A roadmap for ensuring SAML authentication using Identity server for on-premises and cloud

Triveni Kodam

Information Security, master's level (120 credits)
2019

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering
A roadmap for ensuring SAML authentication using Identity server for on-premises and cloud

Triveni Kodam (trikod-5)

Master Thesis (30 credits)
Information Security Program
Department of Computer Science, Space, Electrical Engineering,
Luleå University of Technology
Acknowledgment

This report is submitted as a master thesis for the two-year master’s program in Information Security at the Computer Science and Space department at Luleå University of Technology (LTU). It is written in close supervision with StorigoAB, Luleå.

Deserving of special mention is my industrial advisor Jim Sundqvist, who had given extensive support and initial discussions in the thesis formulation. I would also like to thank Arnold Kopping for all his assistance in the technical aspects of the thesis, especially for improving my expertise in coding. I would like to express my gratitude to StorigoAB for giving me the opportunity to develop the thesis project.

I would also like to thank my supervisor at LTU, Prof. Maung Sein for his kind assistance during the project time and Prof. John Lindström for his kind suggestions for finalizing the report. This thesis would not be what it is right now without their moral support and encouragement. I thank my fellow student Johan Lundström for his time and efforts during the thesis discussions.

Finally, I thank my husband Lavan, for his constant encouragement, love, and support. My daughter, Karthisha (fifteen-months-old), who always cheers me up and relieves stress during the documentation of this thesis. I thank my parents, family, and friends for their tremendous support at every step of my academic career.

Triveni Kodam

13, June 2018
Abstract

Cloud-based applications especially SaaS applications have become essential for startups and various sized businesses. Adapting to these web applications helps to reduce operational costs and further provide flexibility in accessing individual data of the users. On the other hand, usage of these cloud services poses security-related issues such as authentication, authorization, web application security. Additionally, if the on-premises application is moved to the cloud then the traditional Identity solutions will not work, which affects the user authentication. This thesis considers ‘Authentication’ as one of the main security issues to be addressed. Thus, a new federated Identity and Access Management (IAM) system needs to be realized, which can be used for both on-premises and cloud to authenticate users correctly and securely. To meet the described challenges within the cybersecurity domain, this thesis focuses on two aspects of IT Security: 1) SaaS application rely on IAM; 2) IAM for securely authenticating users. This thesis work addresses both these aspects in two parts. First, by developing a SaaS web application that includes an authentication module with the support of the SAML 2.0 standard protocol. Second, the use of open source WSO2 IAM server for authenticating the users securely. To implement a SaaS application, a play framework PAC4j security library is used to support SAML SSO profile for authenticating users. The profile provides functionality for the two scenarios: SAML- Service provider and SAML- Identity Provider. The developed SaaS application acts as a service provider while WSO2 identity server acts as an Identity Provider. The SAML request-response authentication workflow between these providers are verified to prove the correctness and security of user login information. The research presented in this thesis is helpful for startup companies, that are initially looking to minimize application cost that works both on-premises and cloud without compromising on the security of user’s login information.
# Table of Contents

1. Introduction .................................................................................................................. 8  
   1.1 Background and Work plan ...................................................................................... 9
   1.2 Problem statement and Research question ............................................................. 10
   1.3 Structure of the thesis ............................................................................................. 10

2. Literature Review ......................................................................................................... 11  
   2.1 Review Scope ........................................................................................................ 11
   2.2 Conceptualization of research topic ........................................................................ 11
   2.3 Literature search ................................................................................................... 11
   2.4 Literature Review analysis and synthesis ............................................................... 12
   2.5 Defining the research agenda and formulating the research question .................... 20

3. Research Methodology .................................................................................................. 21  
   3.1 Action Design Research ......................................................................................... 21  
      3.1.1 Problem Formulation ....................................................................................... 22
      3.1.2 Building, Intervening, and Evaluation (BIE) ....................................................... 23
      3.1.3 Reflection and Learning ................................................................................... 24
      3.1.4 Formalization of Learning ............................................................................... 24
   3.2 Using Action Design Research Methodology in developing this thesis ..................... 25
      ADR stage 1 – Problem Formulation ....................................................................... 25
      ADR stage 2 – BIE ................................................................................................... 28
      ADR stage 3 – Reflection and Learning ................................................................... 29
      ADR stage 4 – Formalization of Learning ................................................................. 31
   3.3 Reliability and Validity ........................................................................................... 32
      Validity ...................................................................................................................... 32
      Reliability ................................................................................................................ 33

4. Design and Development ............................................................................................... 35  
   4.1 Physical device and specification ........................................................................... 35
   4.2 SAML ..................................................................................................................... 36
   4.3 Web SSO federated Identity ............................................................................... 38
4.4 OpenSSL ........................................................................................................................................... 39
4.5 PAC4J library .................................................................................................................................... 42
4.6 WSO2 Identity server ....................................................................................................................... 42
4.7 SAML metadata ................................................................................................................................. 42
4.8 SAML message exchange .................................................................................................................. 45
5. Output Evaluation ............................................................................................................................... 48
   SAML Authentication request and response validation ................................................................. 52
6. Limitations and Future research ....................................................................................................... 58
7. Discussion and Conclusions ............................................................................................................... 59
8. References ........................................................................................................................................... 61
Appendix ................................................................................................................................................ 64
   Experiment screenshots ....................................................................................................................... 64
   Cloud Deployment ............................................................................................................................... 68
List of Tables

Table 1 Literature search criteria and databases used ................................................................. 11
Table 2 Topic wise papers found in literature search and selected .............................................. 12
Table 3 Comparative analysis of different Federated Authentication protocols .............................. 14
Table 4 Summary of Literature papers used and the reason for selection ...................................... 20
Table 5 Research gap and defining research question .................................................................. 20
Table 6 Roles and responsibilities of involved project members ...................................................... 27
Table 7 Thesis Phases duration ...................................................................................................... 27
Table 8 List of Design principles in the formalization of learning phase ......................................... 32
Table 9 The specification of the device used in this thesis ............................................................. 35
Table 10 The specification of tools and software’s used in this thesis ........................................... 36
Table 11 Summary of SP and IdP role descriptors used in metadata files ....................................... 44
Table 12 SAML Request – Response output evaluation ............................................................... 57

List of Figure

Figure 1 ADR Method: Stages and Principles, Taken from (Sein, et al., 2011) ............................... 22
Figure 2 Design of user Authentication system ............................................................................. 28
Figure 3 IT-Dominant BIE ............................................................................................................ 29
Figure 4 SAML protocol structure, Taken from (Kim, 2009) ......................................................... 38
Figure 5 Using Java Keytool to convert the certificate into .jks format ......................................... 40
Figure 6 Printing certificate into X.509 format ............................................................................ 41
Figure 7 Importing the SP certificate into WSO2 IS certificate store ........................................... 41
Figure 8 Sequence diagram of SAML communication between User browser, SP, and IdP .......... 46
Figure 9 Developed web application index page .......................................................................... 47
Figure 10 SAML Logger output .................................................................................................... 49
Figure 11 Captured SAML data while authenticating user .......................................................... 51
Glossary

AAA  Authentication, Authorization and Accounting
ADR  Action Design Research
ACS  Assertion Consumer Service
BIE  Building, Intervening, and Evaluation
FI   Federated Identity
HTTP Hyper Text Transfer Protocol
HTTPs Hyper Text Transfer Protocol Secure
IAM  Identity and Access Management
IdP  Identity Provider
LDAP Lightweight Directory Access Protocol
OIDC OpenID connect
SSL  Secure Socket Layer
SaaS Software as a Service
SP   Service Provider
SMB  Small and Medium businesses
SAML Security Assertion Markup Language
SSO  Single Sign-On
1. Introduction

Cloud computing is one of the emerging technical topics in the field of information systems. It is a style of computing that extends its deployment model capability to deliver ‘as a service’ on demand to customers over the internet. It provides different deployment models such as private, community, public and hybrid cloud. Further, it provides various delivery models such as SaaS (Software as a Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a service). The deployment and delivery models that are provided to end-users exhibit certain added features like on-demand self-service, multi-tenancy, rapid elasticity, measured services, virtualization, storage, and ubiquitous network (Subashini & Kavitha, 2011). Apart from the described features, there is a need to guarantee the security of the services they provide to the end-users or business (Fernandes, et al., 2014).

