password.
4. Client credentials: is asset of Client Id, Client secret and Access Token URL.

Table summarizes the required fields of information in authorization grant phase:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Authorization Code</th>
<th>Implicit</th>
<th>Client Credentials</th>
<th>Resource Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Id</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Client Secret</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Access Token URL</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Authorize URL</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Name</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Password</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

As simple definition Authorization code and Implicit Authorization Grant are two steps model when the authorization grant issued to client (Authentication phase) and then client provides authorization with authorization code and get the access token in return (Authorization phase).

Resource owner is single step request to resource owner when the client relies on the resource owner and provides all of the required information for access token.

The client credential is used when the scope of the authorization is limited to the resources under client control.

All of the authorization grant requests may have a scope filed containing space delimited strings like “openid email” or “openid profile” in OpenID Connect (OpenID Connect Basic Client Profile, 2013). OpenID connect is a simple identity layer over OAuth to simulate the OpenID. OpenID Connect will provide subject claim information similar to OpenID over OAuth Protocol. For example when the scope is “openid profile” the identity servers return some subject attributes (claims) such as: name, family name, given name, middle name, nickname, preferred username, profile, picture, website, gender, birthdate, zone info and locale.

4.3.2 OAuth vs. OpenID

Although the process of authentication in OAuth and OpenID is almost same but there are a big difference between these two protocols. OpenID identity provider issues the identity claims but OAuth issues the valet key or OAuth Token Key. OpenID Connect is an extension to OAuth to support Subject’s attributes or claims in the response message and is a compensation of OAuth in
addition to simplicity, flexibility and applicability of OAuth. The difference of OpenID and OAuth is shown in Figure 22.

![OpenID Authentication vs OAuth Authentication](image)

Figure 22. OpenID vs. OAuth Authentication (Wikipedia, OAuth, 2013)

### 4.3.3 OAuth Pros and Cons

Most important characteristic or advantage of OAuth 2.0 is supporting a variety of different client types, which access REST APIs either applications running on web servers or cloud environment or applications running on employee or customer mobile devices. This particular characteristic of OAuth is based on multiple mechanisms defined in the specification for “getting a token” and also different mechanisms to acknowledge the client type constraints.

On the contrary, OAuth is not supporting attribute exchange or claim based authentication, so the service provider is not able to get the attributes of subject from Identity provider. But this shortcoming is completely compensated by a new extension to OAuth as OpenID Connect (OpenID Connect Basic Client Profile, 2013).

### 4.4 Discussion about Federated Authentication Systems

Authentication Systems or Identity Provider Systems typically focus on the exchange of certain types of identity attributes or claims. These attributes are either unique user identifiers or more general attributes (such as email address, telephone number, user profile information, etc.).

Table 5 is a summary of all protocols containing User identifier format and the attribute exchange capability of them.
Table 5. Identity Types and Their Attributes

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Identifiers</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML</td>
<td>A variety of identifier types, from pseudonyms to one-time anonymous.</td>
<td>General XML-based syntax for Identity Attributes (Claims) exchange</td>
</tr>
<tr>
<td>OpenID</td>
<td>Persistent user identifier</td>
<td>Access token is a key to retrieve user attributes(Claims)</td>
</tr>
<tr>
<td>OAuth</td>
<td>No identifier</td>
<td>OAuth token can be used only for access to resources or services. Originally, not supporting attribute exchange but OpenID connect as an extension support attribute exchange as well. Since the access token is in the HTTP header, so attribute exchange is possible in the extended protocol.</td>
</tr>
</tbody>
</table>

A summary of discussions on federated identify protocols are as follows:

1. SAML supports all kind of user identities like actual identity, pseudonyms and one-time anonymous identity but OAuth and OpenID support only actual user identity.
2. Generally OAuth is not supposed as an authentication protocol and the main purpose of OAuth is authorization, so attribute exchange is not designed in the basic OAuth specification but OpenID connect as an extension to the OAuth supports Attribute exchange too.
3. OpenID does not required any Identity Provider pre-registration, but SAML and OAuth mandates service providers to pre-registers themselves in the Identity provider and get some secret information like client id , client secret in OAuth or Issuer Id and consumer Index in SAML.
4. SAML supports both IdP and SP initiated transactions. However OAuth and Open ID supports only SP initiated transactions.
5. OpenID does not define security requirements properly and is appropriate to low-sensitivity applications. But SAML and OAuth define a good description of security and privacy requirements.
6. OpenID Connect improves the security requirements of OAuth.
7. SAML defines a more robust security model which is necessary for high-sensitivity use cases. SAML is the mostly known choice for federation between business partners, enterprise and Cloud applications.
In this chapter all of the requirements of this research will be specified precisely. First of all, different categories of devices, then all use cases or scenarios and finally different actors participating in these scenarios will be discussed in details.

5.1 Appliance/Device Categories

In the description of research area, three types of appliances (home devices) were discussed. Based on this categorization, three types of home device access use-cases will be recognizable. To have a good understanding of these scenarios (use-cases), it is necessary to have a definition about devices and their types in a typical secure home or smart metering system.

In this thesis project, a de-facto device categorization standard is used to define the home appliances. This categorization is based on three attributes for any type of appliances. These attributes are as described in the following section:

1. **Type**: Devices are categorized in two high level categories ‘Major’ and ‘Minor’. ‘Major’ appliance or “white goods or white-ware are large machine which accomplishes some routine housekeeping task” (Major appliance, 2012). ‘Minor’ appliances or “small appliance or brown goods are portable or semi portable machines generally used table-top or counter-top” (Small appliance, 2012).

2. **Division**: This attribute define general functionality of the device. This attribute may have some values including ‘Refrigeration’, ‘Stoves’ and ‘Washing’ and so on.

3. **Category**: This attribute defines peculiar functionality of the device. For example the ‘Refrigeration’ division could be specialized in ‘Freezer’ or ‘Refrigerator’.

As a summary of device definition,

- ‘Major’ type is divided into four divisions i.e. ‘Refrigeration’, ‘Stoves’, ‘Washings’ and ‘Miscellaneous’.
- Every division has a verity of categories as well.

The full device or home appliance categorization model is shown in Table 6:

<table>
<thead>
<tr>
<th>Type</th>
<th>Division</th>
<th>Category</th>
<th>Type</th>
<th>Division</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major</td>
<td>Refrigeration</td>
<td>Freezer</td>
<td>Minor</td>
<td>Cooking</td>
<td>Hot Plate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Refrigerator</td>
<td></td>
<td></td>
<td>Slow Cooker</td>
</tr>
<tr>
<td></td>
<td>Stoves</td>
<td>Cooker</td>
<td></td>
<td></td>
<td>Microwave Oven</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Microwave Oven</td>
<td></td>
<td></td>
<td>Rice Cooker</td>
</tr>
<tr>
<td></td>
<td>Washing</td>
<td>Washing Machine</td>
<td></td>
<td></td>
<td>Bread Machine</td>
</tr>
</tbody>
</table>
These attributes are used in the authorization phase of the secure home and smart metering systems. Home devices or appliance are the resources or target of access operations for different subjects or users.

5.2 Use cases or Functional Requirements

In this section use cases or scenarios in the scope of the secure home and smart metering systems will be defined. Use cases are the model of interaction between actors of a system and the appropriate functions of the system to satisfy actor’s goals or requirements. Use case is a general tool to capture functional requirements of the system in software engineering world.

5.2.1 Use Case/Scenario #1: Smart Home Zone

In the first scenario a remote user attempts to access a home service via the SHG. In this scenario, the end user (actor) tries to access a home device in the green zone (Figure 3) or the area in which the Secure Home Gateway (SHG) is the only authority to grant/deny access to home devices. Actors who are interested to interact to the home devices and get services (Turn on/Turn off device and so on) of the SHG or SMG could be distinguished in different roles or access levels like ‘Tenant’, ‘Landlord’, ‘Child’ and ‘Parent’. A brief description of this use case is as follows:

1. This use case starts with user’s request to access a device in the home.
2. The first step is user authentication to system. SHG asks user information and users provide his/her information. Security channel establishment will be done to finalize this step.
3. The second step is authorization. SHG (PEP) establishes an authorization process with PDP and finally decides to give or deny user the access privilege to desired home device.
4. Final step is the presentation of results to the user either desired service or warning message because of denying user request.

