

# Final Year Project II Final Report

## **Integrated Single Sign-On System on Open Nebula**

By

Pornpan Songprasop

Progress Report submitted in partial fulfilment of  
The requirements for the  
Bachelor of Technology (Hons)  
(Information and Communication Technology)

JANUARY 2014

Universiti Teknologi PETRONAS  
Bandar Seri Iskandar  
31750 Tronoh  
Perak Darul Ridzuan

## **CERTIFICATION OF APPROVAL**

Implementation of Single –Sign On system on OpenNebula

By

Pornpan Songprasop

A project dissertation submitted to the  
Information Communication Technology Programme

University Teknologi PETRONAS

In partial fulfillment of the requirement for the

**BACHELOR OF TECHNOLOGY (Hons)**

Information and Communication Technology

Approved by

---

Mr.Izzatdin B Abdul Aziz

Universiti Teknologi PETRONAS

Tronoh, Perak

January 2014

## **CERTIFICATION OF ORIGINALITY**

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and the original work contained herein have not been undertaken or done by unspecified sources or persons.

---

Pornpan Songprasop

## **ABBREVIATION**

IdP – Identity Provider

SP – Service Provider

IdMM – Identity Management Provider

SaaS – Software as a Service

PaaS – Platform as a Service

IaaS – Infrastructure as a Service

SSO – Single Sign-On

UTP – Universiti Teknologi PETRONAS

## **ABSTRACT**

The purpose of this research is to make comparative analysis on the single sign-on systems called Shibboleth and Identity Management Machine in order to choose a preferable system to integrate with a cloud infrastructure, OpenNebula. This paper addresses the criteria that an individual has to look at in order to choose a more suitable single sign-on system for implementation. This study is based on the research methodology started by defining problem statement and then making research and analyse the information. Therefore, it has been done by using research and experiment method. The result of the research was likely impact the experiment phase in this project. The research has been conducted for 27 weeks and the result from the research phase provided valuable information for the next phase of the project which was the experiment phase.

## **ACKNOWLEDGEMENT**

I would like to express my upmost gratitude to the people who have helped and supported me throughout this project.

I would like to express my sincere gratitude to Mr. Izzatdin Bin Abdul Aziz and Mrs. Nazleeni Samiha Binti Haron, my supervisor and my co supervisor respectively, I am very grateful to them for their comments and suggestions for improvements. This project is further improved through their contributions. It would be impossible for me to complete Final Year Project I without their help, cooperation, support, and guidance, understanding and non-stop commitment towards facilitating such a program.

I also would like to appreciate Mr.fazli Anak Jalaluddin for all the help and guidance given to me throughout the time with this project. The supervision and support that was given by him truly helped the progression and smoothness of the project completion.

I would like to extend my appreciation to my family and all people whose name has not been mentioned. They have encouraged me to go through all the hard work and difficult situations I faced. Without all these people, my Final Year Project I would not have been a memorable and knowledgeable journey.

## **EXECUTIVE SUMMARY**

This paper presents the workflow and methodology proposed to study a way to integrate OpenNebula with a single sign-on application.

The initial workflow aims to gain high quality data for this study, which includes architecture of OpenNebula and single sign-on application.

OpenNebula is a web application that provide cloud infrastructure; and single sign-on tool is a tool that allows user to sign in to multiple services at once.

The report is divided into 5 sections:

Section 1 is the problem statement and objectives. It presents problem statement of this project followed by the main objectives. The scope of the study is also discussed in this section.

Section 2 presents literature review of the project.

Section 3 discuss the methodology that is used in the project; the approach and procedures to complete this project. It also explains the equipment that will be used as tools for the study and also milestones and Gantt chart of this project.

Section 4 explains result and discussion of the project so far.

Section 5 summarizes the whole progress of this project.

## Table of Contents

ABBREVIATION .....	IV
ABSTRACT .....	V
ACKNOWLEDGEMENT .....	VI
EXECUTIVE SUMMARY .....	VII
List of Figure .....	X
List of Table .....	XI
CHAPTER 1 .....	1
1.1 BACKGROUND OF STUDY .....	1
1.2 PROBLEM STATEMENT .....	2
1.3 OBJECTIVES.....	3
1.4 SCOPE OF STUDY.....	3
CHAPTER 2 .....	5
2.1 LITERATURE REVIEW .....	5
2.1.1 Cloud computing .....	5
2.1.2 Challenges and issues in cloud computing .....	6
2.1.3 Recommendation .....	8
2.1.4 Functional Requirement in SSO systems .....	8
2.1.5 Shibboleth .....	12
2.1.6 Identity Management Machine (IdMM) .....	16
2.1.7 OpenNebula.....	18
CHAPTER 3 .....	21
3.1 METHODOLOGY .....	21
3.2 PROJECT ACTIVITIES.....	22
3.2 REQUIRED TOOLS .....	22
3.4 KEY MILESTONES .....	23
3.5 GANTT CHART .....	25
CHAPTER 4 .....	26
4.1 RESULT AND DISCUSSION.....	26
4.1.1 Overview of SSO systems .....	26
4.1.2 Comparative Study.....	28



4.1.3	Architecture of OpenNebula.....	31
4.1.4	Expected integrated proposed SSO system on OpenNebula.....	33
4.1.5	Step to deploy IdMM.....	34
4.1.6	Problems and Solutions.....	44
4.2	EXPERIMENTAL SETUP.....	50
4.2.1	Data.....	50
4.2.2	Platform.....	51
4.2.3	Software.....	51
4.2.4	Steps for testing.....	51
4.2.5	Error rate.....	52
4.3	SYSTEM TESTING.....	53
4.3.1	Install ability Testing.....	53
4.3.2	Security Testing (Confidential).....	54
4.3.3	Black box Testing.....	56
4.4	TEST RESULT.....	57
CHAPTER 5.....		59
5.1	Conclusion.....	59
Appendices.....		60
Appendice 1.1: Apache DS - Server.xml.....		60
Appendice 1.2: Tomcat - server.xml.....		64
Appendice 1.3: Apache Directory Studio – Services.Idif.....		66
Appendice 1.4: Apache Directory Studio – Authentications.Idif.....		67
Appendice 1.5: Apache Directory Studio – exampleUser.Idif.....		68
Appendice 1.6: Apache Directory Studio – exampleUser.Idif.....		70
References.....		71

## List of Figure

Figure 1 Shibboleth interface .....	12
Figure 2 IdP's architecture [20].....	14
Figure 3 SP's architecture [20].....	14
Figure 4 IdMM interface .....	16
Figure 5 Architecture's IdMM [25].....	17
Figure 6 OpenNebula interface.....	18
Figure 7 Virtual Machine with OpenNebula.....	19
Figure 8 Method using in conducting and complete project .....	21
Figure 9 Overview of SSO systems .....	26
Figure 10 A sequence diagram's how SSO system works.....	27
Figure 11 OpenNebula's Architecture .....	31
Figure 12 SSO system integrated with OpenNebula .....	33
Figure 13 OpenNebula integration with IdMM.....	34
Figure 14 Apacheds.sh's result screen. ....	37
Figure 15 Apache Directory Studio's result screen. ....	40
Figure 16 apt-get update fail to fetch.....	44
Figure 17 ERR_171 Failed to bind an LDAP service to the service registry .....	45
Figure 18 LDAP: SSL handshake failed .....	46
Figure 19 Could not determine the service's fully qualified domain name .....	47
Figure 20 LDAP: error code 32 – No such object .....	48
Figure 21 SEC error bad database .....	49
Figure 22 SEC error IO – Could not authenticate to taken NSS Certificate.....	50
Figure 23 How does libnss3-tools should look like.....	52
Figure 24 Summary of system testing .....	58

## List of Table

Table 1 Functional requirements for SSO systems.....	9
Table 2 Comparative study of Shibboleth and IdMM .....	29
Table 3 Summary system testing .....	57

# CHAPTER 1

## INTRODUCTION

### 1.1 BACKGROUND OF STUDY

In today's world, cloud computing has been increasingly used as an important tool and is vital for leading a business towards success, especially in medium and small sized companies. "Cloud computing" refers to a combination of many components in the Internet which are organized with the aim of carrying out many requests concurrently [3].

Cloud computing provides its computations and various kinds of resources in a way that is called "services" through the Internet which can be either hardware or software. It allows clients, which are in this context are the companies, especially medium and small sized companies, to now establish their business without having to worry about system building, database, applications among others, which are designed to support their ideas. Such advantages have led to the increasing popularity of cloud computing usage.

Cloud computing services are, therefore, being used in everyone's life without the person even being aware of it. Examples of cloud computing services are Facebook, Twitter, Gmail and YouTube. Social media is also one type of service in cloud computing and it includes site such as Facebook and Twitter. [13] discussed that more than 40% of people using social media service is a member of more than one social media networking site. [4] also mentioned that more than half of them who are active users of Facebook, Twitter, and Gmail log in daily. That means they would have to sign in into three different systems every day and every time they want to use it. To keep the account information more secure, people tend to have different username, password or other credentials for each service. The question brought into attention is

whether there are new technologies that could help to make the way we log in to the service better?

Single sign-on (SSO) system is a tool that allows the users to sign in to multiple services at once. There are many single sign-on systems available. Each of them is different in term of requirements that give them different pros and cons. Therefore, it is important to study each of it to be able to define which SSO system is the most suitable to use in different situation.

Therefore, this paper presents a feasible workflow on how to port and deploy existing single sign-on system on OpenNebula.

## **1.2 PROBLEM STATEMENT**

Although the current cloud computing industry has been dramatically developed into a successful one, a number of challenges and issues are still being faced by most of the users.

In this report, we address the non-technical issues of clouds faced by many cloud users.

### *Non-technical aspect.*

From the view of non-technical issue, the main challenge is faced due to the fact that the number of sites, which require users to apply an account, has been on the rise. Users tend to have low quality security; this is true especially in users who are using many services. [5] and [22] mentioned that there are some users that decide to use passwords that are easy to remember and they also tend to write down their usernames and passwords for every login account they have or reuse same username or password for every account.

Furthermore, today as employee, partner and customer increasingly dependent on cloud application to conduct business. Many organization have many applications that require username and passwords which normally differ in different applications. Most of the big companies tend to have more than 10 applications requiring username and password [11]. The program forces the

users to change their password once a month and log in and log out every time they want to use the programs. This is clearly a waste of time for users.

In addition, these users tend to use weak passwords to access dozens of applications. Due to the proliferation of non-standardized cloud identity, many of the passwords are forgotten, lost and easy to steal because people tend to write down their usernames and passwords in papers. This is the first reason why we need a SSO system to be implemented in.

However, as the author has mentioned before in background of study, there are a lot of SSO protocols and systems available on the internet. All SSO systems are different even though they are providing the same main functionality. Since it is different from one to another, it has its own pros and cons. So in order to choose the most suitable single sign-on for the particular cloud infrastructure, the developer has to look very close to that SSO system. Selecting the incompatible SSO protocol will affect the system.

Therefore, this paper is devoted to address primarily the problem associated with how to choose the more preferable one for a given environment, system or organization.

### **1.3 OBJECTIVES**

The objectives of this project are as follows

1. To carry out a comparative study on 2 SSO systems; Shibboleth and Identity Management Machine (IdMM).
2. To carry out study on OpenNebula
3. To carry out study on how the proposed SSO system can be integrated with OpenNebula.
4. To implement the proposed SSO system on OpenNebula.

### **1.4 SCOPE OF STUDY**

The scope of the study includes the experimental work in studying OpenNebula and 2 SSO systems which are Shibboleth and IdMM. The testing service was a social media named as Twitter. This project is devoted to find a way to integrate the proposed SSO with OpenNebula, and investigate its performance activities through the use of experiments and simulations.

# **CHAPTER 2**

## **LITERATURE REVIEW**

### **2.1 LITERATURE REVIEW**

#### **2.1.1 Cloud computing**

Cloud computing is a term which does not have a formal definition. In [3], the author mentioned that if you ask ten different professionals what cloud computing is, you'll get ten different answers! There is no one exact sentence that can accurately describe cloud computing. Many people have different opinions of it. A researcher said that cloud computing is a construction that allows you to access applications that actually reside at a location other than your computer [3]. While [6] mentioned that cloud computing is a pay-per-use model, which enable user to use minimal effort to manage them. In [11], the author stated that it is an information processing, delivery and storage's model which provide user physical resources to client based on his/her demand. He also mentioned that cloud computing can also be defined as "management of resources, applications and information as service over the internet on demand." Another definition of cloud computing discussed by [9] is that it is an array of IT service groups provided in a network that have ability to scale up or down their service requirement.

Although the definitions are varied, [3], [6], [9], [11], [19] have the same common idea on type of cloud computing service models. They strongly agree that it can divided into three types of models consists of software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS). [3], [6], [9] and [11] mentioned that SaaS allows customer or service's users to access service that is hosted on a network. Gmail, Yahoo and Facebook are examples of an application in this service model. While SaaS provide users applications, PaaS provides all required resource and tools to develop one. Some popular software of a PaaS includes Tomcat, MySQL and Oracle. Unlike SaaS and PaaS, IaaS offer hardware as a service. This type of cloud computing



service model offers users with the authority to control over operating system, storage and deploy application [11]. IaaS approach is also used to cater to more freedom than SaaS and PaaS. More recently, CloudStack, OpenNebula, Openstack have become more popular applications of IaaS.

It is clear that cloud computing had definitely played a role in the organizations. However, there are many issues and challenges that are still being faced in cloud computing. Next section will discuss on the main issues and challenges as well as discuss some solutions.

### **2.1.2 Challenges and issues in cloud computing**

As described earlier, the project is focused on 2 SSO systems, Shibboleth and IdMM, and OpenNebula. Therefore, this document will highlight some of the issues that impedes the acceptance of users towards cloud computing. There are several issues related to managing cloud computing. Many researchers identified and analysed all the factors that are causing problems and its consequences as well as rank them according to the level of consequence. [7], [9], [11], [17] and [19] remarks that there are a number of challenges in cloud computing but the most notable obstacle in shifting towards to cloud computing model is security issues. The problem that is being faced can be viewed as having two frames as there are too many cloud services that require usernames and passwords and how to choose a suitable SSO.

From the view of an end user in cloud service, Cloud services play a key role in helping organizations succeed. The numbers of cloud service being used continue to grow as well as the need of logins and passwords of different sites continue to increase. However, there are difficulties faced in retaining passwords as secrets. Even though strong authentication, passwords, and other techniques are being implemented in cloud services, a research found that 44% of survey respondents admitted to dealing with the dozen of accounts they wrote down their password's accounts on a note sheet and 37% reveal that they share their password among their friends [8]. This indicates that people do not have much awareness on the negative impact of discovering their password

that should be hidden. Furthermore, many organizations have many applications that require username and passwords which normally differ in different application. In addition, many applications also require users to change their password in a set period. The period is vary from organization to organization and application to application. It may be a month or half year or a year. I had a chance to interview my supervisor, Mr. Izzatdin whom has got experience about this matter. He stated that he has to renew his passwords for applications every month as recommendation from his company. It should not be a problem if the number of the application are just 2 to 3 programs and used often. However, it is often not the case. Most of the companies has more than 15 applications and not all of the applications are used monthly; such as student grading system that is normally used once in 4 to 6 months. To renew passwords for 15 applications, it often take time and require a lot of idea to create new qualified passwords. Too often, the ideas will be ran out and end up repeating old passwords or easy ones. Besides that, it is very hard to remember which password is used for each application.

However, the issues are not unsolvable one. [8], [10], [17] are strongly on the same suggesting note that SSO systems as the solutions for addressing the above challenges and issues. It is a tool to allow the user to sign in to multiple service at once. The main purpose of SSO system is to enable user to access to several websites by only log in into SSO system. It is not only reduce the chances for users forgetting their password but it also reduces the number of call to IT help desk about losing password [16]. Furthermore, it also decreases the possibility of phishing to happen [10] [16]. This is because the central SSO system gets user credential directly. It make the credential unable to be cached by the actual service that user is trying to access [23].

[26] mentioned that there are several free and commercial SSO systems available. People often use open source software because they are free. However, Opens source tools are developed, managed and, maintain by volunteers. They often run on limited platforms and may not be well supported. Different tool are also often recommended for different purpose of use.

Different users often have different ways of doing things and things that should be done.

### **2.1.3 Recommendation**

As mentioned above, different tools are also often recommended for different purpose of use. Different users often have different ways of doing things and things that should be done. It is very crucial to choose the right tools to assist them to meet their unique needs. It is a good practice for everyone to understand each of the software before determine which SSO systems will work best in their organizations. The most suitable SSO system does not have to be the latest version that has been released rather should be the one that is able to cater to most of the users' requirement needs. Many of the theories and concepts are not difficult to understand. What is difficult is implementing them in various environments. Developer must consider functionalities and many different issues when integrating SSO with the environment. Just as each SSO system is unique, so is its environment and platform. To be able to make a decision for choosing SSO system, developers need to understand the components involved in SSO system and environments. [21] stated the way to do that is as the following

- 1<sup>st</sup>: Comparative study on the functionality of SSO systems
- 2<sup>nd</sup>: Proposed the preferred SSO system
- 3<sup>rd</sup>: Implement the proposed SSO system on cloud infrastructure

### **2.1.4 Functional Requirement in SSO systems**

There are many different ways to implement SSO system. All products work with various different platforms. [21] suggested that a list of the functional requirements of an ideal security SSO system on the mark as shown in table 1.