Nowadays people are moving towards the cloud-based platforms. The adoption rate is gradually increasing, due to a greater number of users accessing the provided resources and services on demand. Despite several advantages, cloud delivery models bring challenges and issues that need to be solved before the adoption of the above paradigms (Fernandes, et al., 2014). Surveys conducted by Microsoft, NIST, and CSA show that security is one of the major challenges in cloud computing (Cigoj & Blažič, 2015). Thus, many large enterprises, SMB, startup companies, and the user’s adapting to any one of the delivery models as per their business requirements. (Subashini & Kavitha, 2011) listed several security issues related to the service models (i.e. IaaS, PaaS, SaaS) in terms of data isolation, data integrity, authentication, authorization, and web application security, etc.

The thesis focuses on building a SaaS web application that includes an authentication service in relation to Data Security - ‘Identity Management (IdM)’ (Eludiora, et al., 2011) solution. According to Gartner, Identity management is a security solution for the business enablers with ‘Right individuals get access to the right resources at the right times’ within an Information technology (IT) infrastructure. In 2014, Gartner Identity and Access Management (IAM) summit predict by 2020, 70% of all business will use it to protect critical assets, up from 5% today (Thike & Oo, 2015).

The terms IdM and IAM are used interchangeably (Chong, 2004). There is a broad range of security features that an IAM solution is capable of providing access to the users in an organization (Fernandes, et al., 2014). Among them, a required level of data security can be achieved with any of such combinations (or security mechanisms) as Discretionary Access Control (DAC), Mandatory Access Control (MAC), Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) (Thike & Oo, 2015), and federation authentication (Subashini & Kavitha, 2011).

This thesis focuses on two areas within an administration and infrastructure. Identity administration area deals with managing information on user identity (ID) profiles and providing authentication service to the requested application (service provider). Identity infrastructure deals with data stores that hold user accounts and identity information, such as Active Directory or LDAP. A greater number of enterprises are using traditional identity servers, simply because of ‘cloud, cost, complexity, multiple devices, and BYOD’ (Eludiora, et al., 2011).
In some traditional on-premise applications, the administration of the sensitive data resides in each enterprise boundary with various security controls. It even relies on on-premises infrastructure services, such as AD to provide the necessary identity information. However, with the SaaS model when the user tries to access a cloud application or one of the applications of the on-premises model accessing a cloud service, the same identity usually doesn’t work. Where does it get the Identity information? A Federated identity management system (IMS) can address these issues by implementing various models of authentication and access-control for internal and web-based applications (Catuogno & Galdi, 2014). Additionally, management and auditing of user’s profiles become simplified with this centralized IS. In today’s market, there are many authentication mechanisms available for accessing the internal and cloud (SaaS) applications with the same credentials, they are OpenID Connect, OAuth, SAML and so on (Naik & Jenkins, 2016). This thesis focuses on developing a SaaS web application with built-in authentication service to the users, that can be later applied to hybrid environments. The solution is based on implementing the SAML protocol with SSO feature for identity federation and authenticating users securely (Noureddine & Bashroush, 2013). This SAML authentication service enables a smooth but secure and trustworthy communication for user’s authentication information. This developed application must be easily uploaded to a cloud provider without any major changes in the application design and data.

The goal of an IMS is to ensure that only authenticated users have access to specified resources in hybrid environments. In case of malfunction, it raises privacy issues with respect to access compliance and governance (Bradford, et al., 2014). Currently, there are various open source and proprietary IAM solutions available in the market. The selection of the right solution within the enterprise boundaries or cloud depends on various factors, such as scope, feature set, ease of deployment, scalability, security, privacy, and cost. In general, larger organizations opt for distributed IAM solutions for efficiency and effectiveness, as implementation failure results in complex system management issues for user identities (Bradford, et al., 2014). In the case of SMB and startup companies, a single software solution or identities managed through a single implementation tool can be applied across the business systems to fully support end-users within an organization. Further such implementation will also benefit large organizations in terms of reduction in application cost, management overhead and security. This thesis focuses on - How the developed browser-based SaaS web application provides SAML authentication for users on-premises and cloud.

1.1 Background and Work plan

The company ‘Storigo AB’ decided to move its resources towards the cloud. They have started developing a SaaS web application for their customers or end-users. But, the initial problem with their transition is the user identities information. If the on-premises application is moved into the cloud, the existing identity solution does not work as discussed in the introductory part. They decided to look for an open source hybrid IAM solution, that can be used for user identity management. The proposed IAM solution will be the ID server, which will be referred to in all aspects. It should have capabilities to perform authentication of users at the application level.
Integration of the on-premises developed SaaS application with the identity server securely is often challenging for a company.

The thesis study is carried out in the following two main steps:

- The first step is to build a web application using play Framework in Java, which supports authentication service and including basic features like login and log out. In this project, the authentication mechanism should support SAML. The developed web application should use a powerful security engine to secure it. The engine includes a security library to authenticate users using the Identity server and get their profiles.
- The second step is to integrate the open source identity server solution, to solve server-side issues and make sure it is consistent. Then the SAML authentication traffic generated between the service provider application and Identity server is analyzed.

### 1.2 Problem statement and Research question

Today, organizations like SMB and startup companies are increasingly realizing the use of SaaS applications for faster access with minimum cost (Christiansen & Laura, 2016). However, the customers still do not trust or is not allowed to use the cloud services, due to the security concerns. They ended up questioning the security features of the new data-processing model of cloud. The major problems include; they have no control over the data, nor the applications they are trying to authenticate for use and their identities stored in the cloud (Cigoj & Blažič, 2015). The ability to properly authenticate users to the application becomes central to IT security. This aspect of IT security is challenging for startup companies, without having significant authentication capabilities. In support of such authentication services, they are moving towards implementing Federated IAM solutions (Catuogno & Galdi, 2014). There has been limited research in the literature that focuses on startup companies (Christiansen & Laura, 2016). This thesis study attempts to fill this research gap by addressing the research question of “how these companies can integrate their SaaS web application (built in authentication module) with identity server for authenticating users securely?”.

### 1.3 Structure of the thesis

Section 2 describes the corresponding literature review. Section 3 presents the research methodology used for developing the thesis. The design and implementation of the research methodology are given in section 4. Section 5 gives an overview of observations and discussions. Then discussions and conclusions of this thesis are given further.
2. Literature Review

The goal of this thesis is to build a web application that supports SAML authentication between federated entities. The entities involved in the authentication process are a web browser, service provider, and an IDP. The methodological framework of this literature review is motivated and adapted from the work of Vom Brocke et al. (2009). Their framework is carried out in a series of 5 phases that are cyclic in a manner. The phases are as follows: definition of review scope, conceptualization of research topic, literature search, analysis of retrieved literature and defining a research agenda. A detailed explanation of how the literature review is carried out at each phase is described in the following subsections.

2.1 Review Scope

The scope of the literature review is:

- To study various standards (SAML, OAuth, Open ID) used in federated identity management systems and make a comparative analysis to justify why SAML is used in this project.
- To review possible authentication schemes between hybrid environments (on-premise web service and SaaS)
- To find ways to collect and analyze the generated SAML traffic between the parties/entities involved in the communication.

2.2 Conceptualization of research topic

To start with formalizing the research topic, a literature search is carried out to find articles using relevant key terms. The Initial key terms used for searching are as follows: SAML, Federated identity protocols, Identity management infrastructure for the cloud. A preliminary study of these articles helped in deriving more extensive key terms such as Federated IMS+ multi/hybrid environments, SAML authentication + multi-cloud environments + on-premises, X.509 certificates + SAML implementation.

2.3 Literature search

The above keywords are searched in published databases using specific search criteria shown in Table 1 for finding the relevant articles.

<table>
<thead>
<tr>
<th>Search Criteria</th>
<th>Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Includes articles published in</td>
<td>Ltu Library</td>
</tr>
<tr>
<td>- Journals, Conferences, web sources and books</td>
<td>EBSCOhost</td>
</tr>
<tr>
<td>- The time span between 2009-2017</td>
<td>Google Scholar</td>
</tr>
<tr>
<td></td>
<td>ACM digital library</td>
</tr>
<tr>
<td></td>
<td>Science Direct</td>
</tr>
</tbody>
</table>

Table 1: Literature search criteria and databases that are used
The outcome articles are studied based on title, abstract and conclusions. If the studied articles are fit enough, then forward and backward searches are conducted. These searches in the databases are based on citations and references which are used in those articles, respectively. Table 2 indicates the count of searched and used articles in the project.