Abstract model of distributed authorization is used for modelling this scenario, so Authorization system consists of two components, Policy Decision Point (PDP) and Policy Enforcement Point (PEP). PDP is the centre of authorization decision making and PEP is the point of contact with external requesters and manage network communication, security and message handling. The abstract model of SHG is shown in Figure 23.

5.2.2 User Case/Scenario #2: Smart Metering Zone

In the second scenario a remote user attempts to access a home service via the SMSG. In this scenario End user tries to access a device in the blue zone (Figure 3) or the area in which the Smart Metering System Gateway is the only authority to grant/deny access to devices. Limited number of people connects to Smart Metering System. These users could be categorized in different roles like SMS Manager, SMS Controller and so on. The sequence diagram of this scenario is very similar to the first scenario. The abstract model of Smart Metering zone is shown in Figure 24. A short description of this use case is:

1. This use case starts with user’s request to access a device in the home.
2. The first step is user authentication to system. SMSG asks user information and users provide his/her information. This step will be finished by security channel establishment.
3. The second step is authorization. SMSG (PEP) establishes an authorization process with PDP and finally decides to give or deny user the access privilege to desired home device.
4. Final step is the presentation of results to the user either desired service or warning message because of denying user request.

![Figure 23. Abstract Model of SHG](image_url)
5.2.3 User Case/Scenario #3: Common Zone

Third scenario is most complicated scenario in this system. When a remote user attempts to access a home service that is in common controlled zone between SMSG and SHG. In this scenario, the user tries to access a device in the yellow zone (Figure 3) or the area in which the Smart Metering System Gateway shares supervision of the zone with Secure Home Gateway. In this zone grant/deny access to devices shall be authorized by both security gateways. A few people connect to this zone to access devices. These users could be some people like ‘Landlord’, ‘Tenant’, ‘Smart Metering System Service’, and ‘Smart Appliance Support Service’. Having two different security gateways i.e. Secure Home Gateway (SHG) and Smart Metering System Gateway (SMSG) requires to be managed carefully. The communication protocol between two security gateways is mandatory to have a consistent secure and safe architecture.

Every security gateway has its own security policy and it is possible having collisions when a specific request rises in the system. Having stable and user-friendly home services to all end users is depended to a well-defined collaboration of two security gateways. The general security model of collaboration between all entities in the smart home and smart metering system is shown in Figure 25. The alternative scenario occurs when the remote user requests to access a device from SMSG, so this gateway should connect to SHG as well. Below is overall steps of this scenario (Use case):

1. This use case starts with user’s request to access a device in the home.
2. The first step is user authentication to SHG. SHG asks user information and users provides his/her information. Security channel establishment is the final phase of this step.
3. The second step is authorization. SHG (PEP) establishes an authorization process with PDP and finally decides to give or deny user the access privilege to desired home device.
4. At this scenario, the role of SHG is more complicated. SHG must make a communication to SMSG and not only authenticate user but get SMSG’s authorization decision on user request. So the first step of inter communication process is user authentication to SHSG and establishment of a security tunnel to SHSG. Second step is authorization process and get access control privilege of the service request. The final step is aggregation of authorization responses of two PDP i.e. SHG and SMSG.

5. Final step is the presentation of results to the user either desired service or warning message because of denying user request.

Figure 25. General Model of Collaboration of SHG and SMSG

The major issue in this scenario is how authentication and authorization data would be exchanged between two gateways particularly between PEPs in authorization phase, and also between authentication servers in authentication phase. As it mentioned in previous sections, there are several standards to overcome this problem both authentication and authorization. SAML, OAuth and OpenID are supporting federation of user identities and SSO. XACML and EPAL support distributed authorization system. These protocols handle secure approach to exchange information data in authorization and authentication of these systems.
5.3 **Actors or External Entities**

Actors are humans or IT entities who triggers any use case of the system. Different actors (Federal Office for Information Security, 2011) in the SMSG are shown in table 7:

<table>
<thead>
<tr>
<th>Actor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>Owns smart meter data e.g. home owner, tenants and so on</td>
</tr>
<tr>
<td>Grid Operator</td>
<td>Operates grids of electricity</td>
</tr>
<tr>
<td>Supplier</td>
<td>Supplies the electricity</td>
</tr>
<tr>
<td>Producer</td>
<td>Produces the electricity</td>
</tr>
<tr>
<td>Meter Operator</td>
<td>Installs and maintains the smart meters</td>
</tr>
<tr>
<td>Gateway Operator</td>
<td>Install and maintains gateways</td>
</tr>
<tr>
<td>Meter Admin</td>
<td>Admins meters as agent of the meter operator</td>
</tr>
<tr>
<td>Gateway Administrator</td>
<td>Admins gateways as agent of the gateway operator</td>
</tr>
<tr>
<td>Gateway Developer</td>
<td>Develops the gateway and also firmware updates</td>
</tr>
<tr>
<td>Profile Provider</td>
<td>Provides profiles used in information flow control</td>
</tr>
<tr>
<td>External entity/ User</td>
<td>Human or IT entity interacting to the smart metering system</td>
</tr>
</tbody>
</table>

In this project, only some of the actors are involved in the use cases that have described in previous section. These actors are Gateway Operator, Gateway administrator, and user or external entities. The rest of actors are not in the scope of this project. On the other hand in the smart home we have different roles and entities including parent, child, gateway operator, landlord, tenant, guest and so on.

5.4 **Appliance/ Device Required Services**

Any typical home device or appliance supports eight basic functional services. Some of the functions are considered in two security architectures e.g. turn on/off the device and get status of the device but some of them are in the scope of only one of them like reload or hibernate the device. More information of each device function will be described in the following sections.

5.4.1 **Turn on Device**

The actor tries to turn the device on. Device id is a unique identifier of any device in the home. And the result is the status of the device after ‘Turn on’ action. Authorization system defines the availability of this function based on the actor’s role. This function is general in two security architectures.
5.4.2 Turn off Device

The actor tries to turn the device off. Device id is a unique identifier of any device in the home. The result is status of the device after ‘Turn off’ action. Authorization system defines the availability of this function based on the actor’s role. This function is general in two security architectures.

5.4.3 Hibernate Device

The actor tries to hibernate the device. Device id is a unique identifier of any device in the home. And the result is status of the device after ‘Hibernate’ action. Authorization system defines the availability of this function based on the actor’s role. This function is only available in the smart metering system security architectures.

5.4.4 Stand by Device

The actor requests to stand by the device. Device id is a unique identifier of any device in the home. And the result is status of the device ‘Stand by’ action. Authorization system defines the availability of this function based on the actor’s role. This function is only available in the smart metering system security architectures.

5.4.5 Reload Device

The actor requests to reload the device metering middleware. Device id is a unique identifier of any device in the home, and the result is status of the device after ‘Reload’ action. Authorization system defines the availability of this function based on the actor’s role. This function is only available in the smart metering system security architectures.

5.4.6 Restart Device

The actor requests to restarts the device. Device id is a unique identifier of any device in the home. The result is status of the device after ‘Restart’ action. Authorization system defines the availability of this function based on the actor’s role. This function is only available in the smart metering system security architectures.

5.4.7 Get Status of Device

The actor tries to get the status of the device. Device id is a unique identifier of any device in the home, and the result the status of the device. Authorization system defines the availability of this function based on the actor’s role. This function is common in two of security architectures.

5.4.8 Get List of Controlled Devices

Each gateway asks the other gateway to send the list of devices that are controlled by that gateway. The result is a list of devices.
6 ARCHITECTURE AND DESIGN ASSUMPTIONS

In this section, the architecture and design of the secure communication protocol between SHG and SMHG will be specified. In the preceding sections different protocols of Federated authorization and authentication have been discussed and finally SAML and XACML proposed as a well-structured and flexible protocols for communication between these two gateways. At the rest of this section, all design and architectural assumptions will be described precisely.