Table 1 Functional requirements for SSO systems

Functional Requirements	Description
<b>1. Single Point of Administration</b>	Any system with this functional requirement will only enable administration task to be done from only one point.
<b>2. Administration for Multiple Platforms</b>	Any system with this functional requirement enable users, administrator, to use the product in any platform that the product can be implemented on.
<b>3. Common Control Language</b>	Any system with this functional requirement enable developer to work on the product with only one language without considering the type of platform it being implemented on.
<b>4. Auto Revoke after a Number of Attempts</b>	Any system with this functional requirement include security authentication. It will revoke the user's account after a few log in attempts using the wrong username and/or password.
<b>5. Customize in Real-Time</b>	Any system with this functional requirement enable user to customize the product without the need of initializing the product. This is to reassure that the product is available 24 hours a day, 7 days a week.
<b>6. Release Independent/ Backward Compatible</b>	Any system with this functional requirement ensure that the new version or new release of the product can be integrated with the old version or able to work independent. Without this requirement, the product has to be uninstalled and install every time a new version release.

Table 1: Functional requirements for SSO system

Functional Requirements	Description
7. GUI Interface	Any system with this functional requirement provide user an interface to interact with the system. The interface may look different when it runs on different platform.
8. Ability to interface with Application and Database	Any system with this functional requirement enable SSO system to integrate with the existing applications and databases.
9. User Defined Fields	Any system with this functional requirement enable user to define extra fields that is not available in the system. The extra field may vary from organization to organization.
10. Support Password Rules	Any system with this functional requirement implemented common password rules. The rules include password length, aging, syntax rule, and other customer-defined rule as limited amount of access time after the password is expired and uniqueness of the passwords.
11. Flexible Cost	Any system with this functional requirement indicate that the product's price is reasonable.
12. No clear Text Passwords	Any system with this functional requirement will ensure the passwords to be encrypted before passing to the network or database. This is to reassure the integrity of all passwords.
13. One Single Product	Any system with this functional requirement enable user to get product by one time installation.

Table 1: Functional requirements for SSO system

Functional Requirements	Description
14. Insure Loginid Uniqueness	Any system with this functional requirement aware the uniqueness of all loginid. Different users must have different loginid. This is preventing any loginid to be used by more than one person.
15. Encryption should be commercial	Any system with this functional requirement normally use standard encryption to encrypt passwords. The purpose of this requirement is to reassure the way the password is encrypted is secure.
16. Integrity of Security Database	Any system with this functional requirement prevent any changes to take place with the confidential information that reside in database. The change cannot be done from other places than the product itself.
17. Inactive User Time-out	Any system with this functional requirement sign off any inactive user automatically after a specified period.

These are the functionality that should be included in SSO system. Much of these functionalities include basis information such as the level of security in the system, level of open source and so on. These are the least amount of information that the developer should consider before making a decision on which program is going to be implemented since the system are varied for different needs and expectations in term of platform and deployment environment.

This document focuses on application SSO system. Users can only access to the applications only after they log on to single sign-on. It will be implemented

on Open Nebula in this project. This system does not only guarantee a high level of security against hacker, but also make the password to remain confidential.

### 2.1.5 Shibboleth

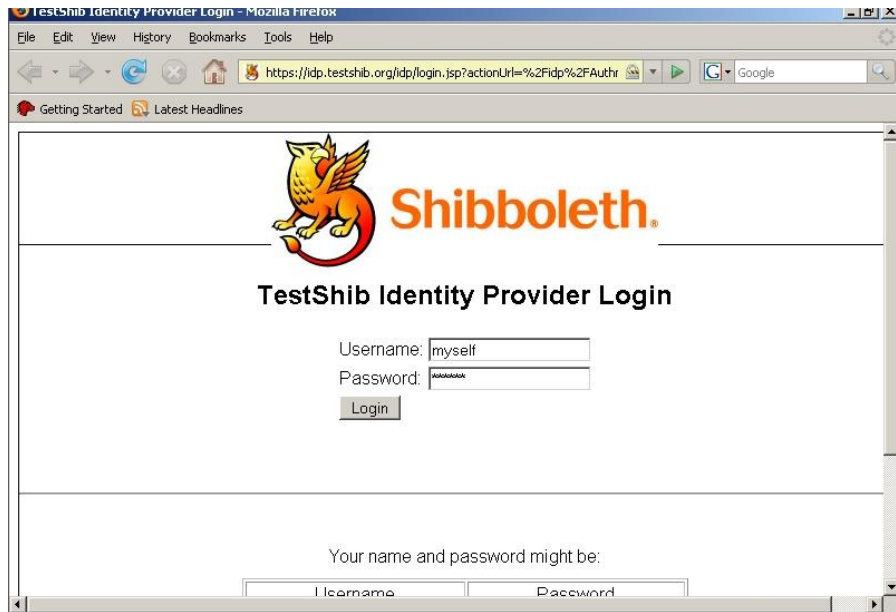


Figure 1 Shibboleth interface

Shibboleth is one of the most popular SSO systems that has been used nowadays. It is used by a lot of universities and companies such as Harvard University, Cornell University and Amazon. Shibboleth is an open source software released under The Apache Software License [9]. Figure 1 illustrates how Shibboleth interface look like.

It is implemented with Security Assertion Markup Language (SAML). [29] discussed that SAML is secure XML base communication mechanism that share identity between multiple organizations and applications. But SAML cutting edge in the cloud system is in its ability to eliminate most passwords and enable single sign on. SSO system with SAML give faster, easier, and trusted access to application without storing password or requiring user to login to each application individually.

Instead of password, application that use SAML accept a secure token which only review what is needed to get access to the application. Since no password exists, there is nothing for employees or customers to forget, lose or have password stolen.

Shibboleth services can be divided into two parts based on its functionality; IdP and SP. IdP responsibility is in sending the user's information that is requested by the service provider regarding the user's attempt in logging in. It can be simply implied that IdP normally uses by for cloud service user and SP normally uses by cloud service provider. SP and IdP come in different package for the installation. At the moment of writing, the latest version of IdP is v. 2.4.0 and SP is v2.5.3. [18]

The architecture for both of them are provided in the figure 2 and figure 3:



- Shibboleth's architecture

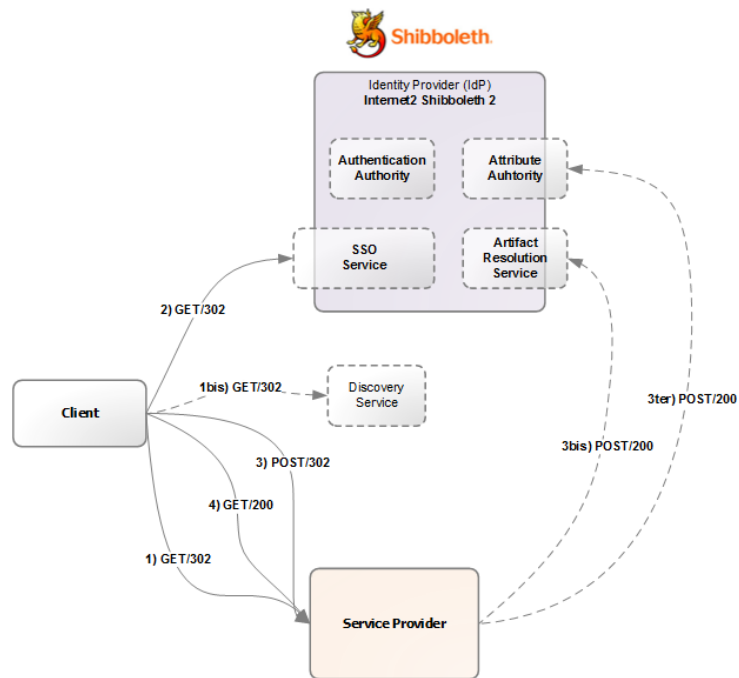


Figure 2 IdP's architecture [20]

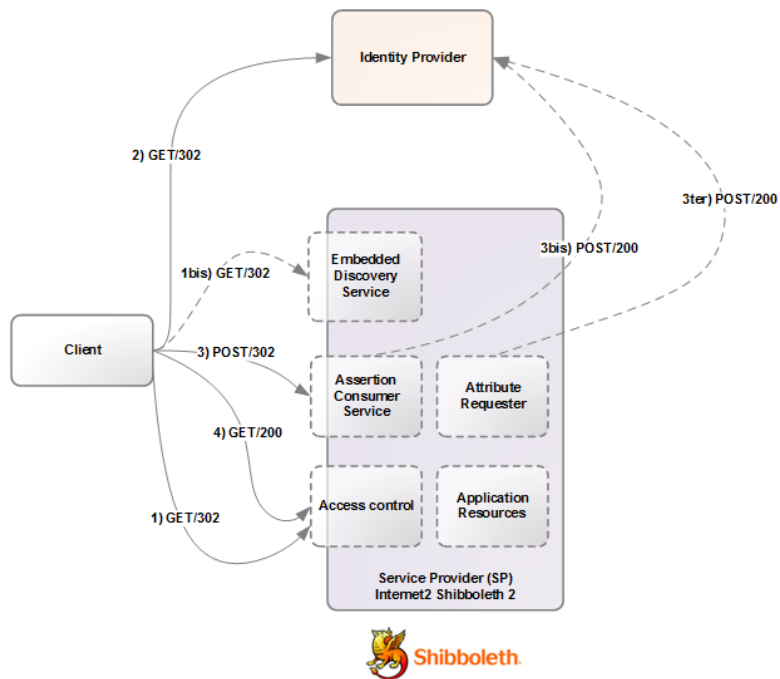


Figure 3 SP's architecture [20]

From the view of IdP's architecture, figure 2, the component in IdP can be divided into 4 elements [20]:

1. **Authentication Authority:** The system will not be able to match the user name identifier and his/her attributes without this element.
2. **SSO service:** this is the element that will be responsible for generating user's information for authenticated process to a given service.
3. **Attribute Authority:** This element responsible is retrieving the information from data sources. Attribute authority is a core of IdP discovery as it enable Shibboleth to be able to control many data repository at once.
4. **Artifact Resolution Service:** This element acts like a translator between a user artifact and authentication assertion.

From the view of SP's architecture, figure 3, the component in SP can be divided into 3 elements [20]:

1. **Access Control:** This element determines whether the request from user to use the service is authorized or not. It either grant the users to use their privacy area in the service or denying the access.
2. **Assertion Consumer Service:** Unauthorized user will be prompted from service for information to authenticate. This component will direct the user who has installed SSO system to his/her IdP.
3. **Attribute Requester:** All of the information that is sent by IdP or user will be collected by this element. When this component works, SP and IdP can communicate directly without going through user agent during the process.

GET 302/ POST 302 is a HTTP response status code that indicate that there is a need for more actions to achieve the request. This code is the most popular using on changing from one web page to another webpage. The process will be done without any action require from users if the second request is GET. [30]

GET 200/POST 200 is also a HTTP response status code. It represents that the request has been accepted and approved by the receiver. The difference between GET 200 and POST 200 is the content in the message. Context in

GET 200 will relate with information of the requested resource. The resource or the result of the request will be the message in POST. [30]

There are a few reasons why Shibboleth has been chosen to be one of the two SSO systems, which involve in the comparative study in this project. First, It is because Shibboleth is one of two SSO systems that role as IdP discovery. [28] IdP discovery simply mean that SP support more than one choice of IdP. Most of the methods involve in IdP discovery involve asking user to identify the IdP for each service directly. Another SSO system that can act as IdP discovery is ZXID. ZXID is not in the list in this project because it is based on C programming language which author does not familiar. [28] The second reason, therefore, due to the familiar level of programming language. Shibboleth uses java-programming language as the core of the system, which author experience with.

### 2.1.6 Identity Management Machine (IdMM)

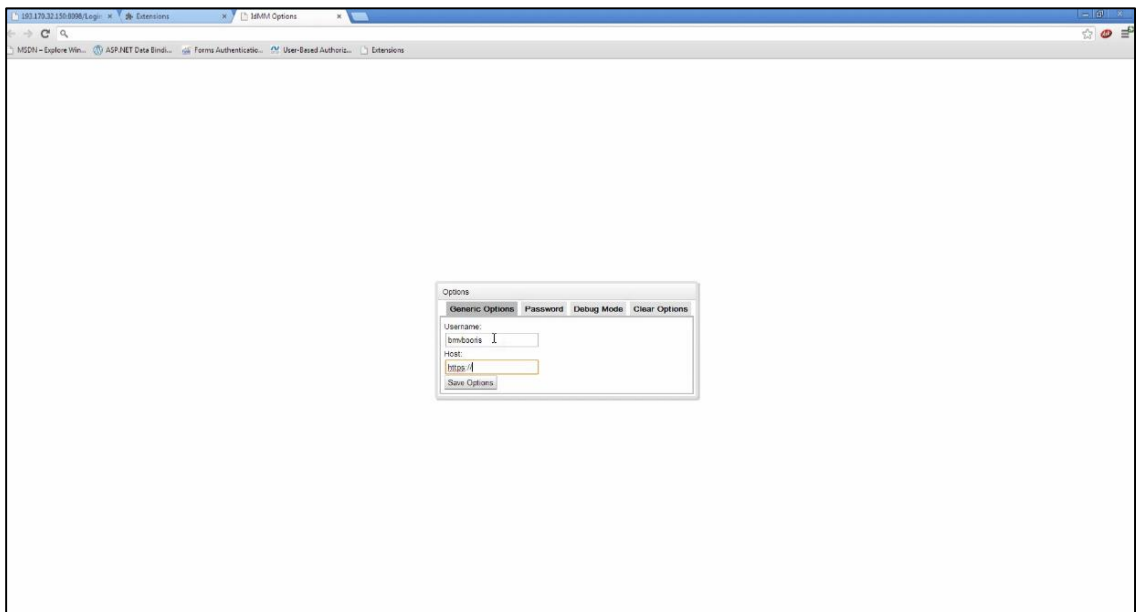


Figure 4 IdMM interface

IdMM is another open source SSO system that is included in this project. Unlike Shibboleth, IdMM can only serve the user as IdP. The system can only automatically sign-in a user to a particular cloud service. IdMM, therefore, can

be only implemented in client side. Mircea BorisVieju, a researcher who works for Client-Centric Cloud Computing, has developed it. [24] Figure 4 illustrates how IdMM interface look like.

Like Shibboleth, [26] remarked that it is also implemented with Security Assertion Markup Language (SAML). As system architecture differs from system to system and perspective to perspective, IdMM's architecture also different from Shibboleth. Figure 5 shows the IdMM's overall architecture.

- **IdMM's architecture**

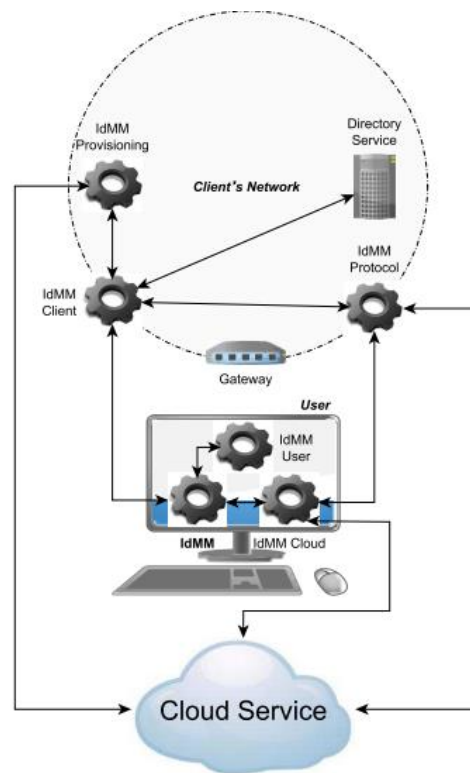


Figure 5 Architecture's IdMM [25]

[25] explained that the core of this system is IdMM. It includes the entire main functionalities to achieve the goal of the system, which is to serve as SSO system. It includes the functionality that allows this SSO system to provide automatic authentication. IdMMuser controls the interaction between IdMM and user, IdMMcloud handles the interaction between cloud service and IdMM. IdMMclient focusses on managing any interaction that take place between client's directory service and the IdMM. The IdMMprovisioning

responsible for any actions regard to de-provisioning and provisioning. Finally, IdMMprotocol decide whether the given cloud service implement any protocol in the authentication process or not.

In this architecture, they assume that the cloud system and client directory have needed information for authentication.

There are also few reasons why IdMM has been chosen as another SSO system, which involve in the comparative study in this project. At the moment of writing, IdMM is the only SSO system proposed as a chrome extension using Java script language that focuses on client-cloud orientation. This simply decreases the possibility of phishing to happen, as it is harder for user to log in via a fake SSO website since user is prompted for his/her credential via chrome extension page. Another reason is that it's Client – Based identity directory. This system allows a client to keep a private directory service to enhance the privacy of user confidential information.