<table>
<thead>
<tr>
<th>Topics</th>
<th>Number of articles found</th>
<th>Number of articles selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication protocols used in FIDM (SAML, OpenID, OAuth)</td>
<td>&gt;10</td>
<td>6</td>
</tr>
<tr>
<td>Authentication methods for hybrid environments</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Survey on Security issues of cloud computing</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Security issues while integrating on-premises applications to the cloud</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>WSO2 identity server implementations</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Action Design Research Methodology</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2 Topic wise papers found in the literature search

2.4 Literature Review analysis and synthesis

The literature articles found in above subsections are reviewed and presented as below:

Comparative analysis of different Authentication protocols used in Federated IMS:

The integration of different domain environments needs some sort of standard authentication protocols to establish secure communication between the involving parties. The three main and widely used protocols are SAML, OAuth, and OpenID Connect. There are many more protocols used (such as link, various federated authentication protocols), but the described are the most suitable protocols for this thesis study.

<table>
<thead>
<tr>
<th>Protocols</th>
<th>OpenID</th>
<th>OAuth</th>
<th>SAML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>It is a lightweight protocol, platform independent, vendor-neutral and open standard.</td>
<td>It is a scalable delegation protocol, platform independent, vendor-neutral and open standard. Its, flexibility in the implementation leads to</td>
<td>It is an XML oriented protocol, platform independent, vendor-neutral and open standard. Its, flexibility in the</td>
</tr>
<tr>
<td><strong>Authentication and Authorization</strong></td>
<td>different design models.</td>
<td>implementation leads to the different design models.</td>
<td></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-------------------------</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Scope</strong></td>
<td>It is a standard for authentication and authorization.</td>
<td>It is standard exclusively used for authorization purposes and not for authentication. But it supports indirect authentication.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>It is used to provide web SSO for simple websites, blogs, and apps. It is not intended to support high trust applications. It also provides Federated Identity management for users.</td>
<td>It allows users to share their private resources (e.g., photos, videos, contacts, apps such as Facebook, Twitter, so on) on the internet with delegated authorization to these protected resources without sharing credentials.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>It is typically used in enterprises SSO or organization scenarios. It also provides federated identity management for users.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Authentication process</strong></td>
<td>The user authenticates the web application using an Identity URL or claimed identities.</td>
<td>The OAuth WRAP authenticates the user to the application via OAuth access tokens on the user’s behalf.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The user authenticates the website, based on the trust relationship between the site, service provider and IDP.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Involved parties</strong></td>
<td>End User (EU), Relying Party (RP), Authorization Endpoint (AE), Token Endpoint (TE) and UserInfo Endpoint (UIE).</td>
<td>Consumer/Client (OC), Authorization Server (AS), Resource Owner (RO), and Resource Server (RS).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SP or IDP initiation.</td>
<td>User, IDP, Service Provider (SP).</td>
<td></td>
</tr>
<tr>
<td><strong>Protocols is used</strong></td>
<td>HTTP, REST, JSON</td>
<td>HTTP, REST, JSON</td>
<td>XML, HTTP, SOAP</td>
</tr>
<tr>
<td><strong>Token Format</strong></td>
<td>It facilitates two types of tokens: Access token and ID token. The ID token is a JWT (JSON web token) that contains information about the authenticated user.</td>
<td>JWT (JSON web token), XML. The Authorization tokens state the right of the client application to access protected resources on the authorization server.</td>
<td>XML signature -X.509. SAML tokens are always signed with a private key. This establishes a trusted relationship between IdP and SP.</td>
</tr>
<tr>
<td><strong>Play framework application support (Java)</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Degree of security</td>
<td>Less Secure</td>
<td>Inherently secure</td>
<td>Highly secure</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>-------------------</td>
<td>---------------</td>
</tr>
</tbody>
</table>

Table 3 Comparative analysis of different Federated Authentication protocols

Based on *Table 3*, comparative analysis of each protocol from its underlying characteristics, design aspects, etc., helps in selecting a suitable protocol for this project implementation. SAML is considered to be the best suitable authentication standard protocol because of its inherent architectural design to support certificate-based trusted authentication between the enterprise-to-enterprise or cloud-to-enterprise. Their communication based on the SAML assertions in metadata are highly secure. The implementation details are clearly given in *Section 4*.

**Authentication schemes between hybrid environments:**

*Indu et al. (2017)* proposed an integrated IMS for cloud web services to provide improved security. They implemented the proposed system, by combining the SAML technology with a token-based for fine-grained authentication. The communication between the IDP’s, service providers, and web providers are established with the help of metadata files. These metadata files contain SAML attributes (or certificates) for secure authentication between the source and destination. When the user successfully authenticates to cloud web services through SSO SAML authentication, the IDP generates access tokens. These copy of access tokens are stored within the IDP and service provider, sequentially. Once these access tokens are used for any web service request from cloud servers, they are removed from the service and IDP’s token store. This way, the same token is prohibited from reuse. They provided a hashing algorithm for the generation of these access tokens in a hexadecimal random string by the IDP. The string considers the parameters like the identity of the user, service provider entity id, current time, and token expiry time for generation of the access token. They implemented this token-based authentication mechanism through SAML protocol metadata files. They modified the existing IdP and SP metadata files with the cloud service provider to verify the access token and responding to the service providers respectively. They provided a performance evaluation based on time complexity (Total time taken for execution of token generation algorithm), comparison of proposed SAML based authentication model with existing authentication models and security analysis of adapted SAML protocol in terms of insider and outsider cyber-attacks.

*Andronache et al. (2011)* achieve implementation of Web Single Sign-On (SSO) authentication using the standard framework SAML 2.0. To implement the standard, they have used an open-source SimpleSAMLphp library. Their library is written in PHP and offers support for integrated web-based PHP application into a federation. The mechanism of SSO permits user’s authentication and authorization access to computers/application resources only once, without the need for multiple passwords. Whereas, the Web SSO features using the SAML allows users to authenticate once between multiple web applications using established trust relationship, not necessarily within the same security domain. Their experimental setup includes the functionality of two federation scenarios: SAML as a Service provider and SAML as an IDP. These SimpleSAMLphp as a service provider connects to the IdP to authenticate users. These implementations include classes of SimpleSAMLphp using API. The API provides basic functionality, such as: check if the user is authenticated, require authentication, login, logout, get
user attributes, get URLs for login and log out. Thus, the web application has no knowledge of the IdP and allows no change in the web application code. These SimpleSAMLphp as an IDP can accept connections from various service providers (SP) using trusted certificates. On receiving the user request from any of the SP, it validates the credentials against various authentication modules, such as LDAP, CAS, SQL, OpenID, Facebook, and Twitter. After login, the session information is stored by the IdP. This experimental setup is carried on two local computers, one as an SP and other as an IdP. On the SP side, a sample page was used for authentication. For configuration, the required application modules (Apache web server, PHP) are loaded for SP to connect to the IdP, additionally includes metadata for login/logout URLs and IdP certificate. On the IdP side, they used a SQL authentication module and stores the credentials in a table format. For configuration, includes the authentication method and metadata of SP. The test was successful, once the user access one of the applications, he/she directly goes to the second application using the same IP. These methods only work within the same browser window.

Rabinovich et al. (2015) propose a solution for better Authentication, Authorization, and Accounting (AAA) in identity federation. They list some simple changes in the vendor products can significantly improve application level AAA in environments with shared Assertion Consumer Service (ACS). The ACS is the main component used for implementing SAML standard for service providers. On SP-side, changes in the SAML metadata includes, ACS acts as a SAML protocol proxy instead of application SAML actual entity names. This information about the application is conveyed to IdP in the request form. The IdP responses to the application entity without knowing on whose behalf those federations requests are made. The author lacks in providing a detail explanation of how application-level granularity is achieved using different vendor products. But a clear idea is given on how SAML standard 2.0 can be used for web SSO authentication and importance of ACS.