6.1 Identity Server and Role Attribute

First Assumption in the design of this solution is on the identity server and role attribute of the user identities. Identity servers are the repositories for identity attributes. Identities could be individuals, resources, machines, IT components and so on. General information of individuals e.g. first name, last name, organization name, country of living, home address, work address and all other attributes are almost same in different systems or domains, so usually all of these attributes are stored and requested on demand in the identity servers. On the other hand, there are some other attributes of the identity that are dynamic. It means the attribute value may differ from one system/domain to another system/domain. In authentication area, systems or domains called ‘Realm’. These attributes usually are not stored in the identity providers and each concrete system or domain is responsible to preserve them, roles of the identities are one of the most important dynamic attribute of identities. So any system has an internal Role Management System (RMS). RMS can be as simple as a database table containing subject name (or any other unique attribute of the identity like email, personal OpenID and so on) and the roles of identity (either a set of roles- flat roles set- or hierarchical roles) but in the complicated systems and domains it will be much more structured and contains more attributes in addition to privileges of actions.

SAML-based identity providers usually support role attribute (claim) in authentication attribute responses but it could be the same value for all realms. So, in this scenario with two systems only one role set could be defined in the SAML provider and it are not satisfying security requirements of these systems. Also OAuth and OpenID are not supporting role attribute (claim).

In the SAML authentication, simplest solution is inter-realm conventions i.e. naming conventions, for example every gateway has a unique prefix in its roles in the SAML provider so service providers can filter the authentication response and access to appropriate role set of the identity in their systems.

Generally speaking, having an internal role management system is a better solution rather than role naming convention. Any service provider is responsible to manage identity roles in the local system. This solution makes a new requirement to the system, one more role management server like WSO2 Identity server or a simpler role management repository like a database table. For the simple domains/systems like secure home and smart metering system it is a relatively high cost to implement, maintenance and support a new server but a simple repository like a database table is much more rational.

Furthermore, it could be a better choice to use either naming convention (only available in SAML) for each realm or establishing a role repository in each system (applicable in all protocols). In the first method, every realm uses a specific naming convention such as particular
prefix or unique hashing values for their roles, so every time the identity servers returns identity attributes or claims, the relying party could be able to extract all related roles of the identity and throw away all other roles. Although first method doesn’t required any extra role management system but any service provider should negotiate to identity server and make a contract on naming conventions on identity roles. So any changes on the roles like renaming, deleting and adding role needs a communication to identity server. In the second method, there is no dependency between identity server and service providers, so service provider is the only element of handling identity roles.

6.2 Identity Role in Inter-gateway Communication

Second assumption, is about role being exchanged in the authorization request between gateways. Actually it is not important which user is requesting any service between different realms. The only important issue is managing requests between two gateways. For example it doesn’t care for utility provider the landlord or the tenant or any other staffs are requesting to turn on/off an appliance in the home. Utility providers only consider requests not requesters from the secure home gateway to turn on/off an appliance or any other requests. So, in this solution role pseudonym is used as a representative identity between two domains, and user identities are not exchanged in authorization request among these two gateways at all.

6.3 Resource’s Action Identification

Third assumption is on resource and action identification. It might be possible to have different standards for resource and action identification, in this research both gateways are using the same and simple model of resource and action identification. Resource is functions of the devices like turn on, turn off, reload, and restart and so on. Actions are device ids containing type, device division, and device category. It is somehow strange to suppose device’s function as resources and device id as action, but it is a simplification strategy in authorization policy definition that will be discussed in more details in Implementation section.

6.4 Business Rules vs. Authorization Policy

The Last assumption is about combination of business rules and authorization policy. Suppose that a customer has an agreement to utility provider to turn on/off particular devices or appliances at a time period in specific days. Certainly, it is compulsory to let a certain role in the smart metering system to turn on/off the device and also prevent other roles to do opposite actions during that period of time. Furthermore, it is required to define not only access conditions for smart metering system to execute this specific business rules but define some more policies to prevent other roles like smart home representative role or pseudonym identity to do any action against this policy.

Administration of the SHG and the SMSG is responsible to align, change and update any required policy, when a new Service-Level Agreement (SLA) is established between the utility provider and the customer. For example, when a customer make a new contract with the utility provider to start charging an electric vehicle during the night (when the price of electricity usage
is less that day cost) the administrator of SM should add a new access permission to the SHG pseudonym role during that time period. So in the smart metering system administrator role should have access to turn on the device during nights for example 2-7AM and simultaneously all other roles including the smart home representative role or pseudonym identity haven’t any access to turn off the device. In this case, the administrator should define all required policies to support this new SLA. The same scenario is possible in the secure home system, for example the secure home application not only permits inhabitants to turn on/off devices but allows defining a duration or period for any action or validity of request action. As an example, parents have permit to turn off the TV set remotely when children are busy to their home works and children has not any permission to turn on the TV set. Actually it is a new business rule or requirement, and should be compiled to several authorization policies. In this implementation the administrator of SHG is responsible to interpret all requirements and provides all authorization policies in the PAP.
As a proof of concept, it is necessary to have a basic implementation of the proposed model. In this section, all details of implementation will be specified. This implementation is based on most famous open source applications in Java language. Java Language and Java Server page (JSP) is used to implement demonstration application for both SHG and SMSG.

According to final discussions about federated authorization and authentication (SSO) standards, SAML and XACM are chosen for the implementation of the secure communication between two SMSG and SHG gateways. OpenID and OAuth are implemented as a demonstration application too. WSO2 Identity Server supports both protocols and could be used as SAML provider or SAML asserting party and XACML PDP. WSO2 Identity server is a Java based application supporting SSO, Role Based Access Control (RBAC), Attribute or Claim based access control via XACML 2.0, 3.0. All WSO2 products are 100% open source and released under the Apache License Version 2.0. Contributing to key international standards organizations and foundations such as W3C, OASIS, OpenID Foundation, Infocard Foundation, Microsoft’s Interop Vendor Alliance, AMQP Working Group, oCERT, and Cloud Security Alliance, we possess the critical skills and experience to competently manage your problems in an open and transparent environment (WSO2 About, 2012).

7.1 General Architecture of Gateways

Based on use case descriptions in section 5, the general architecture of SHG and SMSG are shown in Figure 26 and 27. The simple scenarios #1, 2 as specified in section 5 are shown in Figure 26 and the collaborative scenario of two gateways is shown in Figure 27.

PEP and Federated Identity Server (supporting SSO use case) are two instances of WSO2 Identity Server. PEP is a server machine consists of Application Server (Apache Tomcat) and Database Server (Apache Derby) and PEP server module implemented in Java and JSP. All communication channels between components of this topology are secured by HTTPS and TLS/SSL transport protocols.

The Basic flow of Use case 1, 2 is as described in the following paragraph:

1. User agent request Authenticate request from PEP.
2. PEP redirects user to Identity Server by SAML Authentication Request.
3. Identity server run authentication process and provides an SAML authentication response to PEP through user agent.
4. PEP sends a XACML authorization request to PDP.
5. PDP provides a XACML authorization response to PEP.
6. PEP provides service (warning message in alternative flows) to the end user or user agent.
The Basic flow of Use case 3 is as described in the following paragraph:

1. User agent request Authenticate request from PEP.
2. PEP redirects user to Identity Server by SAML Authentication Request.
3. Identity server run authentication process and provides an SAML authentication response to PEP through user agent.
4. PEP sends a XACML authorization request to local PDP.
5. Local PDP provides a XACML authorization response to PEP.
6. PEP sends a XACML authorization request to remote PDP.
7. Remote PDP provides a XACML authorization response to PEP.
8. PEP combine authorization response of local and remote authorization server and make a decision then provides service (warning message in alternative flows) to the end user or user agent.
### 7.2 Setting up the Demonstration Environment

Configuring the demonstration environment consists of four steps. These steps will be briefly described here but a full installation guideline is presented in appendix A:

1. Since WSO2 servers are Java based applications, the first step of preparing the environment is to install the Java language package, i.e., Java Development Kit (JDK) and Java Runtime Environment (JRE).
2. The second step is installing the appropriate edition of WSO2 Identity server for PDP and Identity Server machines.
3. The third step is setting up an application server with a Java Servlet container like Apache Tomcat in the PEP.
4. The final step is installing and configuring the database server like Apache Derby DB in the PEP machines.

For more information about installation and configuration of the environment, please refer to Appendix A.