### 2.1.7 OpenNebula



Figure 6 OpenNebula interface

OpenNebula is used as a platform aimed at providing private cloud. [1] discussed that it also offer cloud interface for disclosing its functionality for virtual machine, network management and storage. The figure 6 above shows the OpenNebula interface. A virtual machine is a software computer that provides user run operating system and application like a physical computer [3]. Hybrid cloud is also one of the cloud deployment models that OpenNebula supports. It is used to combine public cloud-based infrastructure with local infrastructure. Figure 7 represent how the virtual machine should look like in OpenNebula.

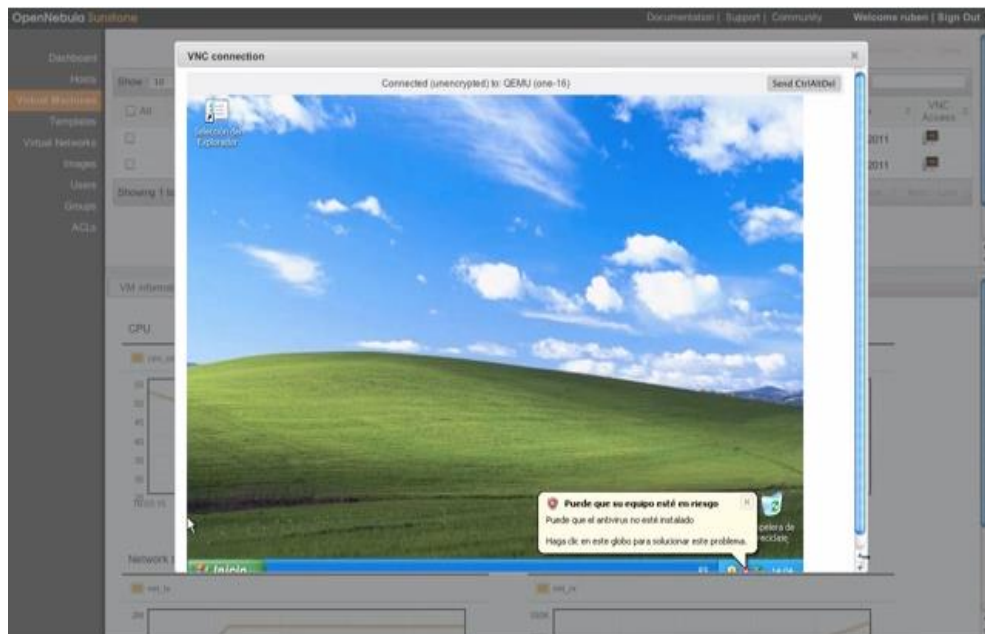


Figure 7 Virtual Machine with OpenNebula

Just as all single sign-on software are different, so does IaaS applications. It is important to take time to understand and review each software to meet the needs of the project environment. The type of work and needed parameters in projects determine which application to be used. Due to many reasons, OpenNebula has been chosen as a framework to proof the concept in this project. [12] and [14] discuss that it is simple yet flexible and rich in users

needed functionality. Simple here refers to easy to use, deploy, update, install and download. The users would not need much time to understand how to use it. It is also focusing more on the Data Centre virtualization which is possible to fit into any datacentre. Furthermore, another big benefit that [12], [14], [15] and [27] remarked that it is fully open-sourced. Users are fully authorized to customize every part of their cloud. This also means that user can download its code to edit or compromise it. The third reason of choosing OpenNebula as proposed model is that it is delivered as a packaged product. [12], [14] and [27] mentioned that user can get every necessary key functionalities with a single install. This principle of OpenNebula make it more reassuring in term of the long term scalability and effective performance through one-stop support, update and integrated patching processing [12]. Furthermore, OpenNebula is also mature and proven. It has been available over the Internet for more than 7 years, which normally indicate that its users and its community have tested it. Finally, it is extensible. OpenNebula achieved the feature because users can integrate which existing hypervisor or new data centre. These are all the reasons why this project has been chosen to use OpenNebula to be hosted of proposed SSO system. [27]

# CHAPTER 3

## METHODOLOGY

### 3.1 METHODOLOGY

The research methodology selected for the study, integrated SSO system on OpenNebula is shown in the figure 8 below. To better understand this approach, one can refer to diagram and explanation below:

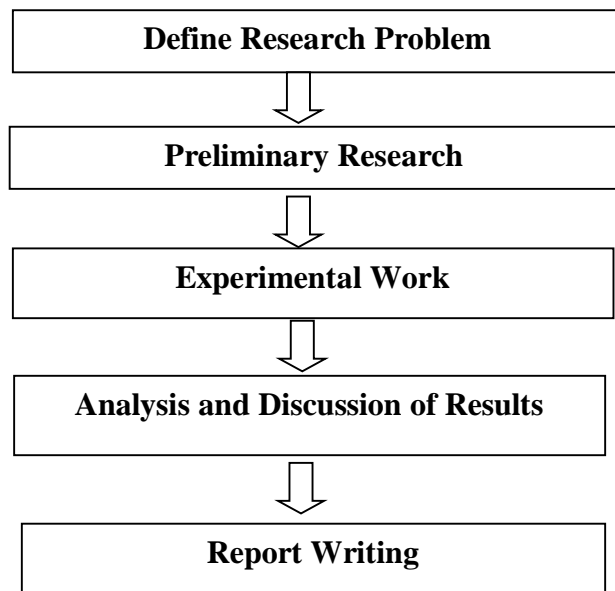


Figure 8 Method using in conducting and complete project

Following the structure presented above, the first action that has been done is defining the occurred problem which is losing sight of quality or customer satisfaction. Then, an initial and high level analysis and design of two current SSO systems; Shibboleth and IdMM, OpenNebula, and cloud computing were conducted by studying research papers and carrying out critical analysis. The comparative study of Shibboleth and IdMM were also be conducted to find the difference of their performance to identify the most preferable SSO system. Integrating it with OpenNebula is also an important step as the project involves experimental work by which the chosen SSO system performance was analysed. After that, discussion on the result was obtained with



regards to theories and objective of the project works. At the last step of the project, compilation of all research findings, literature reviews, and experimental results into a final report was conducted. Despite being time consuming, this approach can allow the project to better leverage the current systems. This process followed a parallel strategy with duration of approximately 28 weeks.

### **3.2 PROJECT ACTIVITIES**

The following activities have been set in order to achieve all the project objectives:

1. To perform a comparative study on two open source SSO systems.
2. Based on the result of comparative study; we proposed the preferable SSO system to be implemented.
3. To study architecture of OpenNebula in order to integrate with the proposed SSO system.
4. Perform the implementation of the SSO system by configuring on OpenNebula.
5. To perform validation which is test and experiment.

The first and second activities are done to achieve the first objective of the project. The second and the third activities are done to achieve the second and third objective. The fourth objective are done to achieve the fourth objective and finally, the last objective is done to verify that the proposed SSO system can be worked on OpenNebula.

### **3.2 REQUIRED TOOLS**

The lists of the required tools are listed below:

#### Tools for cloud infrastructure

- OpenNebula (a particular cloud infrastructure)

OpenNebula is a web application for providing cloud. It also provides virtual machine, which is going to be a place where the entire SSO system be integrated.

#### Tools for open source single sign-on programs

- Shibboleth
- Identity Management Machine (IdMM)

The list above is all open source SSO systems that were analysed in comparative study to propose one of them to be integrated on OpenNebula.

#### Tools for IdMM installation

- OpenNebula
- Ubuntu 12.04
- Window 8
- ApacheDS 2.0.0-M15
- Apache Directory Studio 2.0.0.v20130628
- Apache tomcat or tomcat server
- IdMM package
- Chrome
- Java runtime environment

### **3.4 KEY MILESTONES**

#### **Training Schedule**

The following represent key project milestones, with estimated completion dates:

<b>Milestone</b>	<b>Estimated Completion Date</b>
<b>I:</b> Understand cloud computing, OpenNebula and single sign-on	23/10/2013
<b>II:</b> Comparative 2 SSO systems.....	17/11/2014
<b>III:</b> Study architecture of OpenNebula.....	3/01/2014
<b>IV:</b> Integrate SSO system on OpenNebula.....	11/02/2014
<b>V:</b> Get validation on testing system (complete experimental work)	13/03/2014
<b>VI:</b> Achieved document.....	18/04/2014

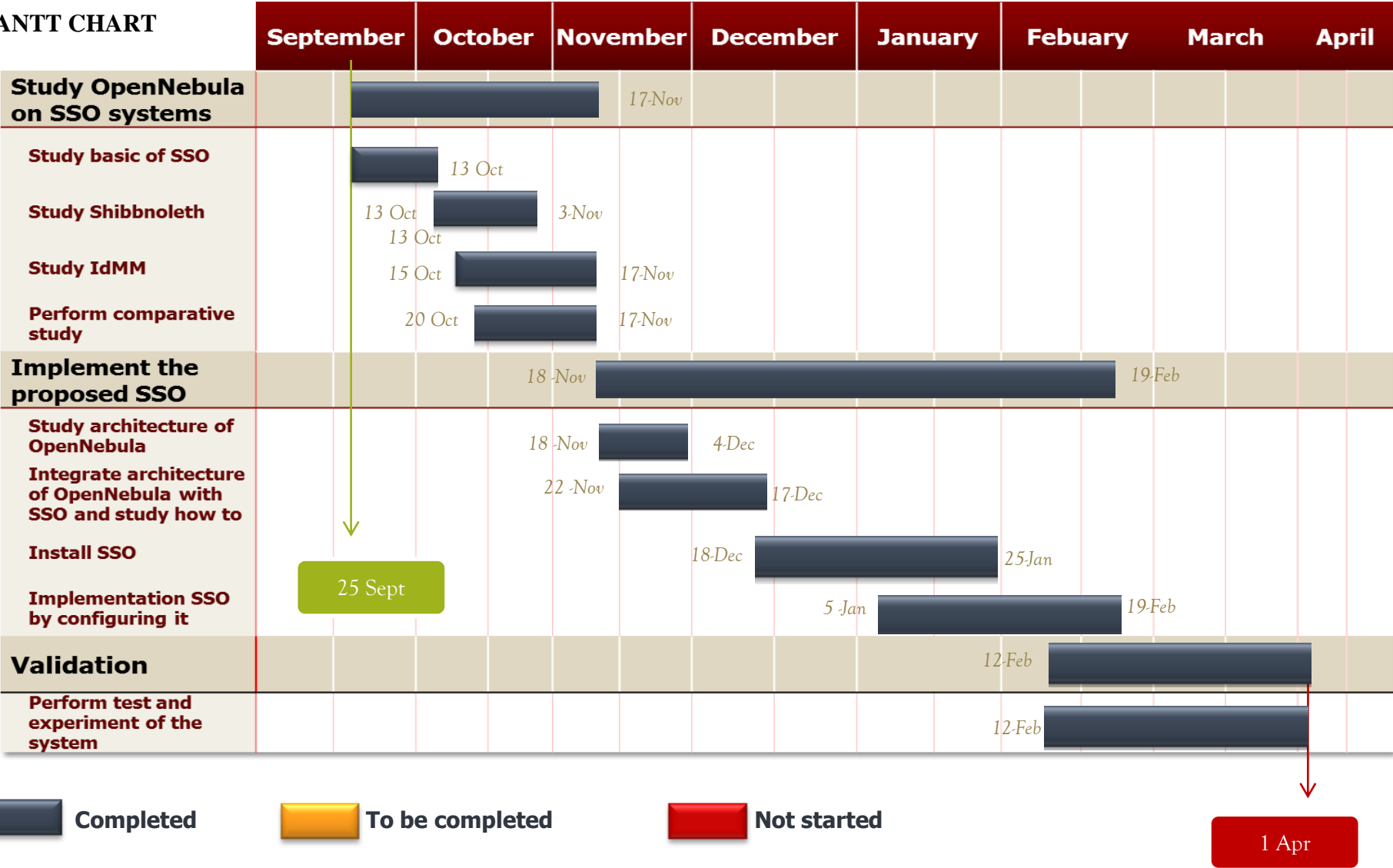
**VII: Gain acceptance**.....18/04/2014

**University Technology PETRONAS Assessments Flow**

The following represent key UTP's assignment milestones, with estimated completion dates:

<b>Milestone</b>	<b>Estimated Completion Date</b>
<b>I: Submission of Extended Proposal</b> .....	30/10/2013
<b>II: Submission of Interim Draft Report</b> .....	18/12/2013
<b>III: Submission of Interim Report</b> .....	25/12/2013
<b>IV: Submission of Progress Report</b> .....	26/02/2014
<b>V: Pre-SEDEX</b> .....	19/03/2014
<b>VI: Submission of Draft Report</b> .....	26/03/2014
<b>VII: Submission of Dissertation (soft bound)</b> .....	2/04/2014
<b>VIII: Submission of Technical Paper</b> .....	2/04/2014
<b>IX: Oral Presentation</b> .....	9/04/2013
<b>X: Submission of Project Dissertation (Hard Bound)</b> .....	23/04/2013

3.5 GANTT CHART



# CHAPTER 4

## RESULT AND DISCUSSION

### 4.1 RESULT AND DISCUSSION

The information has been retrieved from previously research work of the result chapter. In addition, the website that possess the system was taken as a reference furthermore, one of the systems, IdMM was installed and simulated.

#### 4.1.1 Overview of SSO systems

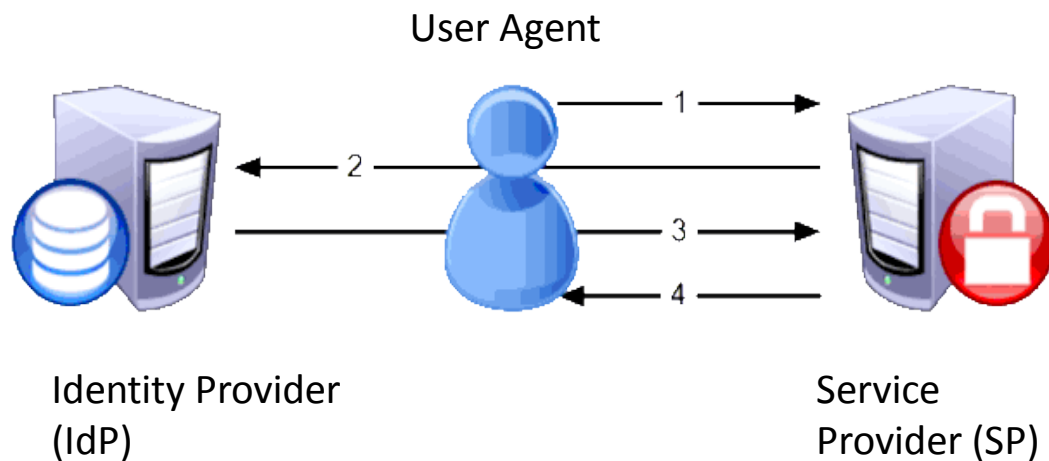


Figure 9 Overview of SSO systems

For SSO systems including Shibboleth and IdMM to work, there are 3 identities involves as shown in figure 9, first is identity provider. An identity provider maintain a directory of user and authentication mechanism. Second is the service provider. Service provider runs a target website, application or service. Identity and service provider maybe in separate organization such as when an employee access an external cloud application like Gmail. The third identity is the user agent who has a known account with identity provider. SSO simplifies the relationship between these entities and strengthen the security of the interaction. A user sign in to company network with their credential when

they click a link to access application or secure content at service provider application. The identity provider generates a SAML token to a service provider. The token grant the access to the application and content but it does not pass any information that can be used by anyone else to access them. Apparently, by architecting and deploying single sign-on solution, the security is increased by eliminating multiple weak passwords for each application. The figure 10 below illustrate how SSO work in sequence diagram.

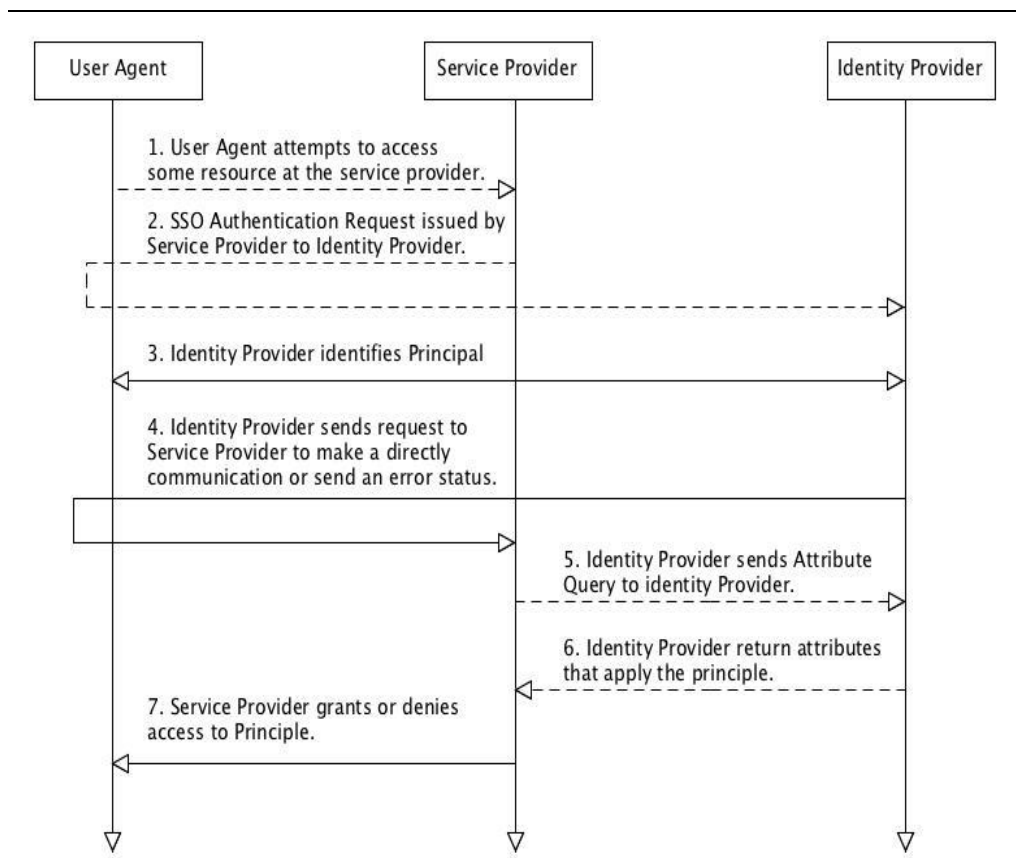


Figure 10 A sequence diagram's how SSO system works

1. User send HTTP request to Service Provider which is considered as cloud service.
2. Service Provider request an authentication information form the user. It will be direct to Identity Provider.