**Security issues in Integrating enterprise infrastructure with cloud computing:**

Pfaff et al. (2014) present some of the security challenges that encounter while integrating traditional enterprise infrastructure with cloud computing. They focused mainly on user repositories, authentication, and authorization systems within the enterprise security boundary and their adoption to cloud offerings. The author referred to cloud offerings as XaaS (w.r.t IaaS, PaaS, SaaS). They analyzed these integration issues are crucial and need to take an extra care when considering cloud offerings for enterprise IT. This analysis presents a few integration challenges, red-flag pitfalls and identifies best practice/mitigation approaches for application owners, architects, service owners, and cloud providers. These analyses are as follows:

1. XaaS requires on-boarding of enterprise users
2. XaaS asks for initial authentication credentials
3. XaaS depends on backchannel interactions for SSO
4. XaaS pulls properties of logged-in users
5. XaaS cares for enterprise-initiated login only
6. XaaS depends on backchannel interactions for authorization decision making
7. XaaS asks for data of not logged-in users
8. XaaS depends on backchannel interactions for other use cases e.g. auditing
9. XaaS uses proprietary Web SSO protocols
10. XaaS does not provide public-facing endpoints
This thesis study considers points 1,2,3,5 and 9 related to authentication repositories/systems/protocols, however, other points are directly or indirectly related to one another. Point 1, focuses on user identity synchronization issues. The on-boarding enterprise users with a cloud provider may lead to duplication of identity information. To avoid this issue; make sure that corporate repositories are not accessible for XaaS components; either by employing SSO process; use of security tokens for verification of identities within the enterprise; use of plug-in interfaces to deploy service clients. Point 2, focuses on authenticating users to XaaS, whereas validation of credentials/identity occurs at enterprise infrastructure. These kinds of validation may increase the risk of identity theft. To solve this issue, XaaS and enterprise providers must adapt to externalize authentication with the help of federated IdM protocols such as SAML, WS-Federation, and OAuth/OpenID Connect. With the adoption of such protocols, the communicating parties must be in synchronization. Point 3, focuses on direct exchanges of SAML user profiles/OpenID UserInfo service between XaaS and enterprise infrastructure. This kind of setup makes certain endpoints in the enterprise security infrastructure public-facing. To reduce exposure to on-premises services, improve SSO user experience through frontend exchanges. Point 5, focuses on on-premises IdP-initiated logins for accessing any XaaS offerings. These make the users bookmark the XaaS resources login within the enterprise boundary and results in unsatisfactory user experience. To solve this issue, XaaS SP-initiated logins are required to be employed. Point 9 focuses on, using proprietary web SSO protocols for authenticating users. These type of traditional web SSO products suffers from limited authentication requests and non-versatile security tokens. To avoid these issues, authors suggest using standards-based web SSO protocols (like SAML/OpenID/OAuth).

Armando et al. (2013) report various authentication flaws in the emerging SSO protocols, namely SAML SSO and OpenID. They evidence a Cross-site scripting attack in SAML-based SSO for Google Apps and in SSO for Novell Access Manager v.3.1. Apart from the demonstration and mitigating the attack, they also described general solutions for the flaws in both protocol implementation. This project study focuses on protocol SAML SSO. Hence, discussion about OpenID is out of scope. If readers are interested, can go through the complete article.

This article focuses on the use of implementing a SAML 2.0 web browser SSO profile. The profile is SP-initiated SSO with Redirect/POST bindings towards IdP for authenticating clients. To demonstrate the authentication flaw, the involved parties in the attack are; a client©, an honest IdP (IDP), an honest SP (sp) and a malicious SP (i). C initiates a protocol call by requesting the resources URI i at SP i. Now SP i acts as a C and request a different resource at sp. The sp starts an authentication request and provide a response to i. Now i maliciously reply to c by sending an AuthReq (id, sp) and URI of AuthReq (id i, i) in form of HTTP redirect response to IDP. Finally, the attack was successful because the client (browser) has no knowledge of the used SAML protocol in the communication. No idea of how to verify, whether the AuthReq and SAML assertions are related to the initial request. Thus, an attack can be exploited in a number of ways, as below:

- Delivery of an unrequested resource.
- The launching pad for Cross-site Request Forgery (CSRF) attacks.
• The launching pad for Cross-site Scripting (XSS) attacks.

The above attack was implemented for Google Apps, Novell Access Manager and SimpleSAMLphp. If the above-identified flaws can be fixed, then these attacks would no longer be exploitable. Apart from these flaws, the paper also discusses several possible measures to tackle the flaws that can be exploitable. Some of these flaws are given below:

- **Signing the Authentication Request**: If the AuthReq is signed, it limits the attacker's ability to alter the content of sensitive fields within the message.
- **Enforcing a single session**: Need to make sure that the SP is interacting with the same C throughout the established session.
- **Cookies**: Enforcing binding on connections by establishing a session cookie.
- **Self-signed Client certificates**: An effective way to establish trust between the communicating SP to IdP. The certificates verify the trust, before allowing the parties to communicate.

These measures will be useful for this thesis study and needs to be followed to improvise the security in communication between C, SP, and IdP.

The literature studies discussed above are summarized in the below Table 4.

<table>
<thead>
<tr>
<th>No.</th>
<th>Reference Title</th>
<th>Area of Interest</th>
<th>Reason for selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Web single sign-on implementation using the SimpleSAMLphp application (Andronache &amp; Nisipasiu, 2011)</td>
<td>An authentication method for SAML implementation</td>
<td>Selected. Web SSO authentication was implemented using the SAML protocol and language library used was SimpleSAMLphp. It gives an idea of how SAML protocol is used for establishing authentication between User, SP, and IdP within the same browser window.</td>
</tr>
<tr>
<td>02</td>
<td>Investigating information systems with action research (Baskerville, 1999)</td>
<td>Thesis research methodology</td>
<td>Not selected. Only used as a reference to differentiate between AR and ADR methodology.</td>
</tr>
<tr>
<td>04</td>
<td>Centralized end-to-end identity and access management and ERP systems: A multi-case analysis using the Technology Organization Environment framework (Bradford, et al., 2014)</td>
<td>Centralized IAM systems</td>
<td>Selected. Gives a clear idea of how the case study research methodology is selected and how to formulate research questions based on theoretical and technical bases. It lists out the various Technological, organizational, and environmental factors that affect CIAM implementation between educational institutes.</td>
</tr>
<tr>
<td>05</td>
<td>Achieving interoperability between federated identity management systems: A case</td>
<td>Federated IAM systems</td>
<td>Selected. Discuss some of the concepts and techniques that are commonly used in user authentication</td>
</tr>
<tr>
<td>No.</td>
<td>Title</td>
<td>Selected.</td>
<td>Abstract</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------------------------------------------------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>06</td>
<td>IAM, Microsoft Corporation (Chong, 2004)</td>
<td>IAM</td>
<td>IAM Selected. Detail description of IAM and its features.</td>
</tr>
<tr>
<td>07</td>
<td>Identity as a Service on the Journey to the Cloud</td>
<td>SaaS application and IAM</td>
<td>Selected. Gives a brief idea on how SaaS applications meet the challenges of IAM from large organizations to startup companies.</td>
</tr>
<tr>
<td>08</td>
<td>An authentication and authorization solution for a multiplatform cloud environment (Cigoj &amp; Blažič, 2015)</td>
<td>Authentication method</td>
<td>Selected. Develops an AA solution based on SSO approach in two platforms (VMware, OpenStack) for cloud users and administrators. Briefs about security problems of user authentication in cloud services.</td>
</tr>
<tr>
<td>09</td>
<td>A user identity management protocol for cloud computing paradigm (Eludiora, et al., 2011)</td>
<td>Authentication, cloud computing.</td>
<td>Selected. Identity management protocol developed to achieve security objectives in cloud computing environments. It addresses various research questions on each topic for authenticating users and its security.</td>
</tr>
<tr>
<td>10</td>
<td>Security issues in cloud environments: a survey (Fernandes, et al., 2014)</td>
<td>Security issues in cloud environments.</td>
<td>Selected. It briefs about each topic in cloud environments (includes delivery and deployment models) features and their security issues. Shows survey statistical data regarding security. It also includes several open research topics of the literature on this subject.</td>
</tr>
<tr>
<td>11</td>
<td>Design science in information systems research (Hevner, et al., 2004)</td>
<td>Thesis research methodology</td>
<td>Selected. Study about design science and research methodology to distinguish between ADR and DSRM.</td>
</tr>
<tr>
<td>12</td>
<td>Encrypted token-based authentication with adapted SAML technology for cloud web services (Indu, et al., 2017)</td>
<td>SAML implementation</td>
<td>Selected. Detail description of how SAML technology is used for implementing token-based for fine-grained authentication. It even details about SAML authentication process, certificates, and metadata files.</td>
</tr>
<tr>
<td>Page</td>
<td>Title</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Securing digital identities in the cloud by selecting an opposite Federated Identity Management from SAML, OAuth and OpenID Connect (Naik &amp; Jenkins, 2017)</td>
<td>Selected. Used for comparing different federated protocols (SAML, OpenID, OAuth).</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Applying action design research (ADR) to develop concept generation and selection methods (Pettersson &amp; Lundberg, 2016)</td>
<td>Thesis research methodology Selected. About used Action Design Methodology for this thesis study.</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Integrating enterprise security infrastructure with cloud computing (Pfaff &amp; Ries, 2014)</td>
<td>On-premises &amp; cloud integration issues Selected. Discusses various integrating issues between on-premises and cloud applications. The discussion points are related to security and general issues.</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>A survey on security issues in service delivery models of cloud computing (Subashini &amp; Kavitha, 2011)</td>
<td>Security issues in cloud delivery models Selected. A survey on different security risks that pose a threat to the cloud is presented and more specific to the different security issues for service delivery models of a cloud computing system is also listed.</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Action design research (Sein, et al., 2011)</td>
<td>Thesis research methodology Selected. About used Action Design Methodology for this thesis study.</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Ensuring Fine-Grained Authorized Access Control for Healthcare Applications on Cloud Provisioned Platform (Thike &amp; Oo, 2015)</td>
<td>WSO2 IS and access controls Selected. Gives an overview of various access controls used, WSO2 identity server and implementation of access control for healthcare application.</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Protecting outsourced data in cloud computing through access management (Wang, et al., 2014)</td>
<td>WSO2 Identity Server Selected. Shows implementation of access control model using WSO2 identity server for outsourced data.</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>An authentication flaw in browser-based single sign-on protocols: Impact and remediations (Armando, et al., 2013)</td>
<td>SAML SSO security issues Selected. Gives a detail description of various security issues that happen in Browser-based SSO implementation. These solutions are helpful for thesis and most of those points were taken</td>
<td></td>
</tr>
</tbody>
</table>
into consideration during the development phase.