### 7.3 SAML-based Authentication

There are some considerations in the implementation of SAML that will be described in this section. Identity Server Configuration and SAML Message Format.

#### 7.3.1 WSO2 Identity Server Configuration

The first step in the configuration of SAML environment is setting up the identities store. Configuring the Identity store includes two sub-tasks: first is adding roles and second is defining users or identities. First, the required roles like ‘Tenant’, ‘Landlord’, ‘admin’, ‘Smart Meter Service’, ‘Smart Home Service’, ‘Grid Operator’ and ‘Meter Operator’ will be added to the Identity store. Then, some users will also be added to the identity provider. Please refer to Appendix A for more information.

The second step is to configure the SAML Provider (AP). Usually, every provider needs some basic settings, e.g., Issuer name, consumer name, public key of the Relying Party of Service provider, and attribute list or claim list requested by the Service provider. WSO2 Identity Server supports a variety of settings and configuration as SAML Identity provider. A screenshot of WSO2 Identity server SAML panel is shown in Figure 28.
7.3.2 SAML Message Format

As it mentioned, SP-Initiated SSO with Redirect and POST Bindings is used in the implementation of SAML in this project. The SAML specification defines several messages will be exchanged between Service Provider and Identity Provider in this use case. The basic flow of this use case has being shown in Figure 19:

1. First of all, client application (User Agent or Browser) initiates request to a service from server (Message#1), Server makes a ‘SAML Authentication Request’ Message to the Identity Server as ‘HTTP Get Redirect’ method. A typical SAML Authentication Request in the system is shown in Figure 29.

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:protocol">
    <soap:Header>
        <AuthnRequest xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
            <Issuer>http://localhost:8080/saml2_demo/consumerSAML</Issuer>
            <NameIDPolicy AllowCreate="true"
                SPNameQualifier="Isso"/>
            <RequestedAuthnContext Comparison="exact"/>
            <AuthnContextClassRef>
                urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
            </AuthnContextClassRef>
        </AuthnRequest>
    </soap:Header>
    <soap:Body>
        <AuthnRequest>
            <Issuer>http://localhost:8080/saml2_demo/consumerSAML</Issuer>
            <NameIDPolicy AllowCreate="true"
                SPNameQualifier="Isso"/>
            <RequestedAuthnContext Comparison="exact"/>
            <AuthnContextClassRef>
                urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
            </AuthnContextClassRef>
        </AuthnRequest>
    </soap:Body>
</soap:Envelope>
```

2. Browser or User Agent sends the SAML Request to the Identity Server.
3. Identity Server initiates the log in process, and gets authentication information through Messages #3, 4.
4. Then Identity Server replies user agent with a ‘SAML Authentication Response’ message wrapped in a ‘HTTP Post Redirect’ message (Message #5, 6). An example of SAML Response Message is shown in Figure 30.
One of the most important security considerations over SAML messages is digital signature of the response message. The value digital signature is calculated based on ‘SAML Authentication Response’ and added to the response. Service Provider will be able to verify the integrity of the SAML Response message by this signature. A sample of SAML Authentication response signature node is shown in Figure 31. This signature is generated by Identity Server based on XML Signature mechanism. Service Provider reads the SAML Response Messages and provides an appropriate message for the Browser either authenticated or not authenticated states.
The next step of implementation phase is configuring the XACML-based Authorization. WSO2 Identity server not only plays as an identities store but provides different services for PAP and PDP. This section is continued with XACML policy definition and XACML Message Format.

### 7.4.1 XACML Policy Definition

According to XACML’s policy definition, every simple policy consists of action, subject and resource.

- **Actions** are based on the functional requirements described in section 5.4. Actions could be some values including ‘Turn on’, ‘Turn off’, ‘Reload’, ‘Restart’, ‘Hibernate’ and so on.
- **Since RBAC authorization system is used in this system, subject is the role of end users like ‘Tenant’, ‘Landlord’, ‘Meter Operator’ and so on.**
- **Resources** are a combination of device’s attributes including ‘Type’, ‘Division’ and ‘Category’. These attributes were defined in section 5.1.

Based on the customer requirements, authorization policies will be defined in any system. All of details of policy definition in this system are explained in Appendix A. To simplify the Policy definition Action and Resource are swapped. As an example, the policy of ‘Turn on’ device in the secure home gateway is shown in the table 8:
Table 8. SHG Turn on Policy

<table>
<thead>
<tr>
<th>Resource</th>
<th>Role</th>
<th>Action</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Type</td>
<td>Division</td>
</tr>
<tr>
<td>Turn on admin</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Tenant</td>
<td>Major</td>
<td>Washing</td>
<td>Dish Washer</td>
</tr>
<tr>
<td>Tenant</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Landlord</td>
<td>Major</td>
<td>Refrigeration</td>
<td>Refrigerator</td>
</tr>
<tr>
<td></td>
<td>Minor</td>
<td>Cooling</td>
<td>Air Conditioner</td>
</tr>
<tr>
<td>Smart Meter</td>
<td>Major</td>
<td>PEV</td>
<td>PEV</td>
</tr>
<tr>
<td>Service(Role</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Pseudonym)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

WSO2 Identity Server Provides different features to define policies in the server including:

- Import from file: you can write you policy in text formatted file and import to the server.
- Registry: you can used predefined policy from Identity server registry.
- Policy Set and Entitlement Policy console: there is a graphical web user interface to define policy and entitlement policy in the system. Normal view of Entitlement definition in the WSO2 Identity server is shown in Figure 32.
Detailed view of any Entitlement Rule is shown in Figure 33. There are several elements for any Entitlement policy including ‘Rule combination algorithms’, ‘Resource’, ‘Subject’, ‘Action’ and A set of ‘Rules definition’. A detail view of Rule definition is shown in Figure 33.
7.4.2 Assumptions and Considerations

In the implementation of this project, there are some assumptions on XACML that will be defined in the following:

1. For any user case specified in the section 5.2, there are an associated entitlement policy in the PDP including ‘TurnOn’, ‘TurnOff’, ‘Hibernate’, ‘Standby’, ‘Reload’, ‘Get Status’….
2. Every rule is based on the user role (as subject), a concatenation string of appliance’s attributes (as Action) and a rule effect (either permit or deny).
3. Regarding to different ‘Rule Combining algorithm’, the priority of rules is very important. So it should be considered what ‘Rule Combining Algorithm’ is chosen and what order is desired for the rules in the Entitlement Policy.
4. ‘Entitlement Policy’ in PAP console should be manually enabled, otherwise the policy is not considered in the WSO2 Identity Server at all.

7.4.3 XACML Policy and Message Format

After defining all entitlement policies and enabling them in WSO2 Identity server, the PDP will be able to responses to XACML authorization requests. A sample ‘Turn off’ Entitlement Policy in the PDP is shown in Figure 34. For any of the subject e.g. ‘Tenant’, ‘LandLord’ and so on, there are two rules with the unique naming pattern as ‘roleAllow’ and ‘roleDeny’. According to below policy rule, tenants have not allowed to ‘Turn off’ any ‘Dishwasher’ or ‘Washing machine’ in the home.