3. The identity provider then identify whether the section will be performed by any protocol or not. It is probably that a new authentication request might take place.
4. In this step, the identity provider will be communicate directly with service provider. The purpose of this transaction is to pass the respond to the service provider. The respond is either an error message or valid assertion message.
5. Service provider then state the attribute or any credential or non-credential information from identity provider.
6. The identity provider send all the requested information to service provider.
7. Once it retrieve all the specified attribute. The authentication process will then begin. User will acknowledge the result of the process from service provider's respond. It will send wither an error message or grant the user to use their confidential area.

#### **4.1.2 Comparative Study**

The functional requirements in this comparative study are taken from [] in literature review part. The comparative study result from Shibboleth and IdMM can be seen from table 2.

Table 2 Comparative study of Shibboleth and IdMM

Functional Requirements	Shibboleth	Identity Management Machine( IdMM)
Single Point of Administration	Yes	Yes
Administration for Multiple Platforms	Yes – It can be on Unix system or Windows system	Yes– It can be on Unix system or Windows system
Common Control Language	Yes – The system is based on Java programming language	Yes - The system is based on JavaScript programming language
Auto Revoke after a Number of Attempts	Yes	Yes
Customize in Real-Time	Yes	Yes
Release Independent/ Backward Compatible	Yes – The latest version is Shibboleth 2.0 which is fully backward compatible with Shibboleth 1.3.	Maybe – There is only one available version of IdMM on Internet. It is very hard to determine that the new version will possess this requirement or not.
GUI Interface	Provides user interface for users but not for developers.	Provides user interface for both users and developers.
Ability to interface with Application and Database	Yes	Yes



Functional Requirements	Shibboleth	Identity Management Machine (IdMM)
Support Password Rules	High level security	Medium level security
Flexible Cost	Fully open source – It provides product, its architecture and code for downloading	Open source - It provides product and its architecture for downloading only
No clear Text Passwords	Yes	Yes
One Single Product	Yes	Yes
Insure Loginid Uniqueness	Yes – An attribute named NameID in SAML and URL has been used to identify the user who has been directed to IdP and has issue assertion about.	Yes – the username is the primary key in the database and determine the user identity number.
Encryption should be Commercial	Yes – Public Key Infrastructure is implemented.	Yes - Public Key Infrastructure is implemented.
Integrity of Security Database	Yes - Database in this system assume located in privacy computer. Only the computers in the network can access.	Yes – Database in this system assume located in privacy computer. Only the computers in the network can access.
Inactive User Time-out	Yes – Shibboleth introduce a software component known as LoginHandlers to authenticated user. It set a period of time that user can be authorized. After that period, user has to login to Shibboleth again.	No

One of the differences that has to be remarked here is that IdMM only provide IdP while Shibboleth provide both IdP and SP. After looking from the

comparative study table, even Shibboleth possesses more functionality remarked from [21] than IdMM. However, after study more information about steps install and implement Shibboleth, the program need a third party software helping the program to communicate to the components consist in it. Shibboleth, therefore, require fund. As discussed in problem statement, cost is one of the major factor in choosing the SSO program. IdMM, therefore, is the proposed program. Unfortunately, the guideline to make IdMM to work is only available in the Internet. Therefore, it is a need to port the guideline to be in Ubuntu version.

### 4.1.3 Architecture of OpenNebula

OpenNebula can be installed in 2 modes

- System-wide: binary, log files and all configuration will be installed and handles by root and only root can configure the system.
- Self-contained: OpenNebula will be installed set by the system. Everyone can handle the configuration in this mode.

An existing SSO system will be chosen to integrate with OpenNebula to be as a model to proof the concept of this project [1].

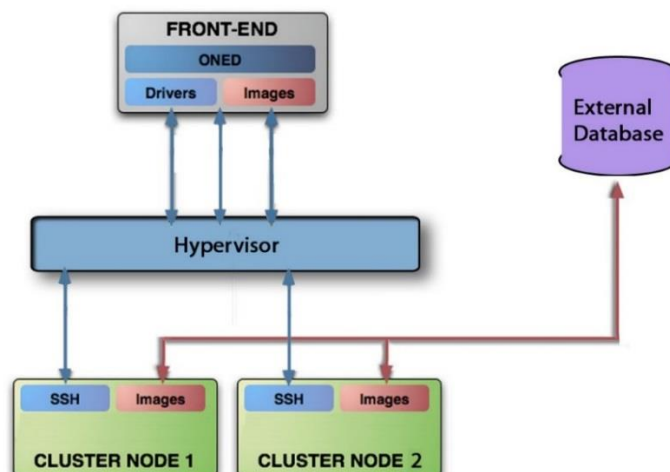


Figure 11 OpenNebula's Architecture

Administrator in OpenNebula use an account named oneadmin which is the manager of OpenNebula services. Figure 11 illustrates a simple architecture of expected outcome:

- Front-end for running OpenNebula and cluster service. The computer that run the service should have a big database to keep virtual machine image.
- Cluster nodes or nodes refer to provision virtual machine.
- Image repository is image of virtual machine's storage.
- Hypervisor refers as a virtualization manager which allow multiple operating systems share a single hardware host. There are many hypervisors available in the Internet as Xen and Hyper-V.
- Drivers is a program for connecting cluster system.
- SSH (Secure Shell) is network protocol for secure data transmission between front-end and cluster node. Front-end communicate with node via SSH by oneadmin by using SSH key.
- ONED (OpenNebula daemon) is a main service of the system. It control life-cycle of virtual machine and handle other system such as network, storage and hypervisor.
- External Database is where image for each virtual machine resides.

Before beginning any service in any cluster node, user must log in to single sign-on system to acknowledge the system that he/she is the authorized user. Single sign-on system reside in one cluster node in a computer cluster which generated by the front-end.

#### 4.1.4 Expected integrated proposed SSO system on OpenNebula

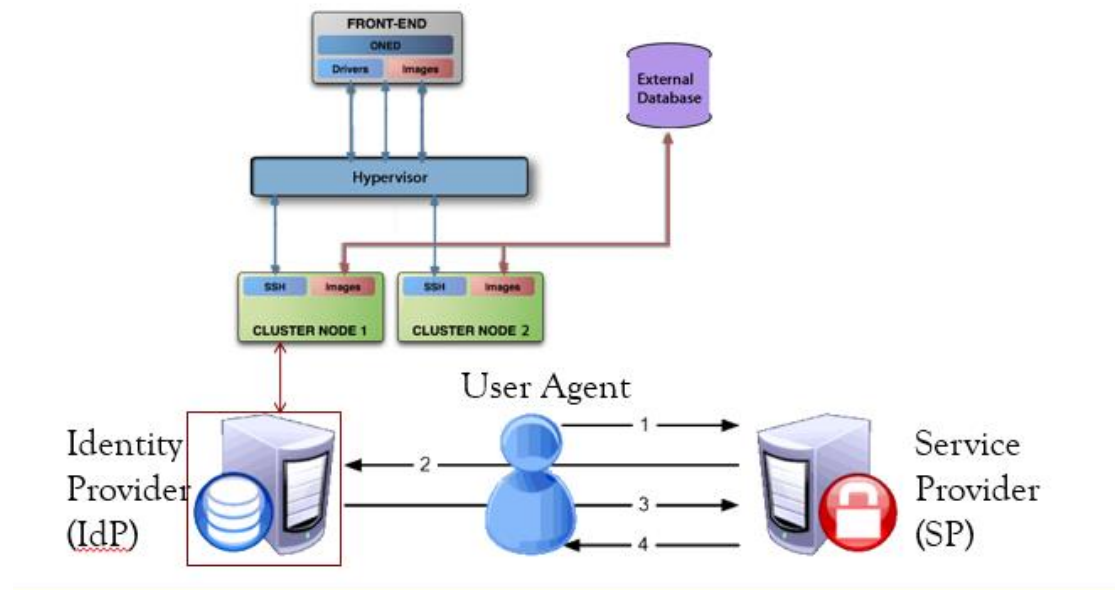


Figure 12 SSO system integrated with OpenNebula

Figure 12 illustrate a framework for a better understanding of expected system architecture. Key elements of this framework include OpenNebula’s architecture and overview of SSO system.

Identity provider of IdMM will be hosted in one of the virtual machine generate by OpenNebula. In addition, service provider can be hosted as well. However, it is outside the scope of this project due to timeframe. A more detail figure 13 is shown in figure 13. Apache DS, Apache Directory Studio, and Apache Tomcat are needed as important element for the IdMM to work. As IdMM is a SSO chrome extension. The extension part is held on the user’s computer.

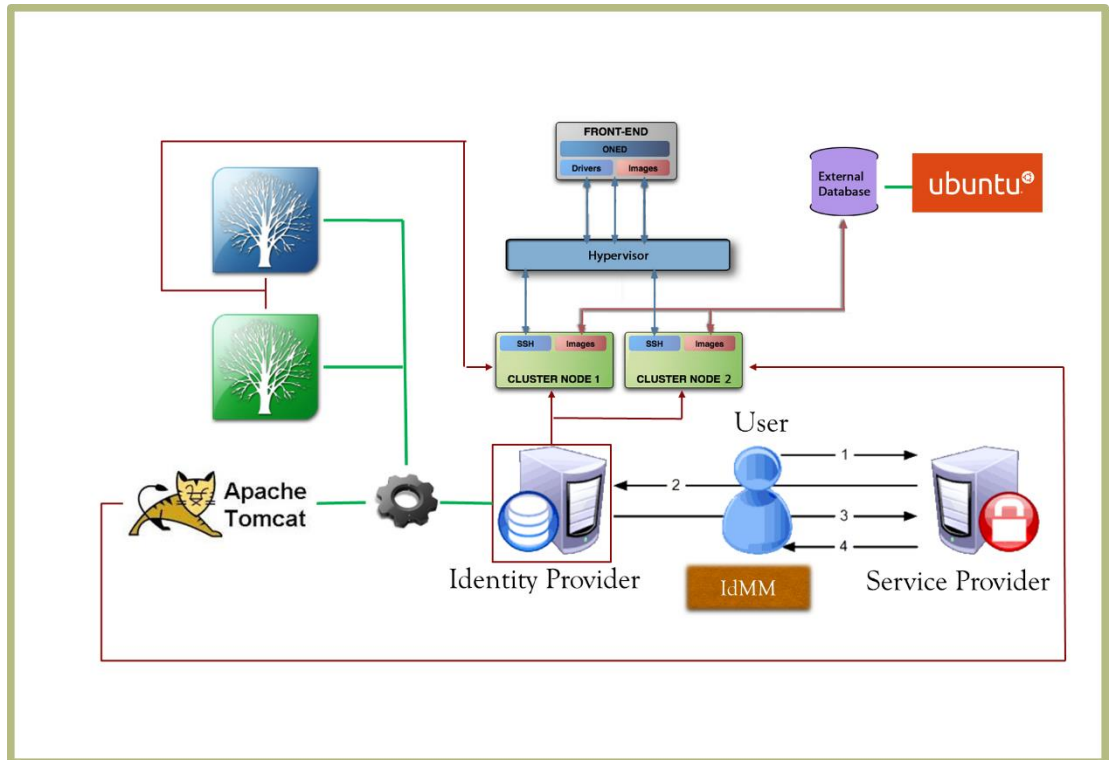


Figure 13 OpenNebula integration with IdMM

#### 4.1.5 Step to deploy IdMM

This section discusses how one can deploy IdMM SSO.

##### 4.1.5.1 Creating Virtual Machines

You will need to create two virtual machines on OpenNebula for Apache DS and Apache Directory Studio, and tomcat to reside. IdMM extension work exclusively on Windows platform. It is installed in any computer of the potential users. The first virtual machine will have a private IP address and the second one have a public IP address. In each virtual machine, installing Java is a must. These are the steps that should be done in all machines.

1. Open Terminal and run as root by entering

*sudo su*

Follow by the password to get administrative privilege.

2. Update Advanced Packaging Tool

*apt-get update*

3. Install Java

*apt-get install openjdk-7-jre*

#### **4.1.5.2 Creating the appropriate certificates**

Before this step begin, the IP address of the public machine must be known.

1. On one of the machines with Java JDK installed run the file `GenerateCertificates.jar`.

a. Open terminal and enter:

*java -jar PathWhereGenerateCertificatesIsLocated.jar*

*PathWhereGenerateCertificatesIsLocated.jar* is the path where *GenerateCertificates.jar* is located or enter to the folder where the file is located and enter.

*java -jar GenerateCertificates.jar*

2. The jar file will generate the following certificates:

a. `idmmclient.ks` – Stores the IdMMClient certificate

b. `apacheds.ks` – Stores the Apache DS certificate

c. `chrome.crt` – Stores the Google Certificate

The step should be done only one time and it is very important to remember the passwords for the `.ks` files that has been entered.

#### **4.1.5.3 Setup Apache DS and Apache Directory Studio**

In This step, Installation and configuration of Apache DS and Apache Directory Studio will be conducted. The configuration done in order to allow IdMMClient to retrieve data reside in Apache Directory Studio. The data consists of credential information of users including user account information, service protocol and authentications and so on. Apache DS and Apache Directory Studio should be resided in the private machine for security purpose. However, public machine will be used in this project.

## Installing and configuration Apache DS

1. Download Apache DS from <https://directory.apache.org/apacheds>
2. Install Apache DS by open terminal, log in as root and enter  

```
dpkg -i apacheds-2.0.0-M11-i386.deb
```

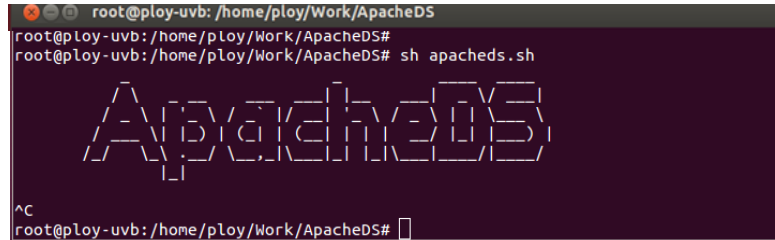
*apacheds-2.0.0-M11-i386.deb* is the name of the Apache DS package at the moment of writing. The package name, therefore, will be based on the package name at the installation time.
3. Move **apacheds.ks**, got the file in section 4.1.5.2, to */var/lib/apacheDS/conf*
4. Open **server.xml** which resides in */var/lib/apacheDS/conf*
  - i. Search for a **jdbmPartition** tag
    - a. Copy the whole tag and paste it next to the tag.
    - b. Change `jdpmPartition id` to *cdcc*

“*cdcc*” is an assume `jdpmPartition id`. The id itself represents the value act as URL or any name that represent the service. This step is made in order to allow Apache Directory Studio to accept that particular attributes.
  - ii. Search for a **ldapservlet** tag
    - a. add **keystoreFile** attribute and set the value to */var/lib/apacheDS/conf/apacheds.ks*
    - b. add **certificatePassword** attribute and set the password to the password you selected for *apacheds.ks* in creating the appropriate certificates section
  - iii. Search for a **transports** tag
    - a. Look for a `transports` tag which attribute **address** = “**0.0.0.0**” and change `nbThreads` to 2 and `enableSSL`.
    - b. Look for a `transports` tag which attribute **address** = “**localhost**” and `unableSSL`.
  - iv. Search for a **simpleMechanismHandler** tag which has `mech-name` attribute
    - a. Comment all the content inside the tag.
  - v. The modified version of this file can be seen in appendix.

5. Start apache by running *apacheds.sh* by entering the folder contain the file and follow the below command

*sh apacheds.sh*

If the server was installed correctly you should see this:



```
root@pLoy-uvb: /home/pLoy/Work/ApacheDS
root@pLoy-uvb: /home/pLoy/Work/ApacheDS#
root@pLoy-uvb: /home/pLoy/Work/ApacheDS# sh apacheds.sh
Apacheds
^C
root@pLoy-uvb: /home/pLoy/Work/ApacheDS#
```

Figure 14 Apacheds.sh's result screen.