Table 4 Summary of Literature papers used and the reason for selection


2.5 Defining the research agenda and formulating the research question

The below-defined literature review analysis highlights some of the important points in building a SAML authentication between On-Premises and cloud environments. This study includes an analysis of various authentication protocols that can be used, in case of security issues and authentication schemes for cloud environments. But none of them focuses on how the on-premises web login page using SAML authentication extends the same flexibility in the cloud. There are different applications and languages used in building the sample login page, SP and IdP. The different languages used by the developers are SimpleSAMLphp library, .NET, etc. This thesis study is carried out in a start-up company, known as StorigoAB, where their web applications are built in Java language using play framework. Therefore, to maintain the application compatibility this study uses JAVA play framework to build a web login page. PAC4J library is a powerful java security engine supported by the play web application is chosen to implement the SAML authentication between the SP and IdP. An open source WSO2 identity server is chosen as an identity server to grant access to the web application. A research question is defined in Table 5, which forms a basis to fulfill the identified research gap.

Table 5 Research gap and definition of the research question

<table>
<thead>
<tr>
<th>Research gap</th>
<th>Research question</th>
</tr>
</thead>
<tbody>
<tr>
<td>- An on-premises web application that supports SAML authentication between the service provider and identity provider through the use of browser exists. But those don’t support deploying the same application to a cloud platform. If needs, then require changes to the internal coding structure. Additionally, providing data access security to the users/customers is one of the major concerns.</td>
<td><strong>How can the developing SaaS web application login page securely interact with identity server to authenticate users?</strong></td>
</tr>
<tr>
<td>- The solutions so far provided for using identity servers (as IdP) as internally or externally for SMB’s are not cost effective.</td>
<td></td>
</tr>
</tbody>
</table>

Table 5 Research gap and definition of the research question
3. Research Methodology

After studying the process of different qualitative research methodologies and my research problem and question it seemed like Design Science Research Methodology (DSRM) and Action Design Research (ADR) were potential research methodologies to use. The DSRM is suitable for building and evaluating of an artifact. Where the artifact evolves from an understanding of an unsolved problem according to the business need (Hevner, et al., 2004). On the other hand, ADR focuses on cycles of building, intervention, and evaluation (BIE) to develop/solve an IT-artifact for use within an organization. The ADR deals with two main situations encountered within an organization as follows (Sein, et al., 2011):

1) Addressing the problem by intervening and evaluating.
2) Constructing and evaluating an IT artifact that addresses the occurred problem.

This thesis requires a methodology that allows performing the following tasks:

1) Literature study about SAML standard, Pac4j security library used in play application and Identity server. Then from gained knowledge and understanding design an IT artifact that resembles the problem.
2) Setup an existing Identity server (Wso2) according to the environment that I choose to work upon. Setup includes (Installing, configuration and then re-evaluating).
3) Build a SaaS web application using Pac4j security library that includes SAML configuration to interact with the identity server (using java language). Then evaluate it.
4) Intervene in both (Developed application, identity server) and evaluate how it works together.
5) Finally verify the generated SAML Traffic, to prove its integrity and security strength of SAML messages while authenticating the users.
6) Upload the SaaS application to an available open source Cloud provider.

All the above steps in this study require clear guidance on designing, building, intervening and evaluation with continuous learning to solve identified research gap. Thus, ADR is the appropriate method and therefore chosen for this thesis as it involves continuous BIE cycles.

3.1 Action Design Research

ADR address research problem through four main stages aligned with seven principles. Thus, the first stage of problem formulation is addressed by the principles of “practice-inspired research” and “theory-ingrained artifact”, the second stage of BIE is addressed by the principles of “reciprocal shaping”, “mutually influential roles” and “authentic and concurrent evaluation”. The third stage of reflection and learning is addressed by the principle of “guided emergence”, and the fourth stage, formalization of learning, is addressed by the principle of “generalized outcomes” (Figure 1; Sein et al., 2011).
3.1.1 Problem Formulation

The first stage in ADR is Problem Formulation. It helps in identifying the primary problem of the organization that needs a change in their internal structure or services they are providing to end-users. It starts by investigating the nature of the problem within the organizational domain. The author at this stage focuses on a few questions that need to be addressed before entering the next stage of research methodology. The underlying questions to support the ‘Formulation of the initial research question’ are as follows:

- What is the initial research scope?
- What are the roles/responsibilities of members involved during the research?
- Does formulated research question fit to solve the identified problem?
- Do involved members agree upon formulated research question?
- How to secure a long-term commitment from the involved members throughout the project?

While answering the above questions, there are two principles to adhere to this stage.
**Practice- Inspired research:** The author develops knowledge by reviewing prior technical problems. The formulated research question should bound to solve a technical problem or a set of problems (rather than theoretical problems) within an organizational domain. The derived solution forms as a basis for creating knowledge opportunities (i.e., technical) within the chosen field of study.

**Theory-Ingrained Artifact:** The use of prior theories are linked to create and evaluate the problems. These theories in addition also help in identifying possible solutions. The identified theory infused for solving the problem will be recognized as a real-life research study.

Adhering to the above principles, formulation of the initial research question is based on contributed theoretical bases and prior technical knowledge. In principle, searching standard databases for existing theories and technological solutions in this field and then followed by a deep study. The literature study on the identified problem will be our baseline for building, intervene and evaluation cycles.

### 3.1.2 Building, Intervening, and Evaluation (BIE)

The second stage in ADR is Building, Intervening, and Evaluation. It involves a series of cycles to build the system, intervene in the target organization and evaluate. The BIE is based on an identified artifact from stage-1 of ADR. After completion of each cycle, feedback from the project members is involved. The outcome is turned out to be the actual design artifact intended for the organization.

Two forms of BIE follow the research design; they are IT-Dominant BIE and Organization-Dominant BIE.

**IT-Dominant BIE:** This approach is best suited for creating an innovative technological design with the help of involved project members. The end user involvement is completely at the end of the project release. The project will go through a series of two versions, one is alpha and other is beta. At the alpha version, the author has an opportunity to build the desired system with gained theoretical and technical knowledge. This includes knowledge, assumptions, and expectations of the other team members. Intervention and evaluation further repeated to follow this initial design of the artifact. At the beta version, a more mature version of the system is released. The end users have an opportunity to test it or use it, based on the feedback provided. This provided feedback results in the exit of the team members or spawn to additional BIE cycles.

**Organization-Dominant BIE:** This approach is best suited for improving the existing technology design and user’s problems. Apart from IT-Dominant BIE, the end-user involvement is from starting version to the end version. The author will improve the design based on more organizational existing idea’s and assumptions. The user expectation of the artifact stops when the organization decides to adopt it or spawn for additional cycles.

This stage has three design principles to adhere to:
**Reciprocal Shaping**: The two domains that are mutually influenced in this principle are IT artifact and organizational context. The increased understanding of the organizational context will influence the artifact, while the result of the artifact influences the practices in the organizational context.

**Mutually Influential roles**: This principle influences mutual learning among the participating project members such as stakeholders, users, system administrators, researchers, board members and so on. The knowledge from all the participating members must be collected and evaluated to provide an expected solution.

**Authentic and Concurrent Evaluation**: This principle highlights the importance of evaluation. It says that evaluation should be a continuous on-going action in the research process (following building phase). Evaluation at each stage may vary with the format, and it may not be the final solution. For example, implementing an initial prototype code and testing its working functionality can be termed as an evaluation.