```
<Policy xmlns="urn:oasis:names:tc:xml:policy:schema-oo" PolicyId="SSO-TURNOFF" RuleCombiningAlgorithm="first-applicable">
  <Rule Effect="Deny" RuleId="adminAllow">
    <SubRuleEffect>
      <AttributeMatch Id="urn:oasis:names:tc:xml:1.0:definition:attributeMatch">
        <AttributeValue DataValue="http://www.w3.org/2001/XMLSchema#string">Tenant</AttributeValue>
      </AttributeMatch>
      <AttributeRepresentation Id="urn:oasis:names:tc:xml:1.0:definition:attributeRepresentation">Tenant</AttributeRepresentation>
    </SubRuleEffect>
    <Condition>
      <Apply Function="urn:oasis:names:tc:xml:1.0:definition:attribute-any-of">
        <AttributeValue DataValue="http://www.w3.org/2001/XMLSchema#string">MajorWashingMachine</AttributeValue>
        <AttributeValue DataValue="http://www.w3.org/2001/XMLSchema#string">MajorDishwasher</AttributeValue>
      </Apply>
    </Condition>
  </Rule>
  <Rule Effect="Permit" RuleId="tenantDeny">
    <SubRuleEffect>
      <Subject>
        <AttributeMatch Id="urn:oasis:names:tc:xml:1.0:definition:attributeMatch">
          <AttributeValue DataValue="http://www.w3.org/2001/XMLSchema#string">Tenant</AttributeValue>
        </AttributeMatch>
      </Subject>
      <Condition>
        <Apply Function="urn:oasis:names:tc:xml:1.0:definition:attribute-any-of">
          <AttributeValue DataValue="http://www.w3.org/2001/XMLSchema#string">MajorWashingMachine</AttributeValue>
        </Apply>
      </Condition>
    </SubRuleEffect>
  </Rule>
  <Rule Effect="Permit" RuleId="tenantAllow">
    <SubRuleEffect>
      <AttributeMatch Id="urn:oasis:names:tc:xml:1.0:definition:attributeMatch">
        <AttributeValue DataValue="http://www.w3.org/2001/XMLSchema#string">Tenant</AttributeValue>
      </AttributeMatch>
    </SubRuleEffect>
    <Condition>
      <Apply Function="urn:oasis:names:tc:xml:1.0:definition:attribute-any-of">
        <AttributeValue DataValue="http://www.w3.org/2001/XMLSchema#string">MajorWashingMachine</AttributeValue>
      </Apply>
    </Condition>
  </Rule>
  <Rule Effect="Deny" RuleId="landlordAllow">
    <SubRuleEffect>
      <AttributeMatch Id="urn:oasis:names:tc:xml:1.0:definition:attributeMatch">
        <AttributeValue DataValue="http://www.w3.org/2001/XMLSchema#string">Tenant</AttributeValue>
      </AttributeMatch>
    </SubRuleEffect>
    <Condition>
      <Apply Function="urn:oasis:names:tc:xml:1.0:definition:attribute-any-of">
        <AttributeValue DataValue="http://www.w3.org/2001/XMLSchema#string">MajorWashingMachine</AttributeValue>
      </Apply>
    </Condition>
  </Rule>
  <Rule Effect="Permit" RuleId="landlordDeny">
    <SubRuleEffect>
      <AttributeMatch Id="urn:oasis:names:tc:xml:1.0:definition:attributeMatch">
        <AttributeValue DataValue="http://www.w3.org/2001/XMLSchema#string">Tenant</AttributeValue>
      </AttributeMatch>
    </SubRuleEffect>
    <Condition>
      <Apply Function="urn:oasis:names:tc:xml:1.0:definition:attribute-any-of">
        <AttributeValue DataValue="http://www.w3.org/2001/XMLSchema#string">MajorWashingMachine</AttributeValue>
      </Apply>
    </Condition>
  </Rule>
  <Rule Effect="Deny" RuleId="smartHomeServiceDeny">
    <SubRuleEffect>
      <AttributeMatch Id="urn:oasis:names:tc:xml:1.0:definition:attributeMatch">
        <AttributeValue DataValue="http://www.w3.org/2001/XMLSchema#string">Tenant</AttributeValue>
      </AttributeMatch>
    </SubRuleEffect>
    <Condition>
      <Apply Function="urn:oasis:names:tc:xml:1.0:definition:attribute-any-of">
        <AttributeValue DataValue="http://www.w3.org/2001/XMLSchema#string">MajorWashingMachine</AttributeValue>
      </Apply>
    </Condition>
  </Rule>
  <Target>
    <Policy/>
  </Target>
</Policy>
```

Figure 34. Entitlement Policy File

The next step is configuring the client application to initiate the XACML authorization process. The client application ‘User Agent’ needs to know not only local and remote PDP URL but the
server certificate of the both authorization servers, so it may connect to those servers and requests some XACML Authorization Request.

After configuring the environment User Agent is ready to request any authorization request based on the desired User requests to access different features of the secure home or smart metering system.

According to XACML data flow described in section 3.1.1, user agent or client application requests a XACML authorization request to the local PDP. An example of XACML authorization request is shown in Figure 35. Any XACML Request should contain ‘Resource’, ‘Subject’ and ‘Action’. This sample is a XACML request to ‘Turn on’ a ‘Freezer’ by ‘Tenant’.

```
<Request xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Resource>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema:string">
      <AttributeValue>TURNON</AttributeValue>
    </Attribute>
  </Resource>
  <Subject>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
               DataType="http://www.w3.org/2001/XMLSchema:string">
      <AttributeValue>Tenant</AttributeValue>
    </Attribute>
  </Subject>
  <Action>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema:string">
      <AttributeValue>MajorRefrigartionFreezer</AttributeValue>
    </Attribute>
  </Action>
</Request>
```

Figure 35. XACML Authorization Request

When the PDP receives a new XACML authorization request, first evaluates the request and then provides the PEP with an appropriate XACML response message. XACML response message is very simple and contains the decision made about the authorization request. The decision may have four different values including ‘Permit’, ‘Deny’, ‘NotApplicable’ and ‘Indeterminate’. A sample ‘XACML Authorization Response’ message is shown in Figure 36:

```
<Response xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <Decision>NotApplicable</Decision>
</Response>
```

Figure 36. XACML Authorization Response

The PDP returns ‘NotApplicable’ decision result when no available policy matches the XACML Request, and ‘Indeterminate’, if any of ‘Subject’, ‘Resource’, ‘Action’ or ‘Environment’ would be indeterminate.
### 7.5 WSO2 Entitlement Architecture

The entitlement architecture is the most important part in authorization process. A model of Entitlement Architecture in WSO2 Identity Server is shown in Figure 38. An overall use case description for the entitlement in WSO2 Identity servers is:

1. User Agent or Browser creates and ‘EntitlementClient’ object and authenticates itself to the ‘AuthenticationAdmin’. To access any services of Entitlement in the XACML PDP, it is mandatory to be authenticated to the server, so no one can be authorized to use any of the identity server services without valid authentication. This authentication is Client Authentication and is automatically done in the PEP by default settings.

2. ‘AuthenticationHandler’ is responsible to manage secure channel between client and WSO2 Identity Server (authorization server). Any authenticated request is forwarded the to ‘EntitlementService’. The ‘EntitlementService’ is the engine of the PDP evaluation.

3. Finally ‘EntitlementService’ provides a XACML Response Message and returns to the ‘EntitlementClient’ in the user agent (browser).

![Figure 37. WSO2 Entitlement Architecture](image)

### 7.6 PDP Flowchart

When the PDP receives a new XACML authorization request, first of all evaluates the request with all available policies and then compile the results with respect to the ‘Policy Combining Algorithm’ and finally provides final decision result. The PDP sends back authorization response to the PEP, so the PEP may run any required actions based on the authorization response result. The XACML authorization request evaluation process in PDP is shown in Figure 37.
The PEP is the first and only contact point for User agents or browsers in the XACML authorization system. The PEP handles all requests, receives all authorization request and forward them to the PDP. In addition returns all response messages from the PDP to browsers or User agents. The flowchart of PEP is shown in Figure 39.a short description of this flowchart is:

1. When the PEP receives a service request, the first step is authorization checking.
2. It is highly recommended to check access privilege or authorization of the request before responding to any request. Furthermore, even though the authorization process is a time consuming operation but it should be done prior to any other tasks e.g. check status of appliance and any other things in the PEP. If the user is authorized to access the requested services based on local authorization policy, the PEP has to get the remote server’s authorization decision on the request if the request is collaborative. Finally the PEP can make final decision for the request authorization.
3. The next step is running the requested service and provides appropriate results for the user. Change status is general action name and could be any of appliance’s functions like ‘Turn on’, ‘Turn off’, ‘Reload’ and so on. After running the service, the results are available to send to the browser or user agent.
One important issue is how the PEP reports authorization response status(s) to the browsers. For example, users need to know which gateway is denying the requested service and what is the reason behind that. Knowing the reason of authorization issues guides users to manage those issues. Customers have to contact to customer support department of the utility provider to solve any issue to access the home appliances regardless of the customer’s service agreement.

In this implementation, appliance information database keeps the current status of each appliance e.g. ‘on’ or ‘off’ and so on. But, in real implementation it is recommended to examine status of the appliances in advance to any other action request in the PEP.
8 DISCUSSIONS AND CONCLUSION

In this section, conclusion and discussions will be presented. Today’s distributed systems and architectures require to be integrated for different reasons e.g. user friendliness of the systems, less maintenance and less development cost of systems and so on. So it is a general requirement for all distributed systems to be communicated to each other. One of the most important and valuable information being exchanged between different enterprise architecture are Authorization and Authentication data that should be protected.

8.1 Discussion

In this section, some of the most important issues and discussions on the proposed model will be presented. The first and most critical issue in the distributed information systems is the federation of identity servers. Federation of Identity servers (SSO) has two major issues that will be described overall:

1. Every individual may have several identities (attribute set) for different building or houses, for example a person might be the owner of a building, the ‘Tenant’ of other, the ‘Service and Repair Staff’ for some other buildings and the ‘Landlord’ of any other building in the region, so the Identity Provider should support not only user driven token like i.e. user name and password but the geographical information and environment variables like building or home address.

2. The other issue is managing identity roles in different systems. The identity server returns all of the roles associated with a user, if all service providers rely on the Federated Identity Provider, they should filter the roles whenever they get an identity attribute set sent by identity provider. There are two possible ways to handle this issue, first having a naming convention between service provider and identity provider, so the service provider is able to filter the authentication response claims based on this naming convention on the roles, another solution is delegation of role management to the service providers, so all service providers have to manage the identity roles in their site.

Some other issues and considerations in the communication of two gateways are:

1. Certificate validation and verification. All communication networks are based on TLS/SSL transport security, so it is necessary to have a precise certificate validation, revocation and verification on each component. Certificate handling should be part of consideration in the final architecture.

2. The next issue is how to handle the cases when one of the gateways isn’t available or not working properly. In this implementation all of the collaborative requests will be drops if the required gateways are not available.

3. XACML is a heavy weight authorization protocol, so it is possible to have some kind of performance issues like customer’s response time in the system. So it is necessary to evaluate the performance criteria for each particular use case.

4. Claim-based authentication and STS are partially used in this implementation but they have more functionality to support complicated scenarios that could be useful to be considered more and more. It is possible to enhance the performance and reliability of the system based on these valuable features in the Identity servers.
8.2 Conclusions

As it mentioned there are different distributed authentication and authorization protocols but OASIS SAML and XACML are most noticeable and recommended protocols in this era. So these protocols have been chosen for the implementation of this project. It must be mentioned OAuth and OpenID as two common federated authentication protocols will be considered and implemented as simple as possible as a proof of concept demonstration. Below is a summary of advantages of using SAML and XACML as authentication and authorization data exchange protocol:

1. SAML and XACML are flexible and ease of use XML based protocols to exchange all kind of secure information e.g. authorization, authentication and identity attributes. There are several open source implementation of these protocols in almost all active and well-known programming languages like Java, C#, C++, Python.

2. OASIS is the creator and supporter of these protocols, so they are highly consistent to each other.

3. And both of these protocols are XML-based open standard protocols. XML standard enables all vendors to extend and customize them upon their particular requirements.

Having a general architecture to exchange secure data between gateways was the main purpose of this research, using SAML and XACML protocols satisfies all functional and non-functional requirements of the research problem. Furthermore, any Secure Home Gateway (SHG) vendor and Smart Metering Gateway System (SMGS) vendor who support SAML and XACML in their gateways are able to develop this model as a protected and secure model of inter-gateway collaboration. And finally they have a well-established secure topology in their customer’s buildings.
9 RECOMMENDATIONS AND FUTURE WORK

In this section, recommendations and future works will be described as a final step of the research report. This section is a summary of all most important recommendations to setup the environment and executing the sample applications, and this section also covers not only weaknesses but possible approach to improve the functionality and usability of this model. Please read this part carefully if you are interested to continue on this research either more reliable and secure model or providing different implementation.

9.1 Recommendations

Since this implementation is based on some open source application, it is highly recommended to consider WSO2 Identity Server software versions for each architectural component like Identity Server and Authorization Server. There are some inconsistencies in different versions of this software, and unfortunately there is not a good documentation and user manual about unpredictable issues like serialization and coding issues. A precise installation guides to setup and configure the environment is available in Appendix A. Based on this installation guides all of the components in any of the gateways will be prepared at a couple of hours.

9.2 Future work

In this research, I tried to develop a secure communication channel between secure home and smart metering system gateways. This secure architecture contains authentication and authorization functions. This research tried to promote the privacy and security of inter-gateway communication but there are some other works could be done to improve this model in future research. These works are as listed below:

1. First, Client application is a common web application; it is a good idea to develop a more user-friendly client application especially for mobile devices.
2. Second, this project is a demonstration of the SHG and SMGS systems, so in the back-end side there is no real environment or homes containing some real appliances. It is another nice idea to evaluate this architecture in an actual environment with some home appliances. The performance of the gateways is a key factor of evacuation, so response time of request processing time should be considered further.
3. Third, developing full OAuth support implementation of the system. Because the OAuth is more flexible and less complex in mobile devices.
4. Forth, developing non Java-based applications such as Microsoft C++/C# and any other client side application. There are several Microsoft based applications for secure home and smart metering systems, so developing a secure protocol of communication in these platforms as a third party module could be very considerable in the market.
10 References


OpenID Connect Basic Client Profile. (2013). Retrieved from OpenId.net: http://openid.net/specs/openid-connect-basic-1_0.html


11 APPENDIX A: INSTALLATION GUIDE

11.1 General Setup

1. Download JRE (http://www.oracle.com/technetwork/java/javase/downloads/index.html) and set the environment variable JAVA_HOME to the local JRE path.
2. Download JDK (http://www.oracle.com/technetwork/java/javase/downloads/index.html) and configure it.

11.2 Database Setup

1. Download Apache Derby (http://db.apache.org/derby/derby_downloads.html) and extract the archive file to local machine.
2. The Command to run Apache Derby server is: "[DERBY_HOME]/bin\startNetworkServer.bat"
3. To setup Derby database for this demonstration, run the “ij” scripting tool from “[DERBY_HOME]/bin\ij.bat” location and execute below instructions:
   a) Create and open a connection to the database using the Derby client driver:
      “CONNECT 'jdbc:derby://localhost:1527/appliances;user=root;password=root;create=true';”
   b) Create the Appliance table containing the information of home appliances using the following statement:
      “CREATE table SHG_Appliance(id bigint, name varchar(100), description varchar(150), type varchar(20), status varchar(100), division varchar(100), category varchar(100), scope varchar(100));”
      “CREATE table SMG_Appliance(id bigint, name varchar(100), description varchar(150), type varchar(20), status varchar(100), division varchar(100), category varchar(100), scope varchar(100));”
   c) Insert appliance information to the database, as a sample data you can use the “SHG_Appliance.txt” and “SMG_Appliance.txt” records to insert in the database. This file contains all SQL insert commands to populate these tables.
   d) Create the Role table containing the information of home users by running the following statement:
      “CREATE table SHG_Role(name varchar(200), roles varchar(200));”
      “CREATE table SMG_Role(name varchar(200), roles varchar(200));”
   e) Insert roles information to the database, as a sample data you can use the “SHG_Roles.txt” and “SMG_Roles.txt” records to insert in the database. This file contains all SQL insert commands to populate these tables.
### 11.3 Web Server Setup

a) Download Apache Tomcat ([http://tomcat.apache.org/download-70.cgi](http://tomcat.apache.org/download-70.cgi)) and extract the archive file to local machine.

b) Configure Apache Tomcat Server to support SSL or Https:

a) Generate Key store: to generate a new key pair the command pattern is:

```
"-genkey {-alias alias} {-keyalg keyalg} {-keysize keysize} {-sigalg sigalg} {-dname dname} 
{-keypass keypass} {-validity valDays} {-storetype storetype} {-keystore keystore} [-
storepass storepass] [-providerprovider_class_name] {-v} {-Jjavaoption}"
```

(keytool - Key and Certificate Management Tool, 2013)

b) So we could use this command to create a key pair for SHG’s Tomcat server:

```
"keytool -genkey -alias shgkeystore -keyalg RSA -keystore c:\shgkeystore".
```

During the process of key generation you must define all required information including username and password to access the keys. This command provide a self-signed certificate protected by user name and password in the key repository.

As a simplicity rule, in this implementation for smart home gateway we use “smarthome” as username and password for the key repository of the tomcat server in this gateway. On the other hand “smartmeter” is used as username and password for key store of smart metering gateway’s Tomcat server. Please pay attention, the default value for validity is 90 days and key size is 1024 bits.

c) Checking new key store by printing out key repository data by this command:

```
"keytool -list -alias shgkeystore"
```

d) Change the Tomcat’s server configuration file at “[TOMCAT_HOME]\conf\server.xml”, modify it by adding a connector element to support for SSL or https connection as follows:

```
"<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true" 
maxThreads="150" scheme="https" secure="true" 
clientAuth="false" sslProtocol="TLS"
keyStoreFile="c:\shgkeystore "
keyStorePass="password" />"
```


f) Run the Apache Tomcat server.

g) Deploy the Web Application WAR file in the server, i.e. “saml2.demo.war”.

h) Open the Web application web.xml file located in “[TOMCAT_HOME]\webapps\saml2.demo\WEB-INF\web.xml” and edit all settings based on your configuration.

i) Web.xml file -the Secure Home Gateway- contains different configurations as described in the following table, for the Smart Metering Gateway the settings should be changed respectively.
### Table 9. Secure Home Gateway Web Application Settings

<table>
<thead>
<tr>
<th>Servlet</th>
<th>Parameter</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML2ConsumerServlet</td>
<td>Issuer</td>
<td>The SAML2 Assertion issuer name</td>
<td>saml2.demo</td>
</tr>
<tr>
<td></td>
<td>IdpUrl</td>
<td>Identity Provider URL</td>
<td><a href="https://localhost:9443/samlsso">https://localhost:9443/samlsso</a></td>
</tr>
<tr>
<td></td>
<td>ConsumerUrl</td>
<td>Service provider URL</td>
<td><a href="http://localhost:8080/saml2.demo/consumerSAML">http://localhost:8080/saml2.demo/consumerSAML</a></td>
</tr>
<tr>
<td></td>
<td>AttributeConsumingServiceIndex</td>
<td>The index of Attribute Consuming Service</td>
<td>1111235732</td>
</tr>
<tr>
<td></td>
<td>ServerTrustStore</td>
<td>The address of certificate store of the Secure Home Gateway server</td>
<td>C:/Private/Thesis/wso2is-3.2.3-SHG/repository/resources/security/wso2carbon.jks</td>
</tr>
<tr>
<td></td>
<td>Username</td>
<td>The user name used to log in to admin service handler of SHG-PDP</td>
<td>admin</td>
</tr>
<tr>
<td></td>
<td>Password</td>
<td>The Password used to log in to admin service handler of SHG-PDP</td>
<td>Admin</td>
</tr>
<tr>
<td></td>
<td>RemoteServerTrustStore</td>
<td>The address of certificate store of the Smart Metering Gateway server</td>
<td>C:/Private/Thesis/wso2is-3.2.3-SHG/repository/resources/security/wso2carbon.jks</td>
</tr>
<tr>
<td></td>
<td>RemoteUsername</td>
<td>The user name used to log in to admin service handler of SMG-PDP</td>
<td>admin</td>
</tr>
<tr>
<td></td>
<td>RemotePassword</td>
<td>The Password used to log in to admin service handler of SMG-PDP</td>
<td>admin</td>
</tr>
</tbody>
</table>

### 11.4 SAML Setup

1. Download WSO2 Identity Server 4.0.0([http://wso2.org/more-downloads/identity-server/](http://wso2.org/more-downloads/identity-server/)) and extract the archive file to local machine. (NOTE: Attribute Profile of SAML is not supported in IS 3.x.x editions and also in 4.1.x is not supporting BASE64 marshaling/un-
marshaling at the moment.) WSO2 server will start on the port 9443 over https and it is nice for this server.

2. Login to the application and define below roles for this demonstration through this address in the Identity server application, “Home>Configure>Users and Roles >Roles”:

Table 10. Roles and Users

<table>
<thead>
<tr>
<th>Role Name</th>
<th>User name</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>admin</td>
<td>admin</td>
</tr>
<tr>
<td>LandLord</td>
<td>landlord</td>
<td>landlord</td>
</tr>
<tr>
<td>Tenant</td>
<td>tenant</td>
<td>tenant</td>
</tr>
<tr>
<td>SmartHomeService</td>
<td>smarthomeservice</td>
<td>smarthomeservice</td>
</tr>
<tr>
<td>SmartMeterService</td>
<td>smartmeterservice</td>
<td>smartmeterservice</td>
</tr>
<tr>
<td>GridOperator</td>
<td>gridoperator</td>
<td>gridoperator</td>
</tr>
<tr>
<td>MeterOperator</td>
<td>meteroperator</td>
<td>meteroperator</td>
</tr>
</tbody>
</table>

3. Define different users in the Identity server associated with different roles, as a simplicity rule at this demonstration, every role has an associated user with the same name and password with Role name and all letters in small face.

4. Now, it is time to configure the SAML2 SSO settings in the identity server, so go to “Home>Manage> SAML SSO” and define new service provider settings with the following values:

Table 11. SAML SSO Settings

<table>
<thead>
<tr>
<th>Gateway</th>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure Home Gateway</td>
<td>Issuer</td>
<td>SHG_WEB_APP</td>
</tr>
<tr>
<td></td>
<td>Assertion Consumer URL</td>
<td><a href="http://localhost:8080/">http://localhost:8080/</a> SHG_WEB_APP /consumerSAML</td>
</tr>
<tr>
<td>Smart Meter Gateway</td>
<td>Issuer</td>
<td>SMG_WEB_APP</td>
</tr>
<tr>
<td></td>
<td>Assertion Consumer URL</td>
<td><a href="http://localhost:8080/">http://localhost:8080/</a> SMG_WEB_APP /consumerSAML</td>
</tr>
</tbody>
</table>

5. For both gateway Attribute Profile option should be enabled and different claims like, role, nickname, full name, last name, organization… must be added to attribute profile.

6. The Identity server already is ready.

7. Command to run each WSO2 server is: “[SERVER_HOME]\bin\wso2server.bat”
11.5 OpenID Setup

1. Download WSO2 Identity Server 4.0.0(http://wso2.org/more-downloads/identity-server/) and extract the archive file to local machine. WSO2 server will start on the port 9443 over https.
2. Find the "identity.xml" file under “/[SERVER_HOME]/repository/conf” and set the “openid_skip_user_consent” to true.

OpenID setup is very trivial and straightforward. There are not any server specific configuration and client can connect to the server without any prior registration. The only required information is OpenID Identifier for any users like “https://localhost:9443/openid/admin” for admin user or “https://localhost:9443/openid/landlord” for landlord user.

On the other hand there are several OpenID Provider like Gmail, Facebook and etc. Simply you can use them for authentication users in this system. Some of most important OpenID Providers are listed in Table 13. It is necessary to remember that some of the OpenID providers requires to have username in the OpenID URL such has AOL, Google Profile, VeriSign and WSO2 OpenID Provider and the rest of them doesn’t including Google, Yahoo! and etc.

<table>
<thead>
<tr>
<th>No.</th>
<th>Site</th>
<th>URL Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>AOL</td>
<td>openid.aol.com/username</td>
</tr>
<tr>
<td>2.</td>
<td>BioID</td>
<td>mybioid.com/username</td>
</tr>
<tr>
<td>3.</td>
<td>ClickPass</td>
<td>clickpass.com/public/username</td>
</tr>
<tr>
<td>4.</td>
<td>Yahoo!</td>
<td>me.yahoo.com</td>
</tr>
<tr>
<td>5.</td>
<td>LiveJournal</td>
<td>username.livejournal.com</td>
</tr>
<tr>
<td>6.</td>
<td>MySpace</td>
<td>myspace.com/username</td>
</tr>
<tr>
<td>7.</td>
<td>WordPress</td>
<td>username.wordpress.com</td>
</tr>
<tr>
<td>8.</td>
<td>Blogger</td>
<td>username.blogger.com</td>
</tr>
<tr>
<td></td>
<td></td>
<td>username.blogspot.com</td>
</tr>
<tr>
<td>9.</td>