### Installing and configuration Apache Directory Studio

6. Download Apache Directory Studio from

<https://directory.apache.org/studio/downloads.html>

7. Install Apache Directory Studio by enter

*tar -xvf ApacheDirectorStudio-linux-x86\_64-2.0.0.v20130628.tar.gz*

*ApacheDirectorStudio-linux-x86\_64-2.0.0.v20130628.tar.gz* is the name of the Apache DS package at the moment of writing. The package name, therefore, will be based on the package name at the installation time.

8. Open Apache Directory Studio and create a new connection with the following data:

- i. Connection Name: **IdMM Connection**
- ii. Hostname: **localhost**
- iii. Port: **389**
- iv. Encryption Method: **Use SSL Encryption**
- v. Authentication Method: **Simple Authentication**
- vi. Bind DN or user: **uid=admin,ou=system**
- vii. Bind Password: **secret**

9. Open the connection. If everything worked well you should see two children under RootDSE:

- i. ou=schema



- ii. ou=system

### **Importing Data into Apache Directory Studio**

There are 2 ways to store information Apache Directory Studio; create from scratch or import it. This section discuss about how one can create a new attribute in import a file in the application. The files for import have already been created by using example information. The files are

- authentication.ldif, contain authentication methods of services
- protocols.ldif, contain authentication methods of services
- services.ldif , contain services
- schema.ldif
- users.ldif contains lists of SSO users
- exampleUser.ldif, contains information of username and password of a particular user.

The steps to import information are listed as below

1. Right click on **ou=system** and select **Import >>LDIF Import >>**  
Use **services.ldif**
  - a. Select **schema.ldif**
  - b. If the import was successful you should see a **cn=idmm** under **ou=system**
2. Right click on the IdMM Connection and select Properties
  - a. Under *Connection* select **Schema**
  - b. Press **Reload Schema** a couple of times
  - c. Press **Ok**
3. Select *Root DSE* and then right click and select **New>>New Entry...** and in the wizard enter the following:
  - a. Create entry from scratch
  - b. Under *Available object classes* press **refresh** and then select **domain** and press **Add**
  - c. RDN is **dc** and the value is **cdcc**
  - d. After the wizard has finished you should see three values under Root DSE:

dc=cdcc  
ou=schema  
ou=system

4. Right Click on **dc=cdcc** and select **New>>New Entry**
  - a. Create entry from scratch
  - b. Available object classes: **organization**
  - c. RDN is **o** and value is **idmm**
5. Right Click on **o=idmm** and select **New>>New Entry**
  - a. Create entry from scratch
  - b. Available object classes: **organizationalUnit**
  - c. RDN is **ou** and value is **protocols**
6. Right Click on **ou=protocols** and select **Import>>LDIF Import>> Use protocols.ldif**
7. Right Click on **o=idmm** and select **New>>New Entry...**
  - a. Create entry from scratch
  - b. Available object classes: **organizationalUnit**
  - c. RDN is **ou** and value is **authentication**
8. Right Click on **ou=authentication** and select **Import>>LDIF Import>> authentication.ldif**
9. Right Click on **o=idmm** and select **New>>New Entry**
  - a. Create entry from scratch
  - b. Available object classes: **organizationalUnit**
  - c. RDN is **ou** and value is **services**
10. Right Click on **ou=services** and select **Import->LDIF Import>> Use services.ldif**
11. Right Click on **o=idmm** and select **New>>New Entry**
  - a. Create entry from scratch
  - b. Available object classes: **organizationalUnit**
  - c. RDN is **ou** and value is **users**
12. Right Click on **ou=users** and select **Import>>LDIF Import>>Use users.ldif**

13. Right Click on **uid=vleju** and select **Import>>LDIF Import>>**  
 Use **exampleUser.ldif**

14. If everything worked out you should see the following:

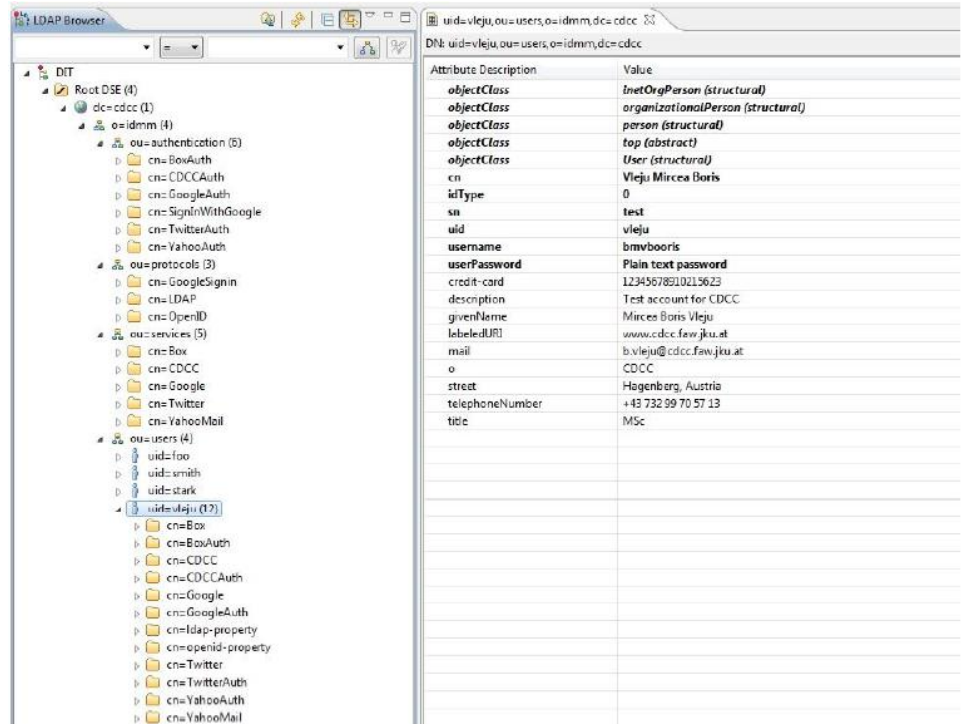


Figure 15 Apache Directory Studio's result screen.

### Creating the ApacheDS User

1. Right click on **ou=system** and select **New->New Entry**
  - a. Create entry from scratch
  - b. Available object classes: **inetOrgPerson**
  - c. RDN is **uid** and value is **idmm**
  - d. Set the values of **sn** and **cn** to **idmm**
  - e. Press the **New Attribute** button (Second button with a plus icon)
    - i. Attribute type: **userPassword** (press Finish)
    - ii. In the Password Editor window under Select **Hash Method** use **PlainText**.
    - iii. Enter a password and this will password will be refer as **ApacheDSUserPassword** in this report.

#### 4.1.5.4 Setup tomcat

In this step, installation and configuration of IdMMClient and Tomcat server which will be residing in public machine. It is responsible to handle the connection between user and Apache DS. The configuration is conducted to make the IdMMClient visible to the other computer and to create a connection between Apache DS and Tomcat server.

### Installing Tomcat

1. Install tomcat 6

```
apt-get install tomcat6 tomcat6-docs tomcat6-user tomcat6-admin tomcat6-example
```

2. Move **idmmclient.ks** and **apacheds.ks** to */var/lib/tomcat6/conf*

3. Open */var/lib/tomcat6/conf/server.xml*

- a. Search for a **Connector** tag which has the attribute

**SSLEnabled="true".**

- i. Add the path of where **idmmclient.ks** resides to the attribute **keystoreFile**

- ii. Add the password for **idmmclient.ks** to the attribute **keystorePass.**

- b. Search for a AJP 1.3 connector tag and uncomment the tag.

- c. The modified version of this file can be seen in appendice.

### Setting up the IdMMClient

1. Open the file **tomcat-users.xml** which resides in

*/var/lib/tomcat6/conf/* and add the following two lines to create account to login tomcat manager page:

```
<role rolename="manager-gui"/>
```

```
<user username="tomcat" password="12345" roles="manager-gui"/>
```

Username and password is an assumed one so it is possible to not follow this.

2. Restart tomcat server

```
/etc/init.d/tomcat6 restart
```

## Setting up the IdMMClient

**ROOT.war** is a file contain information of IdMM and also the connection how can tomcat server can be communicate with Apache Directory Studio. ROOT. War is provided in the IdMM packet.

1. Open **ROOT.war** by double click
  - a. Go to the folder WEB-INF and the in classes.
  - b. Edit the file client.properties
    - i. Set **login.host** to the IP where Apache Directory Studio is located.
    - ii. Set **login.user.password** to the password you used for **ApacheDSUserPassword**
    - iii. Set **login.certificate.path** to path of  
*/var/lib/tomcat6/conf/apacheds.ks*
    - iv. Set **login.certificate.password** to the password for **apacheds.ks**
  - c. Edit the file idmm-log.properties
    - i. Set **root** to */var/lib/tomcat6/logs*
  - d. Edit the file log4j.properties
    - i. Set **log4j.appender.CLIENT.File** to  
*/var/lib/tomcat6/conf/idmmclient.log*

## Deploying the IdMMClient

1. Open the web browser (in the machine) and enter the following address:
  - a. <https://localhost:8080/manager.html>
  - b. Enter username and password to login to **manager-gui** to tomcat manager page.
2. Go to *Deploy directory or WAR file located on server* and enter
  - a. *Context Path (required): /*
  - b. *WAR file to deploy* then upload **ROOT.war**  
At this step, it is important to make sure that there is no **ROOT** folder or **ROOT.war** in */var/lib/tomcat6/webapps*

3. Go to the url:
  - a. <https://localhost:8443/idmm>
  - b. If the deployment was ok, the web page should display the following message:  
*HTTP Status 405 - HTTP method GET is not supported by this URL*

#### **4.1.5.5 Setup Google Chrome**

1. Download Chrome from  
<https://www.google.com/intl/en/chrome/browser/>

#### **Install IdMM extension**

In Chrome go to settings and the extensions.  
Drag and Drop the **idmm.crx** and install the extension.

#### **Setting up the certificate for the IdMMClient for Windows**

In Chrome go to Settings->Show advanced settings->HTTPS/SSL and click manage certificates

- a. Select the **Trusted Root Certification Authorities** and click Import
- b. Select the file FILES/certificates/idmm-google-cert.crt
- c. If everything worked fine you can open the following address without any warning messages:
  - i. <https://x.x.x.x:8443/idmm> where x.x.x.x is the IP of the machine with the tomcat server.

#### **Testing Extension**

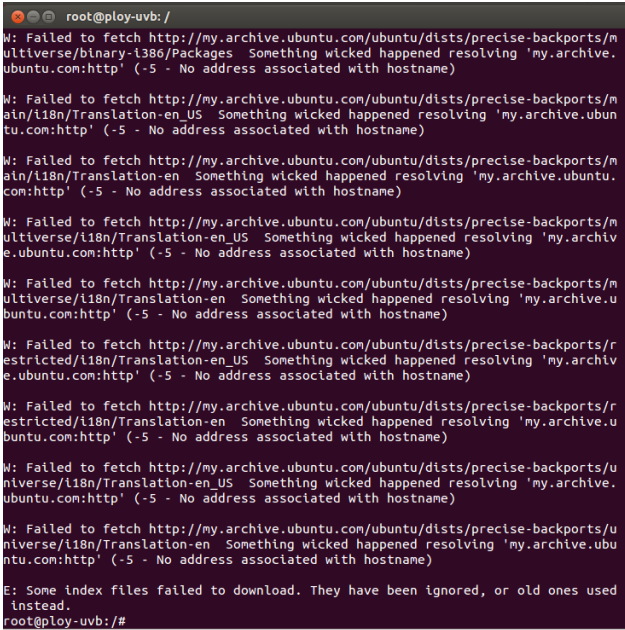
1. Enable the IdMM extension and go to options and select the following:

- a. Username: bmvbooris (it is an assume account)
  - b. Host: https://x.x.x.x:8443 where x.x.x.x is the IP of the machine with the tomcat server
  - c. Password: 12345 (leave the old password field blank)
2. Disable and the re-enable the IdMM extension
  3. If everything worked fine you should be able to go to access website at [www.twitter.com](http://www.twitter.com)

#### 4.1.6 Problems and Solutions

The section discusses about problems that have discovered together with all the attention to solve it. This section also identify the cause of the problem and its solution.

1. Problem[apt-get]: apt-get update fail to fetch



```

root@play-uvb: /
W: Failed to fetch http://my.archive.ubuntu.com/ubuntu/dists/precise-backports/main/i18n/Translation-en_US Something wicked happened resolving 'my.archive.ubuntu.com:http' (-5 - No address associated with hostname)
W: Failed to fetch http://my.archive.ubuntu.com/ubuntu/dists/precise-backports/main/i18n/Translation-en_US Something wicked happened resolving 'my.archive.ubuntu.com:http' (-5 - No address associated with hostname)
W: Failed to fetch http://my.archive.ubuntu.com/ubuntu/dists/precise-backports/main/i18n/Translation-en_US Something wicked happened resolving 'my.archive.ubuntu.com:http' (-5 - No address associated with hostname)
W: Failed to fetch http://my.archive.ubuntu.com/ubuntu/dists/precise-backports/main/i18n/Translation-en_US Something wicked happened resolving 'my.archive.ubuntu.com:http' (-5 - No address associated with hostname)
W: Failed to fetch http://my.archive.ubuntu.com/ubuntu/dists/precise-backports/restricted/i18n/Translation-en_US Something wicked happened resolving 'my.archive.ubuntu.com:http' (-5 - No address associated with hostname)
W: Failed to fetch http://my.archive.ubuntu.com/ubuntu/dists/precise-backports/restricted/i18n/Translation-en_US Something wicked happened resolving 'my.archive.ubuntu.com:http' (-5 - No address associated with hostname)
W: Failed to fetch http://my.archive.ubuntu.com/ubuntu/dists/precise-backports/universe/i18n/Translation-en_US Something wicked happened resolving 'my.archive.ubuntu.com:http' (-5 - No address associated with hostname)
W: Failed to fetch http://my.archive.ubuntu.com/ubuntu/dists/precise-backports/universe/i18n/Translation-en_US Something wicked happened resolving 'my.archive.ubuntu.com:http' (-5 - No address associated with hostname)
E: Some index files failed to download. They have been ignored, or old ones used instead.
root@play-uvb: /#

```

Figure 16 apt-get update fail to fetch

**Cause:** The figure 16 shown when trying to improve information of apt-get function.

**Solution:**

The error prompted to the screen respond to the first command to prepare the machine which lead that the problem related

to the system itself. The first attempt, therefore, was to restart the machine. However, it was not the problem's solution.

The second attempt was to delete the virtual machine and create it again. This consider as an anticipated potential solution. The problem might occur from impropriate configuration during creating process. In response, the error was still unsolvable.

After seeking information and opportunities for overcome this situation, the third attempt was adding a Domain Name Server (DNS) or Internet server. One reason for selecting this method based on the communication of Internet service provider and DNS. Adding a DNS by open terminal and type the following command:

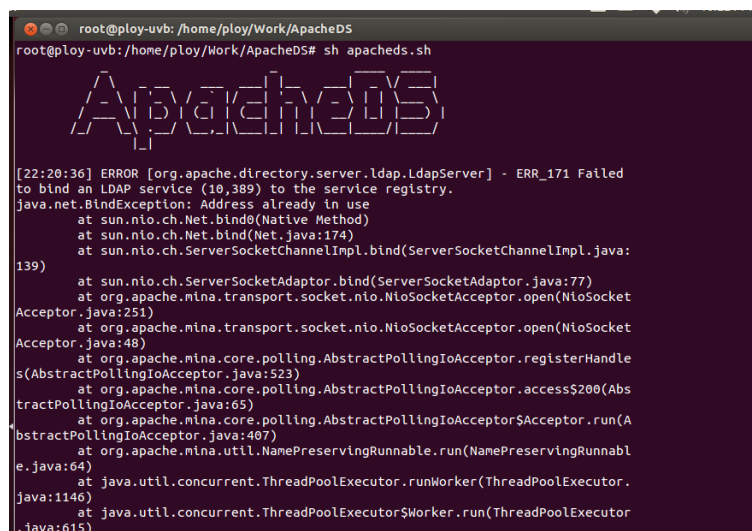
```
Echo "nameserver 8.8.8.8" | sudo tee /etc/resolv.conf > /dev/null
```

Then enter

```
Apt-get update
```

As conclusion, internet service provider is not correctly forwarding Internet naming or Domain Name Server (DNS) to either its or external DNS server.

## 2. Problem[Apache DS]: ERR\_171 Failed to bind an LDAP service to the service registry



```
root@pjoy-uvb: /home/pjoy/Work/ApacheDS
root@pjoy-uvb: /home/pjoy/Work/ApacheDS# sh apacheds.sh

ApacheDS

[22:20:36] ERROR [org.apache.directory.server ldap.LdapServer] - ERR_171 Failed
to bind an LDAP service (10,389) to the service registry.
java.net.BindException: Address already in use
    at sun.nio.ch.Net.bind0(Native Method)
    at sun.nio.ch.Net.bind(Net.java:174)
    at sun.nio.ch.ServerSocketChannelImpl.bind(ServerSocketChannelImpl.java:
139)
    at sun.nio.ch.ServerSocketAdaptor.bind(ServerSocketAdaptor.java:77)
    at org.apache.mina.transport.socket.nio.NioSocketAcceptor.open(NioSocket
Acceptor.java:251)
    at org.apache.mina.transport.socket.nio.NioSocketAcceptor.open(NioSocket
Acceptor.java:48)
    at org.apache.mina.core.polling.AbstractPollingIoAcceptor.registerHandle
s(AbstractPollingIoAcceptor.java:523)
    at org.apache.mina.core.polling.AbstractPollingIoAcceptor.access$200(Abs
tractPollingIoAcceptor.java:65)
    at org.apache.mina.core.polling.AbstractPollingIoAcceptor$Acceptor.run(A
bstractPollingIoAcceptor.java:407)
    at org.apache.mina.util.NamePreservingRunnable.run(NamePreservingRunnabl
e.java:64)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.
java:1146)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor
.java:615)
```

Figure 17 ERR\_171 Failed to bind an LDAP service to the service registry



Cause: As the figure 17 shown, terminal reported an error after Apache DS is started.