### 3.1.3 Reflection and Learning

The third stage in ADR is Reflection and Learning. It is a continuous stage and works parallel to the above two stages as shown in *Figure 1*. Learning at each phase of the project (i.e., use case creation to final solution) is reflected in the form of results. The author provides a continuous reflection on the problem, chosen theories, selected technical sources, design, redesign, and the outcome adhering to the above 1 to 5 principles. This involved unexpected errors or changes (from feedback) that occur at each phase adds knowledge and value to the time spent on the research.

This stage has one principle to adhere to:

**Guided Emergence**: It focuses on the interaction between the initial design and principles adhere by the researchers. It also involves reshaping the outcome according to organizational use, feedback from members and continuous evaluation results. Thus, the researchers need to maintain a balance between those and should not deviate to conflict with the original theories until the desired output is achieved.

### 3.1.4 Formalization of Learning

The fourth stage in ADR is Formalization of Learning. The learning through all the stages must be formalized according to the organizational structure. The final solution to the problem is specified as design principles and contribution to theory. These results must be documented and formalized for further analysis and discussions.

This stage has one principle to adhere to:

**Generalized Outcomes**: Generalizing the outcome is one of the most challenging tasks, as it includes the implementation of IT artifact (problem) within organizational settings, which results
in a change of organizational structure. The solution to address the problem can be further generalized into a wider problem or set of problems. It can be achieved in three levels:

- Generalizing the problem instance
- Generalizing the solution instance
- Deriving the design principles from the research outcomes.

3.2 Using Action Design Research Methodology in developing this thesis

ADR stage 1 – Problem Formulation

The purpose of this stage is to identify the initial research scope of the project. This thesis initially started with a simple problem statement within the company context that needs a change. The statement is:

‘How to build a SaaS application, that allows users to securely redirect their authentication request to an external federated identity server. The identity server must have the capability to store the company user details and seamlessly interact with the developed SaaS application. All these implementations should take place through a browser for easy user interaction with their applications’.

This statement was taken as a baseline for formulating the problem from a researcher point-of-view. It started with finding the relative literature database based on a few keywords. The gathered articles that have insights into such problems were studied, both from theoretical and technical perspectives. This base knowledge helped in finding the research gap. Then it was decided to choose the identified research gap as a thesis subject since it poses a valid and relevant IT/IS problem.

The next step of this stage is formulating a research question from the identified gaps. Before making a precise research question, there were some technical and non-technical scope points that need to be addressed:

- The company has their working projects built in Play-platform with Java as a source of coding. The SaaS application that we are going to build for authenticating users must be compatible with the platform and language used. Therefore, we need to build a web application using play framework with JAVA.
- The identity server they have support only authenticating user’s on-premises. Therefore, the developing SaaS application needs an identity server, that has extended features to support and manage identities securely for internal and SaaS services without any issues. Due to the cost constraint, they ended up in looking at the openly available IAM solution WSO2 Identity server.
- The web application that we are going to build must support the following functionality:
  a. The developing SaaS web app must include a secure authentication mechanism to interact with the identity server:
The chosen mechanism must interact with an external identity server for securely authenticating the users with same credentials on-premises (direct clients) and even on the cloud (indirect clients). For this purpose, we decided to use the SAML standard protocol.

b. Application Logout:

Logout the user from the application.

c. Additionally, extending to support fine-grained authorization in the future (Not in the scope of this thesis).

- The company decided to implement this solution in a testbed environment, without disturbing the actual project resources. Therefore, no end-users involvement in this thesis study.
- The project members are responsible to involve at each module testing and providing individual feedbacks.
- The final output evaluation presented in this thesis will be the generated SAML traffic. The traffic needs to be verified, by looking at the generated attributes value and cross verify with the value of the actual attribute given in XML metadata files of the application. This proves the application correctness and security.
- Check if this application can be upload to a cloud provider (not including any output evaluation).

The list of required hardware and software resources for building the solutions is listed in the device specification section 4.1.

The next step of this stage is assigning roles and responsibilities for the members involved in this research study. The below Table 9 lists the people involved and their roles.

<table>
<thead>
<tr>
<th>Roles</th>
<th>Responsible For</th>
<th>Personal Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADR researcher</td>
<td>Research the problem from a scientific perspective. Involving ADR methodology from problem formulation, designing, building, intervening in organizational settings, evaluating the output and finally summarizing the study the learning and reflecting.</td>
<td>Triveni Kodam, LTU student</td>
</tr>
<tr>
<td>Project Developer</td>
<td>Helping in coding the application and provide feedback for the action taken</td>
<td>Arnold Kopping, Developer</td>
</tr>
<tr>
<td>Independent Reviewer</td>
<td>Provides necessary comments/ feedback from the technical and non-technical point of view. Project change approval if</td>
<td>Jim Sundqvist, CEO</td>
</tr>
</tbody>
</table>
The next steps of the stage include research question approval and long-term commitment from the above members. Once the research question was formularized an initial draft was sent to the Independent Reviewer for approval. We had a general meeting to discuss the draft. There was given feedback with some modifications from the technical side. Initially, we decided to build an application, that not only supports authentication but also to include fine-grained authorization. When we listed out the project phases and duration of each phase made us proceed with authentication only, due to limited time constraint. The project phases over a period of six months from January to June 2018 are listed in below Table 10.

<table>
<thead>
<tr>
<th>Phase No.</th>
<th>ADR phases with principles</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 1: Problem Formulation</td>
<td>1. Determine the scope of the project and preparing the initial draft of the given problem statement</td>
<td>2 weeks</td>
</tr>
<tr>
<td></td>
<td>2. Literature review and formulating the research question</td>
<td>2 weeks</td>
</tr>
<tr>
<td></td>
<td>3. Obtaining the project commitment from company and approval from the university</td>
<td>2 weeks</td>
</tr>
<tr>
<td>Phase 2: BIE</td>
<td>4. Application prototype design</td>
<td>2 weeks</td>
</tr>
<tr>
<td></td>
<td>5. Implementation of the design</td>
<td>4 weeks</td>
</tr>
<tr>
<td></td>
<td>6. Installation and configuration of the required software’s for the thesis study</td>
<td>4 weeks</td>
</tr>
<tr>
<td></td>
<td>7. Testing the solution</td>
<td>3 weeks</td>
</tr>
<tr>
<td>Phase 3: Reflection and Learning</td>
<td>8. Evaluating the output</td>
<td>Occurred throughout the thesis study</td>
</tr>
<tr>
<td>Phase 4: Formalization of Learning</td>
<td>9. Results / feedback / Lessons Learned /Documenting the changes</td>
<td>4 weeks</td>
</tr>
</tbody>
</table>
ADR stage 2 – BIE

The purpose of this second stage is to build, intervene an evaluate the artifact to solve the problem. The plan is to build a web application on-premises that includes a security library for implementing a SAML authentication module and to return user profile information once authenticated. Then the company is looking to integrate it with a proposed open source Wso2 identity server (Xu & Yang, 2015) for storing the user identities and authenticating the users, respectively. It is successful in many healthcare companies and data outsourcing (Thike & Oo, 2015) (Wang, et al., 2014). These works of literature provide enough information to move forward in this business direction. Additionally, suggests that it can be applied to different application domains as per their data and requested resources. This experimental output analysis will be carried out by validating the user identity information over SAML communication with the metadata files information in the application. SAML is an open standard for exchanging authentication and authorization data between the parties. The involving parties are the identity provider (WSO2) and a service provider (Storigo.se). This below plan of action will be done together with the company practitioners by an intervening approach as shown in Figure 4.

![Diagram of user authentication system](image)

**Figure 2 Design of user Authentication system**

The more detailed explanation about building the application, intervene in company settings and evaluating the output is discussed in Section 4 and Section 5.

As explained, the author at this stage will continue to work on any one of the BIE forms. That helps to achieve the desired results of this thesis study. As stated in the scope of the project, there
is no end-user involvement until the project completes. Looking into the BIE forms, we choose IT-Dominant BIE as our model. At the alpha stage, we started with initial design in a testbed environment. Initial feedback at this stage for each module will be made by the involved project members.

![IT-Dominant BIE Diagram](image)

*Figure 3 IT-Dominant BIE*

When we started with the initial design according to the theory and technical knowledge gained, we discovered some issues that forced to run another cycle of BIE. A brief description of the issues are as follows:

I. Undiscovered bugs while coding and building the web application.
II. Incompatibility issues with the initial operating system used.
III. Issues with initial configuring of Identity server.
IV. Communication issues between the SP and IdP include:
   - Certificates of format issues.
   - Misconfigurations in building the SAML XML metadata files.

These issues will be discussed in detail in the next phase of ADR methodology.