<td>Google Profile</td>
<td>google.com/profiles/username</td>
</tr>
<tr>
<td>11.</td>
<td>Verisign</td>
<td>username.pip.verisignlabs.com</td>
</tr>
<tr>
<td>12.</td>
<td>Typepad</td>
<td>username.typepad.com</td>
</tr>
<tr>
<td>13.</td>
<td>MyOpenID</td>
<td>username.myopenid.com</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>14. ClaimID</td>
<td>claimid.com/username</td>
<td></td>
</tr>
<tr>
<td>15. SceneID</td>
<td>username.sceneid.net</td>
<td></td>
</tr>
<tr>
<td>16. Clavid</td>
<td>username.clavid.com</td>
<td></td>
</tr>
<tr>
<td>17. Steam</td>
<td>steamcommunity.com/openid/</td>
<td></td>
</tr>
<tr>
<td>18. Orange</td>
<td>openid.orange.fr/username or just orange.fr/</td>
<td></td>
</tr>
<tr>
<td>20. Launchpad</td>
<td>launchpad.net/~username</td>
<td></td>
</tr>
<tr>
<td>21. Ubuntu</td>
<td>login.ubuntu.com</td>
<td></td>
</tr>
<tr>
<td>22. seznam.cz</td>
<td>username.id.seznam.cz username.id.email.cz</td>
<td></td>
</tr>
<tr>
<td>23. xlogon.net</td>
<td><a href="http://xlogon.net/username">http://xlogon.net/username</a></td>
<td></td>
</tr>
<tr>
<td>24. Hyves</td>
<td>hyves.nl</td>
<td></td>
</tr>
<tr>
<td>25. Mixi</td>
<td>mixi.jp</td>
<td></td>
</tr>
<tr>
<td>26. Virgilio.it</td>
<td>virgilio.it</td>
<td></td>
</tr>
<tr>
<td>27. Wirtualna_Polska</td>
<td>openid.wp.pl</td>
<td></td>
</tr>
<tr>
<td>28. Яндекс</td>
<td>username.ya.ru or username.some-another-YaRU-domain</td>
<td></td>
</tr>
<tr>
<td>29. Mail.ru</td>
<td>username.id.mailru-domain.ru</td>
<td></td>
</tr>
</tbody>
</table>

**11.6 OAuth Setup**

1. Download WSO2 Identity Server 4.0.0([http://wso2.org/more-downloads/identity-server/](http://wso2.org/more-downloads/identity-server/)) and extract the archive file to local machine. WSO2 server will start on the port 9443 over https.

2. Login to the application and Register New Application through this address in the Identity server application, “Home>Manage>OAuth>Register New Application”. Required fields to register a new application are:
Table 13. OAuth Application Settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAuth Version</td>
<td>1.0 or 2.0</td>
</tr>
<tr>
<td>Application Name</td>
<td>Any desired name</td>
</tr>
<tr>
<td>Callback Url*</td>
<td><a href="http://localhost:8080/playground/oauth2client">http://localhost:8080/playground/oauth2client</a></td>
</tr>
</tbody>
</table>

Identity server will automatically generates some new attributes for registered application that will be used in different steps of connection from client to server. These fields are listed in Table 15:

Table 14. OAuth Identity Server Auto-generated Settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Authorization Code</th>
<th>Implicit</th>
<th>Client Credentials</th>
<th>Resource Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Id</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Client Secret</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Access Token Url</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Authorize Url</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

11.7 XACML Setup

1. Download WSO2 Identity Server 3.2.3(http://wso2.org/more-downloads/identity-server/) and extract the archive file to local machine.
2. It is necessary to set an offset for each server in which multiple servers are running, you can set “Ports.Offset” in: “[SERVER_HOME]/repository/conf/carbon.xml”. If you running two different PDP in the same machine do the same things for both of them. At this demonstration the offset for SHG is 1(port: 9444) and for SMS is 11(port: 9454) on the https network protocol.
3. Import policy files in the administration panel, for this demonstration there are two policy file for each gate way e.g. “SHG_turnon_Policy.xml” and “SHG_turnoff_Policy.xml” for SHG and “SMG_turnon_Policy.xml” and “SMG_turnoff_Policy.xml” for SMG (import entitlement policy does not work properly in this software version, so it should be defined manually through Entitlement Admin Console). The policy set of the SHG is defined Table 16:
### Table 15. Secure Home Gateway Policy Set (Turn on)

<table>
<thead>
<tr>
<th>Resource</th>
<th>Role</th>
<th>Action</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Type</td>
<td>Division</td>
<td>Category</td>
</tr>
<tr>
<td>Turn on</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>admin</td>
<td>*</td>
<td>*</td>
<td>Permit</td>
</tr>
<tr>
<td>Tenant</td>
<td>Major</td>
<td>Washing</td>
<td>Dish Washer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Washing Machine</td>
</tr>
<tr>
<td>Tenant</td>
<td>*</td>
<td>*</td>
<td>Permit</td>
</tr>
<tr>
<td>Landlord</td>
<td>Major</td>
<td>Refrigeration</td>
<td>Refrigerator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Misc.</td>
<td>Air Conditioner</td>
</tr>
<tr>
<td>Minor</td>
<td>Cooling</td>
<td>Air Conditioner</td>
<td></td>
</tr>
<tr>
<td>Smart Meter</td>
<td>Major</td>
<td>PEV</td>
<td>PEV</td>
</tr>
<tr>
<td>Service(Role</td>
<td></td>
<td>Rechargeable</td>
<td></td>
</tr>
<tr>
<td>Pseudonym)</td>
<td></td>
<td>Battery</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>*</td>
<td>Permit</td>
</tr>
</tbody>
</table>

### Table 16. Secure Home Gateway Policy Set (Turn off)

<table>
<thead>
<tr>
<th>Resource</th>
<th>Role</th>
<th>Action</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Type</td>
<td>Division</td>
<td>Category</td>
</tr>
<tr>
<td>Turn off</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>admin</td>
<td>*</td>
<td>*</td>
<td>Permit</td>
</tr>
<tr>
<td>Tenant</td>
<td>Major</td>
<td>Refrigeration</td>
<td>Refrigerator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Misc.</td>
<td>Freezer</td>
</tr>
<tr>
<td>Tenant</td>
<td></td>
<td>*</td>
<td>Permit</td>
</tr>
<tr>
<td>Landlord</td>
<td>Major</td>
<td>Refrigeration</td>
<td>Refrigerator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Misc.</td>
<td>Air Conditioner</td>
</tr>
<tr>
<td>Minor</td>
<td>Cooling</td>
<td>Air Conditioner</td>
<td></td>
</tr>
<tr>
<td>Smart Meter</td>
<td>Major</td>
<td>PEV</td>
<td>PEV</td>
</tr>
<tr>
<td>Service(Role</td>
<td></td>
<td>Rechargeable</td>
<td></td>
</tr>
<tr>
<td>Pseudonym)</td>
<td></td>
<td>Battery</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>*</td>
<td>Permit</td>
</tr>
</tbody>
</table>

The Policies for smart metering system gateway could be as defined in the following tables:
Table 17. Smart Metering Gateway Policy Set

<table>
<thead>
<tr>
<th>Resource</th>
<th>Role</th>
<th>Action</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Type</td>
<td>Division</td>
</tr>
<tr>
<td>Reload</td>
<td>Grid Operator</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Meter Operator</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Smart Home Service (Role Pseudonym)</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Hibernate/Standby/Restart</td>
<td>Meter Operator</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Grid Operator</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Smart Home Service (Role Pseudonym)</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Turn on/ Turn off</td>
<td>Smart Home Service (Role Pseudonym)</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Meter Operator</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Grid Operator (Major)</td>
<td>Major</td>
<td>PEV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. Don’t forget to enable the policies in the administration panel, by default all policies are disabled.
5. PDP server(s) already is ready.
6. Command to run each WSO2 server is: “[SERVER_HOME]\bin\wso2server.bat”

11.8 Run Demonstration Application

1. Run Apache Tomcat server.
2. Run Identity Server.
3. Run local and remote PDPs.
4. Run Apache Derby Server.
5. Open Web Application and run demonstration, Sign In, Refresh Appliance list, Turn on/Off appliances.