Solution:

The first attempt was simply login to Apache Directory Studio to identify the effect of the error. The system was unable to complete building a connection by using SSL encryption as an encryption method which shown as the figure 18 below:

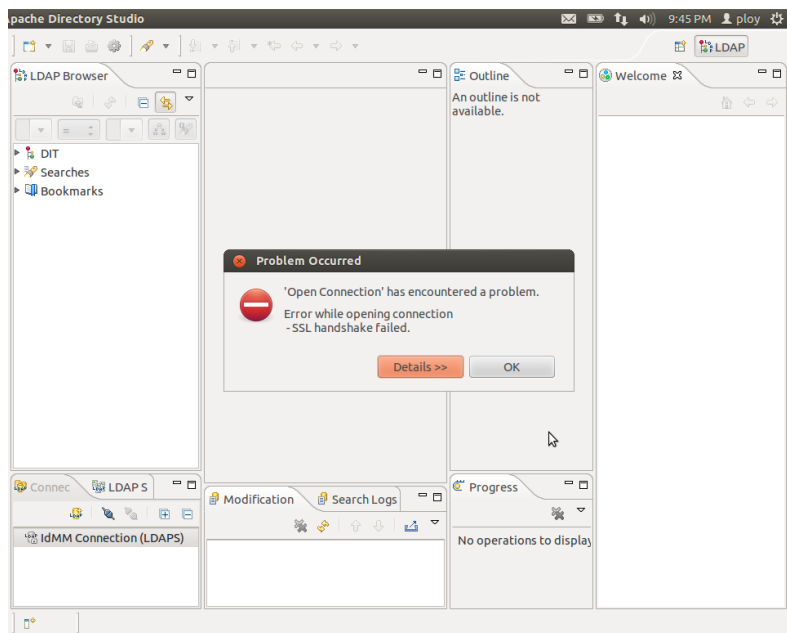
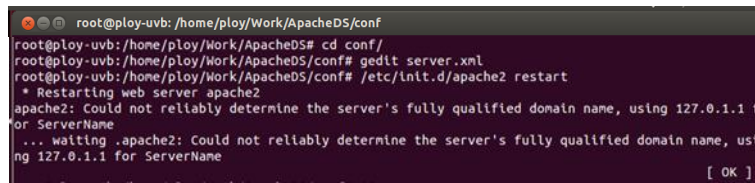


Figure 18 LDAP: SSL handshake failed

The second attempt was changing the port number to be alternative ones. 10636 was the first number and it was a failure. The same error is prompted to the screen. After taking some time to gather information to review the alternative ports. Port 389 was being use as the delegate, and the connection could be build.

As conclusion, due to the specified port is being used by other programs, Apache Directory Studio cannot use the port to create a connection.

3. Problem[Apache2]: Could not determine the service's fully qualified domain name

A terminal window showing the process of restarting Apache2. The user is in the directory /home/pjoy/Work/ApacheDS/conf. They run 'cd conf', 'gedit server.xml', and '/etc/init.d/apache2 restart'. The terminal output shows the Apache2 service starting and then displaying an error: 'apache2: could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName'. The error message is repeated twice. The terminal ends with a '[ OK ]' prompt.

```
root@pjoy-uvb: /home/pjoy/Work/ApacheDS/conf
root@pjoy-uvb: /home/pjoy/Work/ApacheDS# cd conf/
root@pjoy-uvb: /home/pjoy/Work/ApacheDS/conf# gedit server.xml
root@pjoy-uvb: /home/pjoy/Work/ApacheDS/conf# /etc/init.d/apache2 restart
 * Restarting web server apache2
apache2: could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName
... waiting ,apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName
[ OK ]
```

Figure 19 Could not determine the service's fully qualified domain name

Cause: Figure 19 show a caution when restarting Apache2.

Solution:

Unlike others, this issue is not exactly a problem. It is more to a warning from the system saying that any error that occurring might happened due to the fact that the server, Apache2, point to user customize host instead of local host. 127.0.0.1 refers to localhost and 127.0.1.1 refers to myhostname. Changing the point can be done by open terminal and do as follow:

Open **httpd.conf** or **apache2.conf** in */etc/apache2* and write the following command

*ServerName localhost*

And restart the service

#### 4. Problem: LDAP: error code 32 – No such object

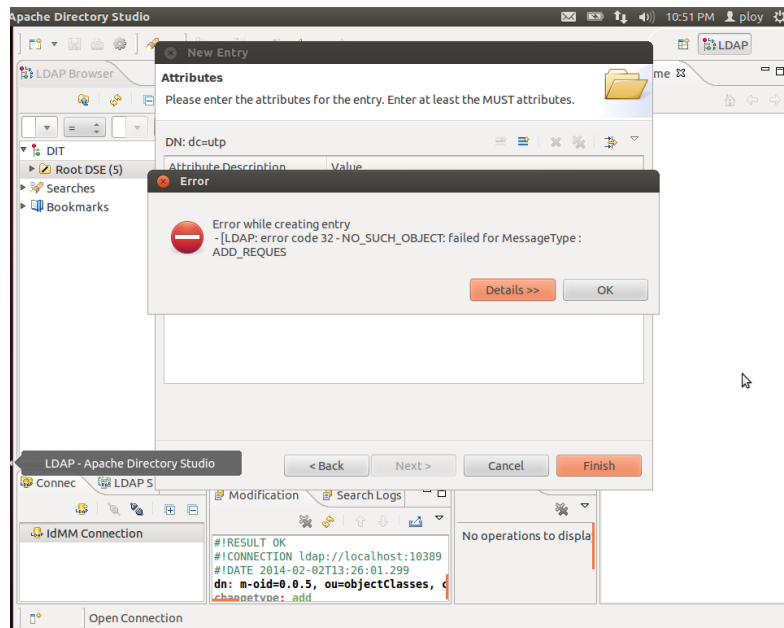


Figure 20 LDAP: error code 32 – No such object

**Cause:** The error window prompted on the screen when trying to add new entry to Root DSE. It stated that the system failed to add the requested attribute (RDN as dc and the value as cdcc) into the list.

**Solution:**

The first attempt for solving problem was randomly add the attributes to search for all the possibilities ones that might be the potential attributes. However, the actions were failures.

The second attempt was simply uninstall the program and reinstall it again. Perhaps part of the reason this problems were because of some configurations that has been made during attempting of experiment and solving other problems. But that was the cause of the problem.

The third attempt was looking into the configurations of the Apache DS in server.xml. The attempt was to simply define an attribute in the file at **jdbmPartition** tag as follow:

```
<jdbmPartition id="cdcc" cacheSize="100" suffix="dc=cdcc" optimizerEnabled="true" syncOnWrite="true">
```

```

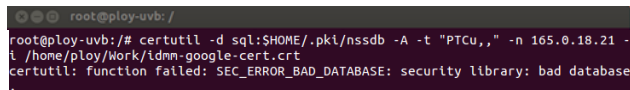
<indexedAttributes>
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.1" cacheSize="100"/>
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.2" cacheSize="100"/>
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.3" cacheSize="100"/>
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.4" cacheSize="100"/>
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.5" cacheSize="10"/>
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.6" cacheSize="10"/>
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.7" cacheSize="10"/>
<jdbmIndex attributeId="dc" cacheSize="100"/>
<jdbmIndex attributeId="ou" cacheSize="100"/>
<jdbmIndex attributeId="krb5PrincipalName" cacheSize="100"/>
<jdbmIndex attributeId="uid" cacheSize="100"/>
<jdbmIndex attributeId="objectClass" cacheSize="100"/>
</indexedAttributes>
</jdbmPartition>

```

And it is a success. The system added new attribute once the attribute defined dc and cdcc as RDN and RDN's values respectively.

As conclusion, the problem, therefore, occur because the attribute that tried to add is not defined in server.xml.

## 5. Problem[Certutil]: SEC error bad database



```

root@pjoy-uvb: /
root@pjoy-uvb: /# certutil -d sql:$HOME/.pki/nssdb -A -t "PTCu,," -n 165.0.18.21 -t /home/pjoy/Work/idm-google-cert.crt
certutil: function failed: SEC_ERROR_BAD_DATABASE: security library: bad database

```

Figure 21 SEC error bad database

Cause: Figure 21 illustrates an error when authenticating certificate.

The certificate database is not initialized.

Solution:

After determining the cause the problem, the error is based on the existence of the database in the virtual machine. The certificate database was not created. In response, the certificate could not be added into the system. To complete the action, one must do the following command:

```
mkdir -p $HOME/.pki/nssdb
certutil -d $HOME/.pki/nssdb -N
```

The below figure 22 appeared once the first command is performed.



Figure 22 SEC error IO – Could not authenticate to taken NSS Certificate

## 4.2 EXPERIMENTAL SETUP

### 4.2.1 Data

The following data in the below tables are input data which be tested in system testing.

Input ID	Input Data
In -1	IdMM Extension
In -2	Certificate
In -3	Operating System

Input ID	{“username“, “password“, “host“, “service provider URL“ }
In -4	"bmvboomris"; "12345"; " https://:8443"; "-";
In -5	"username"; "12345"; " https://:8443"; "-";
In -6	"bmvboomris"; "password"; " https://:8443"; "-";
In -7	"bmvboomris"; "12345"; " https://:8080"; "-";
Input ID	{“username“, “password“, “host“, “service provider URL“ }
In -8	"bmvboomris"; "12345"; " https://192.168.56.10:8443"; "-";
In -9	"bmvboomris"; "12345"; " http://:8443"; "-";
In -10	"bmvboomris"; "12345"; " https://:8443"; "www.hotmail.com";
In -11	"bmvboomris"; "12345"; " https://:8443"; "www.twitter.com";

<b>Input ID</b>	{“username“, “old password“, “new password“, “ host“, “service provider URL“ }
In -12	"bmvboomris"; "password";"12345";" https://:8443"; "-";

The correct information that allow program to be worked is in-4.

#### 4.2.2 Platform

Just as Ubuntu12.04 play an essential Platform in this project, so as in the testing process. All of the the softwares involving in the testing run on it. There are, however, some test cases and some programs that run on different operating system. Any tools including test case id 1 and 1.2, which focus on the IdMM extensions and IdMM certificates, run on Windows 8.

#### 4.2.3 Software

To complete the testing process, the tools were used are same as listed in section 3.2 required tools.

#### 4.2.4 Steps for testing

Except test case id IT2 and IT2.2 the setp to prepare the testing are as discussed in section 4.

#### Setting up the certificate for the IdMMClient for Ubuntu

Before this step is conducted, Chrome needs to be closed while the certificate is being installed.

1. Open terminal and login as root user. Install the **certutil** tool.

*apt-get install libnss3-tools*

This step must be done only once. Figure 23 shows how the result might look like

```

root@pjoy-uvb: /
root@pjoy-uvb: /# apt-get install libnss3-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  thunderbird-globalmenu gtr1.2-ubuntuoneut-3.0 firefox-globalmenu
  libubuntuoneut-3.0-1
Use 'apt-get autoremove' to remove them.
The following NEW packages will be installed:
  libnss3-tools
0 upgraded, 1 newly installed, 0 to remove and 15 not upgraded.
Need to get 461 kB of archives.
After this operation, 1,119 kB of additional disk space will be used.
Get:1 http://my.archive.ubuntu.com/ubuntu/ precise-updates/universe libnss3-tools
  i386 3.15.4-0ubuntu0.12.04.1 [461 kB]
Fetched 461 kB in 18s (25.2 kB/s)
Selecting previously unselected package libnss3-tools.
(Reading database ... 167333 files and directories currently installed.)
Unpacking libnss3-tools (from ../libnss3-tools_3.15.4-0ubuntu0.12.04.1_i386.deb)
...
Setting up libnss3-tools (3.15.4-0ubuntu0.12.04.1) ...

```

Figure 23 How does libnss3-tools should look like

2. Create a certificate database

```
mkdir -p $HOME/.pki/nssdb
```

```
certutil -d $HOME/.pki/nssdb -N
```

3. Install the certificate. Enter the following command:

```
certutil -d sql:$HOME/.pki/nssdb -A -t "PTCu,," -n <ip> -i
<certificate>
```

Where:

- i. <ip> is the IP of the virtual machine with tomcat
- ii. <certificate> is the location of the certificate (also include the file name)
  - a. To check that the certificate was installed run the command:

```
certutil -d sql:$HOME/.pki/nssdb -L -n <ip>
```
  - b. To remove the certificate run the command:

```
certutil -d sql:$HOME/.pki/nssdb -D -n <ip>
```

#### 4.2.5 Error rate

The value to measure the accuracy for the experiment result is error rate. This is a way to measure the result of each test. The error rate for each test is listed as the following:

- Install Ability Test

$$P(\text{Making an error}) = 0.0$$

Therefore, the error rate for install ability test is 0

- Security (Confidentiality) Test

$$P(\text{Making an error}) = 0.9$$

Therefore, the error rate for security test is 0.9 or 90 %

- Black Box Test

$$P(\text{Making an error}) = 0.5$$

Therefore, the error rate for black box test is 0.5 or 50%

### 4.3 SYSTEM TESTING

#### 4.3.1 Install ability Testing

Test Case Identifier	IT1
Purpose	Verify that IdMM can boot up its extension in Windows 8, Operating System
Input	In-1, In-2, and In-3
Expected Output	IdMM extension is added into the listed of the Chrome extension.
Actual Output	IdMM extension is added into the listed of the Chrome extension.
Intercase Dependencies	-
Test Case Identifier	IT1.2
Purpose	Verify that login function work preoperly and login successfully.
Input	In-1, In-2, and In-3
Expected Output	Certificates is installed and accepted by Chrome
Actual Output	Certificates is installed and accepted by Chrome
Intercase Dependencies	IT1 actual result pass
Test Case Identifier	IT2
Purpose	Verify that IdMM can boot up its extension in Ubuntu 12.04, Operating System
Input	In-1, In-2, and In-3



Expected Output	IdMM extension is added into the listed of the Chrome extension.
Actual Output	IdMM extension is added into the listed of the Chrome extension.
Intercase Dependencies	-
Test Case Identifier	IT2.2
Purpose	Verify that login function work preoperly and login successfully.
Input	In-1,In-2, and In-3
Expected Output	Certificates is installed and accepted by Chrome
Actual Output	Certificates is installed and accepted by Chrome
Intercase Dependencies	IT2 actual result pass

#### 4.3.2 Security Testing (Confidential)

Test Case Identifier	ST1
Purpose	Verify that login function work preoperly and login successfully.
Input	In-4
Expected Output	Login to IdMM successfully
Actual Output	Login to IdMM successfully
Intercase Dependencies	IT1, IT1.2, IT2 and IT2.2 are tested and either IT1 and IT1.2 or IT2 and IT2.2 actual results or all of them pass.

Test Case Identifier	ST2
Purpose	Verify that login function work preoperly, detect wrong data and fail to login.
Input	In-5
Expected Output	Fail to login IdMM
Actual Output	Failed to login IdMM

Intercase Dependencies	IT1, IT1.2, IT2 and IT2.2 are tested and either IT1 and IT1.2 or IT2 and IT2.2 actual results or all of them pass. ST1 actual result pass.
Test Case Identifier	ST3
Purpose	Verify that login function work preoperly, detect wrong data and fail to login.
Input	In-6
Expected Output	Fail to login IdMM
Actual Output	Failed to login IdMM
Intercase Dependencies	IT1, IT1.2, IT2 and IT2.2 are tested and either IT1 and IT1.2 or IT2 and IT2.2 actual results or all of them pass. ST1 actual result pass.
Test Case Identifier	ST4
Purpose	Verify that login function work preoperly, detect wrong data and fail to login.
Input	In-7
Expected Output	Fail to login IdMM
Actual Output	Failed to login IdMM
Intercase Dependencies	IT1, IT1.2, IT2 and IT2.2 are tested and either IT1 and IT1.2 or IT2 and IT2.2 actual results or all of them pass. ST1 actual result pass.
Test Case Identifier	ST5
Purpose	Verify that login function work preoperly, detect wrong data and fail to login.
Input	In-8

Expected Output	Fail to login IdMM
Actual Output	Failed to login IdMM
Intercase Dependencies	IT1, IT1.2, IT2 and IT2.2 are tested and either IT1 and IT1.2 or IT2 and IT2.2 actual results or all of them pass. ST1 actual result pass.
Test Case Identifier	ST6
Purpose	Verify that login function work preoperly, detect wrong data and fail to login.
Input	In-9
Expected Output	Fail to login IdMM
Actual Output	Failed to login IdMM
Intercase Dependencies	IT1, IT1.2, IT2 and IT2.2 are tested and either IT1 and IT1.2 or IT2 and IT2.2 actual results or all of them pass. ST1 actual result pass.
Test Case Identifier	ST7
Purpose	Verify that login function work preoperly, detect wrong data and fail to login.
Input	In-12
Expected Output	Fail to login IdMM
Actual Output	Failed to login IdMM
Intercase Dependencies	IT1, IT1.2, IT2 and IT2.2 are tested and either IT1 and IT1.2 or IT2 and IT2.2 actual results or all of them pass. ST1 actual result pass.