**ADR stage 3 – Reflection and Learning**

Throughout the thesis, the below reflections were recorded.

<table>
<thead>
<tr>
<th>No</th>
<th>Reflection</th>
<th>Reflection Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-01</td>
<td>Interaction with Stakeholder</td>
<td>Each stage of the thesis progress the stakeholders were involved. Start with problem formulation, identify and selection of key resources for the implementation and finally to test and evaluate the expected artifact. Each of the application module built was offered to the practitioner.</td>
</tr>
</tbody>
</table>
Whether the action was successful or unsuccessful, feedback or comments were discussed, and changes are made accordingly. As each task progresses within the project, evaluating the output is a continuous process. With such continuous ongoing cycles, unexpected error or changes can be made easily with added value to the time spent on the project.

<table>
<thead>
<tr>
<th>R-02</th>
<th>Technical challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The thesis faced a lot of technical problems starting from building a testbed environment, with application coding, establishing secure communication between the parties and to implement the SAML protocol. The below points list the detail description of each problem:</td>
</tr>
<tr>
<td></td>
<td>• We started with initial application coding using Eclipse IDE on my local windows computer, as it is compatible with any of the operating systems. Then decide to build a testbed environment (a replica to the actual company environment), that helps us to load, run and save all the application data in one place. After installing the windows software into the VirtualBox, we faced an issue with the domain name configuration, as the rest of the company projects runs on a centralized domain name. The application we are going to build needs to be in the domain to communicate with the WSO2 identity server for authenticating the users. We discarded the entire setup and built another virtual box with Ubuntu OS. As Ubuntu has a special system daemon called ‘avahi’. This daemon makes the system to run on a local domain name.</td>
</tr>
</tbody>
</table>
|      | • We decided to use self-signed certificates for the providers to trust each other. Faced issues while importing the service provider certificate keys to the IdP client-trust store. As we were not aware of the supported key format must be in ‘.JKS’.
|      | • Initial misconfigurations with the SP and IdP metadata file generation. When the application runs, doesn’t redirect to the requested identity provider. Making changes to the configurations in the identity provider, a successful IdP login page appeared. |

<table>
<thead>
<tr>
<th>R-03</th>
<th>Non-Technical Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Non-Technical challenges faced in the project is the Cost factor. As mentioned, it’s a startup company and has a very low budget. So, opting for a paid software only for testing is difficult. That’s the main motive for selecting open source software’s and making it customized to work according to the organizational settings.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R-04</th>
<th>Backup Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>To avoid data loss, we planned to upload the code to the GitHub. Each update to the application code is updated to the GitHub user profile. We also used it as a medium for communication for submitting the application code to the project head. They evaluate the output and gives me feedback.</td>
</tr>
</tbody>
</table>
The thesis reflection is summarized as follows:

- **Adherence to principles**: The main goal of this thesis is to build a SaaS application for the startup company with a secure authentication mechanism. It must support authenticating users both on-premises and cloud. The solution must be secure, strong, simple, and effective for the company. The results of the project show that the research principles are applied at each stage until the completion stage.

- **Design and Structure**: The system design was simple and cost-effective for the company. However, the system was implemented in a testbed environment without any end-user involvement. The actual user test reveals the effectiveness of the solution.

- **Scalability of the application**: The web application was developed in play framework. This web framework is so flexible and easily adaptable with new modules implementation (for instance authorization module) without affecting the old module (for instance authentication).

- **Platform Independent**: The application was developed in an IDE environment, it is suitable for most of the platforms. The chosen identity server is platform independent application.

- **Security of the application**: The integrity of the application is well tested, by taking into consideration each attribute or elements in a SAML request-response communication. The security of the application lies with the chosen security library Pac4j to implement SAML.

**ADR stage 4 – Formalization of Learning**

The problem or artifact was identified, and the solution was achieved accordingly through ADR principles. At each stage starting from problem identification, artifact building, system designing, coding, testing, re-designing and to final testing involves a lot of learning. This learning through all the stages must be formalized according to the organizational structure. The entire ADR process is documented according to the project timelines for further discussions. This discussion involves a set of new design principles that I have learned through the study can be applied to projects that intersect with ours. The list is in Table 11.

<table>
<thead>
<tr>
<th>No.</th>
<th>Design Principles</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP01</td>
<td>Constant communication with the project members (Reflection R-01, R-04)</td>
<td>The success of any project directly depends on the involvement of members. Any new ideas or changes must be discussed with all the members and should agree on a common decision. The status of the project must be updated to all members.</td>
</tr>
<tr>
<td>DP01</td>
<td>Delimited Project Scope</td>
<td>The thesis scope must be precise and clear. In other words, the scope must be limited to a circle of the boundary. If not, the author efforts spread to a wide area and result in insignificant information to the study.</td>
</tr>
</tbody>
</table>
**Table 8 List of Design principles in the formalization of a learning phase**

The final solution to the problem is specified as design principles and contribution to theory. These results must be documented, formalized for further analysis and discussions.

### 3.3 Reliability and Validity

Any research study needs to prove its Reliability and Validity, for judging the quality of the designed thesis. These terms change their definitions based on the methodology used in the research. The current thesis uses ADR methodology and is one of the Qualitative research studies. The Reliability and Validity of this study are defined in the following text.

**Validity**

*Validity is a research process refers to the accuracy of research data. A researcher’s data can be said to be valid if the results of the study measurement process are accurate. In other words, it involves the approval of research findings and those findings are congruent with the theory* (Yilmaz, 2013) (Riege, 2003).

As discussed in *Section 2.4* there are various federated authentication protocols, that can be suitable to establish secure communication between the involving parties (IdP and SP). Among those protocols, SAML is chosen as the best authentication protocol for this thesis study based on the comparative analysis *Table3*. The underlying architecture of SAML protocol is designed to support certificate-based trusted authentication and data security in the form of SAML assertions of metadata between cloud-to-enterprise and vice-versa. Once the protocol has been selected, there were theories discussed on the possible methods of authentication and how they were applied on on-premises and cloud platforms. Most of the articles include; how the authentication has been applied on on-premises and cloud platforms individually but not together; how the SAML authentication modules can be implemented in different code languages, such as PHP, SimpleSAMLphp, .NET, etc. Additionally, some of the security issues in implementing the module...
were studied and all these identity methods not cost effective. The current literature review analysis along with the company’s application compatibility is studied and discussed with the project members to achieve validity in research study. Based on the decision, we decided to develop a security module for implementing the web login page to authenticate users securely using the same identity credentials on-premises and cloud. Therefore, to maintain the company application compatibility this study uses JAVA play framework to build a sample login page (supports SAML) and uses an open source WSO2 identity server (featuring cost effective identity solution) to grant secure access to the web application login page as discussed in Section 2.5.

**Reliability**

Reliability is a research process of ‘demonstrating’ the selected qualitative method and justifying the applied research strategies, procedure and methods are stabilized and consistence (Riege, 2003) (Yilmaz, 2013).

The applied research methodology in this process includes Problem formulation, Building Intervention, and Evaluation (BIE) cycles, Reflection and learning, and Formalization of learning. These stages should be demonstrated to prove its ‘Stability and Consistency’ in the process of research inquiry.

*Problem formulation stage (Data collection stage)*

It involves formulating the initial research question and defining the needed data resources. The thesis research question is clearly defined, and the principles of the study design are congruent with them at Section 3.2 ADR stage 1. The thesis main goal is to build an authentication module that can securely authenticate users on-premises and on cloud platform using the same identity credentials. The required hardware and software’s resources are clearly listed in Section 4.1 Table 9 & 10. The roles and responsibilities are explicitly mentioned in Section 3.2 Table 6.

*BIE stage (Building, Intervening and Evaluating the artifact)*

The building, intervening and evaluating the artifact was done with reasonable care and the resulted design depicted in Section 3.2 Figure 2. Each progress in application coding and its required parameters were shared and discussed with project members for achieving the desired result. The consistency of the code was first written in the prototype model before writing the actual code. The strength of the SAML request-response messages while authenticating the users is verified with the help of two main attributes. The one with the SAML assertion attribute value (terms of entity ID and URL), on the other hand with the certificate used for communication and its sequence communication is shown in the form of the diagram at Section 4.8 Figure 4. The current SAML authentication module potentially authenticates users securely by satisfying the various SAML attributes defined in the metadata files of the IdP and SP respectively. The designed security module method is proposed in the form of an algorithm below.
@SAMLSecurityModule provides

protected SAML2Client provideSaml2Client()
{
    cfg.setMaxAuthenticationLifetime(3600);
    cfg.setServiceProviderEntityId("< Entity Name of the SP used in the SP metadata file >");
    cfg.setServiceProviderMetadataPath(new File ("target","<SP-Metadata.xml filename>").getAbsolutePath());
    return new SAML2Client(cfg);

---

**Reflection and Learning (Analysis and documented)** (Riege, 2003) (Yilmaz, 2013)

The development of application code and its checks at each project stage are achieved within the mentioned time phase of Section 3.2 Table 7 with the involved project members. The code maintained its stability by solving its technical and non-technical issues that are encountered and its details are described in Section 3.2 ADR stage 3. The web application reliability is achieved by analyzing the integrity of the used SAML for communication between SP and IdP is shown in Section 5 Table 12 SAML Request-Response evaluation.
4. Design and Development

The current section describes the experiment setup, data collection, and methodology.