### 4.3.3 Black box Testing

Test Case Identifier	BT1
----------------------	-----

Purpose	Verify that the system work properly,
Input	In-10
Expected Output	Hotmail is not login by IdMM
Actual Output	Hotmail is not login by IdMM
Intercase Dependencies	All the test cases in security test has been tested and passed
Test Case Identifier	BT2
Purpose	Verify that the system work properly,
Input	In-11
Expected Output	Twitter is being login automatically by IdMM.
Actual Output	Twitter is being login automatically by IdMM.
Intercase Dependencies	All the test cases in security test has been tested and passed

#### 4.4 TEST RESULT

A total of 11 test cases were designed to be executed. A total of 2, 7 and 2 test cases were planned to be executed in install ability test, security test and black box test respectively. Table 3 and figure 24 summaries test result in section 4.3

Table 3 Summary system testing

Test	The number of total test	Passed	Failed	Invalid	Executed	Passed/Executed (%)
Install ability Test	2	1	1	0	2	50
Security (Confidentiality) Test	7	7	0	0	7	100

Black box Test	2	2	0	0	2	100
----------------	---	---	---	---	---	-----

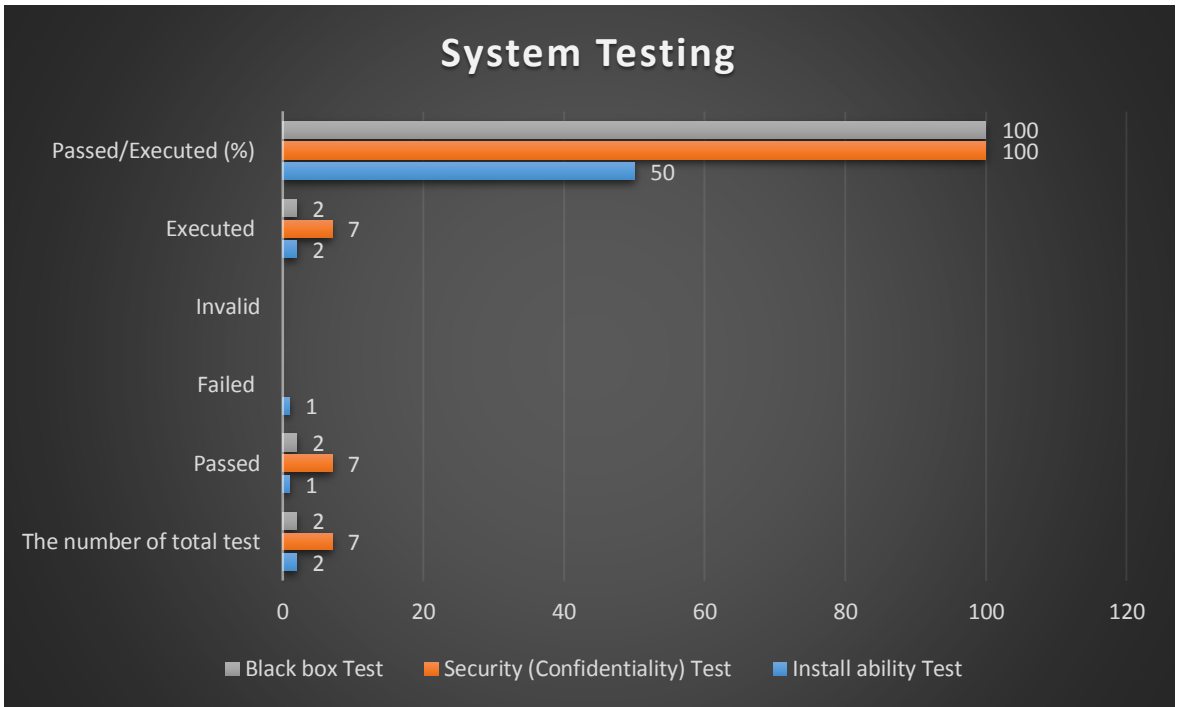


Figure 24 Summary of system testing

# CHAPTER 5

## CONCLUSION

### 5.1 Conclusion

The project has been conducted to meet the need of how one can decide which SSO system is suitable for them and how to implement one. There were two Single Sign On systems involve in the project Shibboleth and Identity Management Machine. The project are now achieved by proposing IdMM as a preferable SSO system. Even Shibboleth has better functional requirements found in the comparative study but the program requires extra payment for third party to make the communication between the program and service providers to work. However, guideline how to install IdMM is only available in Windows version. A guideline how to install it in Ubuntu, therefore, has been developed. After the installation process, system testing was conducted to verify that the system work as expectation. Twitter involved in this project as the service to proof the concept and it did.

For future work, more SSO systems should be involved in comparative study. The proposed SSO is chosen on their functional and the context of the situation. Having more SSO systems in the comparative analysis will increase the chance for the project to have a better suitable proposed SSO system including more options to decide. And to get a better result in system testing, more services should be included as Google, Facebook, and many more including a new create service.

As the conclusion for the project, IdMM can be integrated with OpenNebula and used with Twitter. By installing IdMM in OpenNebula, one can deploy a SSO system that offer a minimum level of security, at a low cost in OpenNebula.

# Appendices

## Appendice 1.1: Apache DS - Server.xml

<!--

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-->

```
<spring:beans xmlns="http://apacheds.org/config/1.5.6" xmlns:spring="http://xbean.apache.org/schemas/spring/1.0" xmlns:s="http://www.springframework.org/schema/beans">
```

```
<defaultDirectoryService id="directoryService" instanceId="default" replicaId="1" workingDirectory="idmm" allowAnonymousAccess="false" accessControlEnabled="false" denormalizeOpAttrsEnabled="false" syncPeriodMillis="15000" maxPDUSize="200000">
```

```
<systemPartition>
```

```
<!--
```

use the following partitionConfiguration to override defaults for

```
-->
```

```
<!-- the system partition
```

```
-->
```

```
<jdbmPartition id="system" cacheSize="100" suffix="ou=system" optimizerEnabled="true" syncOnWrite="true">
```

```
<indexedAttributes>
```

```
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.1" cacheSize="100"/>
```

```
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.2" cacheSize="100"/>
```

```
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.3" cacheSize="100"/>
```

```
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.4" cacheSize="100"/>
```

```
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.5" cacheSize="10"/>
```

```
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.6" cacheSize="10"/>
```

```
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.7" cacheSize="10"/>
```

```
<jdbmIndex attributeId="ou" cacheSize="100"/>
```

```
<jdbmIndex attributeId="uid" cacheSize="100"/>
```

```
<jdbmIndex attributeId="objectClass" cacheSize="100"/>
```

```
</indexedAttributes>
```

```
</jdbmPartition>
```

```
</systemPartition>
```

```
</partitions>
```

```
<!--
```

NOTE: when specifying new partitions you need not include those

```
-->
```

```
<!--
```

attributes below with OID's which are the system indices, if left

```
-->
```

```
<!--
```

out they will be automatically configured for you with defaults.

```
-->
```

```
<jdbmPartition id="example" cacheSize="100" suffix="dc=example,dc=com" optimizerEnabled="true" syncOnWrite="true">
```

```
<indexedAttributes>
```

```
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.1" cacheSize="100"/>
```

```
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.2" cacheSize="100"/>
```

```
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.3" cacheSize="100"/>
```

```
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.4" cacheSize="100"/>
```

```
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.5" cacheSize="10"/>
```

```
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.6" cacheSize="10"/>
```

```
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.7" cacheSize="10"/>
```

```
<jdbmIndex attributeId="dc" cacheSize="100"/>
```

```
<jdbmIndex attributeId="ou" cacheSize="100"/>
```

```
<jdbmIndex attributeId="krb5PrincipalName" cacheSize="100"/>
```

```
<jdbmIndex attributeId="uid" cacheSize="100"/>
```

```
<jdbmIndex attributeId="objectClass" cacheSize="100"/>
```

```
</indexedAttributes>
```

```

</jdbmPartition>
<jdbmPartition id="cdcc" cacheSize="100" suffix="dc=cdcc" optimizerEnabled="true" syncOnWrite="true">
<indexedAttributes>
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.1" cacheSize="100"/>
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.2" cacheSize="100"/>
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.3" cacheSize="100"/>
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.4" cacheSize="100"/>
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.5" cacheSize="10"/>
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.6" cacheSize="10"/>
<jdbmIndex attributeId="1.3.6.1.4.1.18060.0.4.1.2.7" cacheSize="10"/>
<jdbmIndex attributeId="dc" cacheSize="100"/>
<jdbmIndex attributeId="ou" cacheSize="100"/>
<jdbmIndex attributeId="krb5PrincipalName" cacheSize="100"/>
<jdbmIndex attributeId="uid" cacheSize="100"/>
<jdbmIndex attributeId="objectClass" cacheSize="100"/>
</indexedAttributes>
</jdbmPartition>
</partitions>
<interceptors>
<normalizationInterceptor/>
<authenticationInterceptor/>
<referralInterceptor/>
<aciAuthorizationInterceptor/>
<defaultAuthorizationInterceptor/>
<exceptionInterceptor/>
<operationalAttributeInterceptor/>
<!--
Uncomment to enable the password policy interceptor
  <passwordPolicyInterceptor/>
  <keyDerivationInterceptor/>

-->
<schemaInterceptor/>
<subentryInterceptor/>
<collectiveAttributeInterceptor/>
<eventInterceptor/>
<triggerInterceptor/>
<!--
Uncomment to enable replication interceptor
  <replicationInterceptor>
    <configuration>
      <replicationConfiguration serverPort="10390" peerReplicas="instance_b@localhost:10392">
        <replicaId>
          <replicaId id="instance_a"/>
        </replicaId>
      </replicationConfiguration>
    </configuration>
  </replicationInterceptor>

-->
</interceptors>
<!-- Uncomment to enable replication configuration -->
<!--
replicationConfiguration>
  <providers>
    <provider id="1" type="refreshAndPersist" timeLimit="1000" sizeLimit="1000">
      <url>
        ldap://ldap1.acme.com:10389/ou=data,dc=acme,dc=com?*,+?sub?(objectClass=*)
      </url>
      <connection bindMethod="simple">
        <principal>
          uid=admin,ou=system
        </principal>
        <credentials>secret</credentials>
      </bind>
    </provider>
    <provider id="2" type="refreshAndPersist" timeLimit="1000" sizeLimit="1000">
      <url>
        ldaps://ldap2.acme.com:10389/ou=data,dc=acme,dc=com?*,+?sub?(objectClass=*)
      </url>
      <connection bindMethod="simple">
        <principal>
          uid=admin,ou=system
        </principal>
        <credentials>secret</credentials>

```



```

        </bind>
      </provider>
    </providers>
  </replicationConfiguration>
-->
</defaultDirectoryService>
<!--

+=====+
| ChangePassword server configuration          |
+=====+

-->
<!--
missing atou=users,dc=example,dc=com
<changePasswordServer id="changePasswordServer">
  <transports>
    <tcpTransport port="60464" nbThreads="2" backlog="50"/>
    <udpTransport port="60464" nbThreads="2" backlog="50"/>
  </transports>
  <directoryService>#directoryService</directoryService>
</changePasswordServer>
-->
<!--

+=====+
| Kerberos server configuration              |
+=====+

-->
<!--
missing atou=users,dc=example,dc=com
<kdcServer id="kdcServer">
  <transports>
    <tcpTransport port="60088" nbThreads="4" backlog="50"/>
    <udpTransport port="60088" nbThreads="4" backlog="50"/>
  </transports>
  <directoryService>#directoryService</directoryService>
</kdcServer>
-->
<!--

+=====+
| NtpServer configuration                   |
+=====+

-->
<!--
ntpServer
  <transports>
    <tcpTransport port="60123"/>
    <udpTransport port="60123" nbThreads="1"/>
  </transports>
</ntpServer
-->
<!--

+=====+
| DnsServer configuration                  |
+=====+

-->
<!--
missing atou=users,dc=example,dc=com
<dnsServer>
  <transports>
    <tcpTransport port="8053"/>
    <udpTransport port="8053"/>
  </transports>
  <directoryService>#directoryService</directoryService>
</dnsServer>
-->
<!--

+=====+

```

| LDAP Service configuration |

```
+=====+
-->
<ldapServer id="ldapServer" allowAnonymousAccess="false" saslHost="localhost" saslPrincipal="ldap/localhost" searchBase
Dn="ou=users,ou=system" maxTimeLimit="15000" maxSizeLimit="1000" keystoreFile="Certificate File
Here" certificatePassword="Certificate Password Here">
<transports>
<tcpTransport address="0.0.0.0" port="10389" nbThreads="3" backlog="50" enableSSL="true"/>
<!--
        <tcpTransport address="localhost" port="10636" enableSSL="false" />
-->
</transports>
<directoryService>#directoryService</directoryService>
<!-- The list of supported authentication mechanisms. -->
<saslMechanismHandlers>
<simpleMechanismHandler mech-name="SIMPLE"/>
<cramMd5MechanismHandler mech-name="CRAM-MD5"/>
<digestMd5MechanismHandler mech-name="DIGEST-MD5"/>
<gssapiMechanismHandler mech-name="GSSAPI"/>
<ntlmMechanismHandler mech-name="NTLM" ntlmProviderFqcn="com.foo.Bar"/>
<ntlmMechanismHandler mech-name="GSS-SPNEGO" ntlmProviderFqcn="com.foo.Bar"/>
</saslMechanismHandlers>
<!--
The realms serviced by this SASL host, used by DIGEST-MD5 and GSSAPI.
-->
<saslRealms>
<s:value>example.com</s:value>
<s:value>apache.org</s:value>
</saslRealms>
<!--
the collection of extended operation handlers to install
-->
<extendedOperationHandlers>
<startTlsHandler/>
<gracefulShutdownHandler/>
<launchDiagnosticUiHandler/>
<!--
The Stored Procedure Extended Operation is not stable yet and it may cause security risks.
-->
<!-- storedProcedureExtendedOperationHandler/ -->
</extendedOperationHandlers>
</ldapServer>
<apacheDS id="apacheDS">
<ldapServer>#ldapServer</ldapServer>
</apacheDS>
<!--
uncomment the below line to start the jetty(v6.1.14) http server
This can be used to provide access to the data present in DIT via http
using a web application
-->
<!--
        <httpServer id="httpServer" port="7009" >
        <webApps>
        <webApp warFile="/path/to/war/file" contextPath="/myApp"/>
        </webApps>
        </httpServer>
-->
</spring:beans>
```

## Appendice 1.2: Tomcat - server.xml

<!--

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-->

<!--

Note: A "Server" is not itself a "Container", so you may not define subcomponents such as "Valves" at this level. Documentation at /docs/config/server.html

-->

<Server port="8005" shutdown="SHUTDOWN">

<!--

Security listener. Documentation at /docs/config/listeners.html  
<Listener className="org.apache.catalina.security.SecurityListener" />

-->

<!--

APR library loader. Documentation at /docs/apr.html

-->

<Listener className="org.apache.catalina.core.AprLifecycleListener"SSLEngine="on"/>

<!--

Initialize Jasper prior to webapps are loaded. Documentation at /docs/jasper-howto.html

-->

<Listener className="org.apache.catalina.core.JasperListener"/>

<!--

Prevent memory leaks due to use of particular java/javax APIs

-->

<Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/>

<ListenerclassName="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"/>

<ListenerclassName="org.apache.catalina.core.ThreadLocalLeakPreventionListener"/>

<!--

Global JNDI resources

Documentation at /docs/jndi-resources-howto.html

-->

<GlobalNamingResources>

<!--

Editable user database that can also be used by  
UserDatabaseRealm to authenticate users

-->

<Resource name="UserDatabase" auth="Container"type="org.apache.catalina.UserDatabase" description="User database that can be updated and saved"factory="org.apache.catalina.users.MemoryUserDatabaseFactory"pathname="conf/tomcat-users.xml"/>

</GlobalNamingResources>

<!--

A "Service" is a collection of one or more "Connectors" that share a single "Container" Note: A "Service" is not itself a "Container", so you may not define subcomponents such as "Valves" at this level. Documentation at /docs/config/service.html

-->

<Service name="Catalina">

<!--

The connectors can use a shared executor, you can define one or more named thread pools

-->

```

<!--
    <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
        maxThreads="150" minSpareThreads="4"/>
-->
<!--
A "Connector" represents an endpoint by which requests are received
and responses are returned. Documentation at :
Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
Java AJP Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL HTTP/1.1 Connector on port 8080
-->
<!--
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
-->
<Connector SSLEnabled="true" keystoreFile="Path to key store file"keystorePass="the
password" maxThreads="150" port="8443" protocol="HTTP/1.1"scheme="https" secure="true" sslProtocol="TLS"/>
<!-- A "Connector" using the shared thread pool -->
<!--

    <Connector executor="tomcatThreadPool"
        port="8080" protocol="HTTP/1.1"
        connectionTimeout="20000"
        redirectPort="8443" />
-->
<!--
Define a SSL HTTP/1.1 Connector on port 8443
This connector uses the JSSE configuration, when using APR, the
connector should be using the OpenSSL style configuration
described in the APR documentation
-->
<!--
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
    maxThreads="150" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS" />
-->
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443"/>
<!--
An Engine represents the entry point (within Catalina) that processes
every request. The Engine implementation for Tomcat stand alone
analyzes the HTTP headers included with the request, and passes them
on to the appropriate Host (virtual host).
Documentation at /docs/config/engine.html
-->
<!--
You should set jvmRoute to support load-balancing via AJP ie :
    <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
-->
<Engine name="Catalina" defaultHost="localhost">
<!--
For clustering, please take a look at documentation at:
    /docs/cluster-howto.html (simple how to)
    /docs/config/cluster.html (reference documentation)
-->
<!--
    <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
-->
<!--
Use the LockOutRealm to prevent attempts to guess user passwords
via a brute-force attack
-->
<Realm className="org.apache.catalina.realm.LockOutRealm">
<!--

```

This Realm uses the UserDatabase configured in the global JNDI resources under the key "UserDatabase". Any edits that are performed against this UserDatabase are immediately available for use by the Realm.

```

-->
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"resourceName="UserDatabase"/>
</Realm>
<Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true">
<!--
SingleSignOn valve, share authentication between web applications
Documentation at: /docs/config/valve.html
-->
<!--

    <Valve className="org.apache.catalina.authenticator.SingleSignOn" />

-->
<!--
Access log processes all example.
Documentation at: /docs/config/valve.html
Note: The pattern used is equivalent to using pattern="common"
-->
<Valve className="org.apache.catalina.valves.AccessLogValve"directory="logs" prefix="localhost_access_log." suffix=".txt"
pattern="%h %l %u %t \"%r\" %s %b"/>
</Host>
</Engine>
</Service>
</Server>

```

## Appendix 1.3: Apache Directory Studio – Services.Idif

version: 1

dn: cn=Twitter,ou=services,o=idmm,dc=cdcc

objectClass: top

objectClass: Service

authenticationService: TwitterAuth

cn: Twitter

uri: https://twitter.com/

acl: 7

idType: 0

map: user[description]=description, user[name]=givenName, user[url]=labeledU

RI, user[location]=street

serviceInformation: givenName

serviceInformation: street

serviceInformation: description

serviceInformation: labeledURI

createTimestamp: 20130116134714Z

creatorsName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system

entryCSN: 20130116144714.073000Z#000000#000#000000

entryUUID:: MGUwNGU0MDQtMDdlZS00YjNiLWJiYWU0OTAyOGU3ZDczZDBi

modifiersName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system

modifyTimestamp: 20130116164125Z

## Appendice 1.4: Apache Directory Studio – Authentications.Idif

version: 1

dn: cn=CDCCAuth,ou=authentication,o=idmm,dc=cdcc

objectClass: top

objectClass: Service

authenticationService: FALSE

cn: CDCCAuth

uri: at.jku.faw.cdcc.plugin.CDCCPlugin

acl: 7

idType: 0

map: username=username, password=userPassword

serviceInformation: userPassword

serviceInformation: username

createTimestamp: 20130130080232Z

creatorsName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system

entryCSN: 20130130090232.603000Z#000000#000#000000

entryUUID:: OTMyYjRhNGYtOTE1My00MjY2LWJiMzMtOTU2MDc2YmVhZDZi

modifiersName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system

modifyTimestamp: 20130130081156Z

dn: cn=TwitterAuth,ou=authentication,o=idmm,dc=cdcc

objectClass: top

objectClass: Service

authenticationService: FALSE

cn: TwitterAuth

uri: com.twitter.plugin.TwitterPlugin

acl: 7

idType: 0

map: session[username\_or\_email]=mail, session[password]=userPassword

serviceInformation: userPassword

serviceInformation: mail

createTimestamp: 20130116134808Z

creatorsName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system

entryCSN: 20130116144808.911000Z#000000#000#000000

entryUUID:: ZmI2ODFIODUtN2VhNi00YmUyLWFhMmUtMmE1MDYwNGNkMWE5

modifiersName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system

modifyTimestamp: 20130116153627Z

## Appendix 1.5: Apache Directory Studio – exampleUser.Idif

version: 1

dn: cn=CDCC,uid=vleju,ou=users,o=idmm,dc=cdcc

objectClass: top

objectClass: User-Service

acl: 7

cn: CDCC

member: cn=CDCC,ou=services,o=idmm,dc=cdcc

createTimestamp: 20130130081217Z

creatorsName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system

entryCSN: 20130130091217.626000Z#000000#000#000000

entryUUID:: YTA1MTZlNjYtYjliOC00OGU4LTk4NWYtZmM5M2Q5YWQ4YWFK

modifiersName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system

modifyTimestamp: 20130130081226Z

dn: cn=CDCCAuth,uid=vleju,ou=users,o=idmm,dc=cdcc

objectClass: top

objectClass: User-Service

acl: 7

cn: CDCCAuth

member: cn=CDCCAuth,ou=authentication,o=idmm,dc=cdcc

username: vleju

userPassword:: amRVMkYwRjBXag==

createTimestamp: 20130130081120Z

creatorsName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system

entryCSN: 20130130091120.104000Z#000000#000#000000

entryUUID:: NDdiMGU0OGYtMDIyNi00N2NhLWJhMDUtYjRlNjI5OWE4MDMy

modifiersName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system

modifyTimestamp: 20130130083006Z

dn: cn=ldap-property,uid=vleju,ou=users,o=idmm,dc=cdcc

objectClass: top

objectClass: User-Protocol

cn: ldap-property

member: cn=LDAP,ou=protocols,o=idmm,dc=cdcc

provider: Microsoft

createTimestamp: 20120629142438Z  
creatorsName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system  
entryCSN: 20120629162438.840000Z#000000#000#000000  
entryUUID:: MGJjMGM0NTktZDc3Yy00YmI5LTgyMGQtYzVlMGZzMDM4ZTVl  
modifiersName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system  
modifyTimestamp: 20120629142504Z  
dn: cn=Twitter,uid=vleju,ou=users,o=idmm,dc=cdcc  
objectClass: top  
objectClass: User-Service  
acl: 7  
cn: Twitter  
member: cn=Twitter,ou=services,o=idmm,dc=cdcc  
createTimestamp: 20130116143049Z  
creatorsName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system  
entryCSN: 20130116153049.330000Z#000000#000#000000  
entryUUID:: Zml2ZWVmY2UtNDc2My00MGVklWIzODItZGFiZDA5ZWQ5NjVi  
dn: cn=TwitterAuth,uid=vleju,ou=users,o=idmm,dc=cdcc  
objectClass: top  
objectClass: User-Service  
acl: 7  
cn: TwitterAuth  
member: cn=TwitterAuth,ou=authentication,o=idmm,dc=cdcc  
userPassword:: S3JIWXIOazVnYIRKemNmSVB0MDQ=  
createTimestamp: 20130116135301Z  
creatorsName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system  
entryCSN: 20130116145301.409000Z#000000#000#000000  
entryUUID:: M2I4OGI3ZGIYzgyNy00MzNmLWJlNjUtOTQ0Y2IwODI0ZmUy  
modifiersName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system  
modifyTimestamp: 20130117171425Z



## Appendice 1.6: Apache Directory Studio – exampleUser.Idif

dn: uid=vleju,ou=users,o=idmm,dc=cdcc  
objectClass: top  
objectClass: person  
objectClass: organizationalPerson  
objectClass: inetOrgPerson  
objectClass: User  
cn: Vleju Mircea Boris  
idType: 0  
sn: test  
uid: vleju  
username: bmvbooris  
userPassword:: MTIzNDU=  
credit-card: 12345678910215623  
description: Test account for CDCC  
givenName: Mircea Boris Vleju  
labeledURI: www.cdcc.faw.jku.at  
mail: b.vleju@cdcc.faw.jku.at  
o: CDCC  
street: Hagenberg, Austria  
telephoneNumber: +43 732 99 70 57 13  
title: MSc  
createTimestamp: 20120629142220Z  
creatorsName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system  
entryCSN: 20120629162220.615000Z#000000#000#000000  
entryUUID:: ZDY5ZGE3M2MtM2FjMi00MWVlTk3OTUtMjIwZDE0YWwNINjhi  
modifiersName: 0.9.2342.19200300.100.1.1=admin,2.5.4.11=system  
modifyTimestamp: 20130129124850Z

## References

- [1] Cantor, S. (2005). Shibboleth Architecture. Retrieved 11 29, 2013, from Shibboleth: <http://shibboleth.internet2.edu/shibboleth-documents.html>
- [2] Chalothorn, A. (2010). Creating cloud infrastructure with OpenNebula Part1. Retrieved 10 15, 2013 from Thai Open Source: <http://www.thaiopensource.org/howto/%E0%B8%AA%E0%B8%A3%E0%B9%89%E0%B8%B2%E0%B8%87-cloud-infrastructure-%E0%B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2-opennebula-%E0%B8%95%E0%B8%AD%E0%B8%99%E0%B8%97%E0%B8%B5%E0%B9%88-1>
- [3] Cloud Computing Basic. (n.d.). Cloud Basic (pp. 3-22). Retrieved 10 17, 2013, from [www.south.cattелеcom.com/Technologies/CloudComputing/0071626948\\_c\\_hap01.pdf](http://www.south.cattелеcom.com/Technologies/CloudComputing/0071626948_c_hap01.pdf)
- [4] Creativo (2012).100 Social Networking Statistics & Facts for 2012. Retrieved 10 16, 2013, from Visual.ly: <http://visual.ly/100-social-networking-statistics-facts-2012>
- [5] Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology (n.d.). Chapter 6 Access Control. Retrieved 10 15, 2013, from Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology : [http://www.ce.kmitl.ac.th/download.php?DOWNLOAD\\_ID=1353&database=subject\\_download](http://www.ce.kmitl.ac.th/download.php?DOWNLOAD_ID=1353&database=subject_download).
- [6] Ertaul, L., Singhal, S., Saldamli, G. (2010). Security Challenges in Cloud Computing. Retrieved 10 16, 2013, from California State University: <http://www20.csueastbay.edu/directory/profiles/mcs/ertaullevent.html>
- [7] Gopalakrishnan, A. (2009). Cloud Computing Identity Management: Online security concerns are on the rise and a robust identity management is what cloud needs now. Retrieved 10 16, 2013, from Clark Atlanta University:

<http://cis.cau.edu/cms/files/CIS509-OAUTH/cloud-computing-identity-management.pdf>

- [8] Infosecurity. (2013). Identity and Access Management in the Cloud. Retrieved 10 19, 2013, from Infosecurity: <http://www.infosecurity-magazine.com/view/30544/identity-and-access-management-in-the-cloud/>
- [9] Kuyoro, S. O., Ibikunle, F., Awodele, O. (2011). Cloud Computing Security Issues and Challenges. Retrieved 10 17, 2013, from CORE: <http://core.kmi.open.ac.uk/download/pdf/1130945.pdf>.
- [10] Lencioni, J. (2009). The Benefits of Single Sign-On (SSO). Retrieved 10 15, 2013, from Gustavus Adolphus College: <http://webservices.blog.gustavus.edu/2009/09/16/the-benefits-of-single-sign-on-ss/>
- [11] Malik, A., Mohsin M. N. (2012). Security Framework for Cloud Computing Environment: A Review. Retrieved 10 16, 2013, from cisjournal: [http://cisjournal.org/journalofcomputing/archive/vol3no3/vol3no3\\_13.pdf](http://cisjournal.org/journalofcomputing/archive/vol3no3/vol3no3_13.pdf)
- [12] Martin, C. (2011). Why to use OpenNebula?. Retrieved 10 17, 2013, from OpenNebula: <http://lists.opennebula.org/pipermail/users-opennebula.org/2011-September/006313.html>
- [13] Mylife (2013). Infographic: Today's Social Media User Has Multiple Accounts. Retrieved 10 16, 2013, from hashtags.org: <http://www.hashtags.org/platforms/infographic-todays-social-media-user-has-multiple-accounts/>
- [14] OpenNebula. (2012). About the OpenNebula Technology. Retrieved 10 18, 2013, from OpenNebula: [http://opennebula.org/about:technology#what\\_are\\_its\\_benefits](http://opennebula.org/about:technology#what_are_its_benefits)
- [15] OpenNebula. (2012). Why OpenNebula?. Retrieved 10 18, 2013, from OpenNebula: <http://opennebula.org/about:why>
- [16] Pfoutz, J. (2012). The Advantages and Disadvantages of Single-Sign-On (SSO) Technology (mini-white paper). Retrieved 10 15, 2013, from Secure

Connexion: <http://secureconnexion.wordpress.com/2012/08/24/the-advantages-and-disadvantages-of-single-sign-on-sso-technology-mini-whitepaper/>

- [17] Reddy, V. K., Reddy, L.S.S. (2011). Security Architecture of Cloud Computing. Retrieved 10 19, 2013, from ijest: <http://www.ijest.info/docs/IJEST11-03-09-146.pdf>
- [18] Shibboleth (2011). Understanding Shibboleth. Retrieved 11 29,2013, from <https://wiki.shibboleth.net/confluence/display/SHIB2>
- [19] Shin, S., Kobara, K. (2010). Towards Security Cloud Storage. Retrieved 10 16, 2013, from SALSAHPC: <http://salsahpc.indiana.edu/CloudCom2010/Edemo/Towards%20Secure%20Cloud%20Storage.pdf>
- [20] Thia, J. & Thia, M. (2013). Microsoft Office 365 Single Sign-On (SSO) with Shibboleth2. Retrieved 11 29, 2013, from Microsoft: <http://www.microsoft.com/en-us/download/details.aspx?id=35464>
- Tipton, H.F., Krause M.(2004). Information Security Management Handbook. Retrieved 11 28, 2013, from <http://library.riphah.edu.pk/books%5Cmgmt%5Chrm%5CISMHandbook.pdf>
- [21] Tsyркlevich, E., Tsyркlevich, V. (2007). Single Sign On for the Internet: A Security Story. Retrieved 10 15, 2013, from blackhat: <https://www.blackhat.com/presentations/bh-usa-07/Tsyркlevich/Whitepaper/bh-usa-07-tsyркlevich-WP.pdf>
- [22] University of Guelph (n.d.). SSO Benefits: What are the benefits of Single Sign-On (SSO)?. Retrieved 10 16, 2013, from University of Guelph: <https://www.uoguelph.ca/ccs/security/internet/single-sign-sso/benefits>
- [23] Vleju, M.B. (2012). A Client-Centric ASM-Based Approach to Identity Management. In Cloud Computing. Retrieved 11 28, 2013, from Springer Link: [http://link.springer.com/chapter/10.1007%2F978-3-642-33999-8\\_5#page-1](http://link.springer.com/chapter/10.1007%2F978-3-642-33999-8_5#page-1)

- [24] Vleju, M.B. (2012). A Client-Centric Identity Management Tool for Small and Medium Enterprise Using Cloud Services. Retrieved 11 28, 2013, from [http://www.academia.edu/2725374/ISES\\_Virtual\\_Energy\\_Lab\\_an\\_Overview](http://www.academia.edu/2725374/ISES_Virtual_Energy_Lab_an_Overview)
- [25] Vleju, M.B. (2012). Interaction of the IdMM with a Client-Side Identity Management Component. Retrieved 11 29, 2013, from CDCC: [http://www.cdcc.faw.jku.at/pdf/get.php?f=bvlejutechnical\\_report\\_1.pdf&d=/public/publications](http://www.cdcc.faw.jku.at/pdf/get.php?f=bvlejutechnical_report_1.pdf&d=/public/publications).
- [26] Wikipedia. (2013). List of single sign-on implementations. Retrieved from 10 5, 2013, from [http://en.wikipedia.org/wiki/List\\_of\\_single\\_sign-on\\_implementations](http://en.wikipedia.org/wiki/List_of_single_sign-on_implementations)
- [27] Wikipedia. (2013). OpenNebula. Retrieved 10 18, 2013, from Wikipedia: <http://en.wikipedia.org/wiki/OpenNebula>
- [28] Wikipedia. (2013). SAML-based product and service. Retrieved 11 29, 2013, from Wikipedia: [http://en.wikipedia.org/wiki/SAML-based\\_products\\_and\\_services](http://en.wikipedia.org/wiki/SAML-based_products_and_services)
- [29] Wikipedia. (2013). Security Assertion Markup Language. Retrieved 11 29, 2013, from Wikipedia: [http://en.wikipedia.org/wiki/Security\\_Assertion\\_Markup\\_Language](http://en.wikipedia.org/wiki/Security_Assertion_Markup_Language)
- [30] Wikipedia. (2013). รายชื่อรหัสสถานภาพของเอชทีทีพี. Retrieved 11 29, 2013, from Wikipedia: <http://th.wikipedia.org/wiki/รายชื่อรหัสสถานภาพของเอชทีทีพี>