We start with initial design; a demo web application is developed in JAVA play framework using PAC4J security library for providing SAML authentication between SP and IdP. The web app consists of an index page with an URL to SAML authentication. Here the web app deployed system acts as a service provider. When the user clicks on the SAML URL, then it is redirected to the requested identity server (Wso2) login page. Once the user enters his/her credentials to requests access to a specific resource or service, the identity server validates the authentication request. If the authentication is successful, then it returns a SAML auth cookie information on the browser along with the profile information.

The redirection from the web app to the WSO2 identity server is established using the X509 certificate. The communication between the SP and IdP is verified by this trusted certificate. For this sample project, to test we have created a self-signed certificate in X509 certificate format. The web app X509 certificate additionally helps in establishing a trusted authentication to the Identity Server. The generated certificate is loaded into the WSO2 identity server trusted KeyStore. The role of IS with respect to certificate authentication is; it will authenticate the client (SP) using the client’s public key certificate.

After the user enters the credentials, the authentication validation takes place through the exchange of SAML metadata. The developed web application includes a service provider metadata in form of the XML format. The identity provider metadata is downloaded or gets automatically updated to the SP web app. Thus, the identity server verifies the SP metadata, before authenticating the user. A series of SAML communication messages takes place between SP and IP.

A more detail explanation for the above experiment setup was discussed in the above Section 3.5 considering ADR methodology.

4.1 Physical device and specification

The web application described in Section 3.5 is installed on an Ubuntu virtual machine hosted on a Virtual box. The virtual box is installed on a personal laptop. The specification of the laptop and a virtual machine used in this thesis work are given in Table 6 and Table 7.

<table>
<thead>
<tr>
<th>Device</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laptop</td>
<td>Processor: Intel(R) Core™ i5-6200U CPU @ 2.30GHz 2.40 GHz</td>
</tr>
<tr>
<td></td>
<td>Memory: 8:00 GB</td>
</tr>
<tr>
<td></td>
<td>System type: 64-bit Operating System, x64-based processor</td>
</tr>
<tr>
<td></td>
<td>OS: Windows 10</td>
</tr>
</tbody>
</table>

Table 9 The specification of the device used in this thesis
The specificatio
4.2 SAML

This section describes SAML (Security Assertion Markup Language) used in this study. It is an open standard protocol developed by Security Services Technical Committee of "Organization for the Advancement of Structured Information Standards" (OASIS), for establishing secured communication of identity information between participating entities (Anon., 2005) (Anon., u.d.). The entities involved are generally referred to as, the Identity provider(IdP) and Service provider (SP). The role of the IdP entity is to verify and validate the identity of a user asking for a service. In SAML context, they are also known as SAML authorities and ‘Asserting’ parties. Whereas, the role of SP entity is to use the IdP to validate the identity and provide the requested service to the user. In SAML context, they are also known as ‘Relying’ parties because they rely on information provided by Asserting party. 

It is an XML-based protocol framework, platform-independent, and working specification combines six main components (Kim, 2009):

- Security Assertions,
- Protocols,
- Bindings,
- Profiles,
- Metadata and,
- Authentication Context

### 4.2 SAML

This section describes SAML (Security Assertion Markup Language) used in this study. It is an open standard protocol developed by Security Services Technical Committee of "Organization for the Advancement of Structured Information Standards" (OASIS), for establishing secured communication of identity information between participating entities (Anon., 2005) (Anon., u.d.). The entities involved are generally referred to as, the Identity provider(IdP) and Service provider (SP). The role of the IdP entity is to verify and validate the identity of a user asking for a service. In SAML context, they are also known as SAML authorities and ‘Asserting’ parties. Whereas, the role of SP entity is to use the IdP to validate the identity and provide the requested service to the user. In SAML context, they are also known as ‘Relying’ parties because they rely on information provided by Asserting party.

It is an XML-based protocol framework, platform-independent, and working specification combines six main components (Kim, 2009):

- Security Assertions,
- Protocols,
- Bindings,
- Profiles,
- Metadata and,
- Authentication Context

### Table 10 The specification of tools and software’s used in this thesis

<table>
<thead>
<tr>
<th>Tools / Software’s</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenSSL</td>
<td>1.0.2-x86_64</td>
</tr>
<tr>
<td>Oracle Java SE Development Kit (JDK)</td>
<td>Build 1.8.0_152</td>
</tr>
<tr>
<td>Eclipse Java EE IDE for web developers</td>
<td>Oxygen.1a Release(4.7.1a) Build id: 20171005-1200</td>
</tr>
<tr>
<td>Web browser</td>
<td>Mozilla Firefox 57.0.1 (64-bit)</td>
</tr>
<tr>
<td>Sbt</td>
<td>sbteclipse-plugin - 5.2.2</td>
</tr>
<tr>
<td>Oracle VM VirtualBox Manager</td>
<td>Version 5.2.2 r119230(Qt5.6.2)</td>
</tr>
<tr>
<td>WSO2 Identity server</td>
<td>5.3.0</td>
</tr>
<tr>
<td>Heroku</td>
<td>Cloud Application Platform</td>
</tr>
</tbody>
</table>
method used and time of authentication from relying upon parties (SP) to (IdP). The entitlement statements indicated whether the authenticated user is subjected to perform the desired action. These action decisions are provided through access control rights. The attribute statement indicates whether the defined attribute for example name, age, job and so on, are used in access control decision making between relying parties.

2. **Protocols:**

The protocols are used for issuing and exchanging the above security assertions in a form of the package between the SP and IdP. Such Protocols are Authentication request, request-response, Assertion query, Artifact resolution and so on. Each package specification varies with a set of rules and SAML elements such as, request and response elements that the IdP or SP should follow.

3. **Bindings:**

The SAML binding is a mapping of the above protocol messages with standard messaging formats and/or communication protocols. Some of these bindings are SAML SOAP, HTTP Redirect(GET), HTTP Post, SAML URI and so on.

4. **Profiles:**

The SAML profile is a technical description of how SAML assertions, protocols, and bindings combine to address a specific use case of a project or a deployment scenario. Some of these profiles are Web browser SSO, Single logout profile, Enhanced client or proxy (ECP), IdP discover profile, SAML attribute profile and so on. The most important and extensively used profile is the Web Browser SSO profile. For example, if the Web SSO Profile is used for SAML authentication, it details how authentication assertions are exchanged between entities (including what SAML protocols and bindings are used).

5. **Metadata:**

The SAML metadata defines a configuration data in an XML schema format and exchanges this data between the entities. It defines data like what service is available, addresses, operational roles (IdP or SP), bindings, certificates with Key information for encryption and signing and so on. The SP uses the metadata to know how to communicate with the IdP and vice versa.

6. **Authentication Context:**

The Service provider request IdP to uses some type of Authentication context for the user. This information includes the type and strength of authentication.

The relationship between these components is like building blocks as shown in below Figure. When they are put together, provides flexibility and extensibility to implement a
system and supports various business use cases. Among the supported use cases, Web single sign-on profile use case and identity federation use case are considered for this study.

![SAML protocol structure](image)

*Figure 5 SAML protocol structure, Taken from (Kim, 2009)*

### 4.3 Web SSO federated Identity

This section describes Web Single sign-on and federated identity.

With the involvement of cloud usage for enterprise applications arises many problems. Among them, user identity management is one of the tops most concern. How such multiple user login identities for a single user can be maintained in both on-premises and cloud accounts, password recovery, etc. Also, an increase in fraud rate and user’s negligence in writing down multiple passwords/forgetting possessing an additional burden. To solve these problems, two new solutions took a way through to benefit both the businesses and end-users. They are a Single sign on and new identity management structure call as Federated identity management (Andronache & Nisipasiu, 2011).

- **Web Single Sign-on (SSO):**

  Single Sign-on (SSO), is one of the best solutions for accessing multiple applications with a single user account. In simple terms, it is an authentication
service that allows users to use one set of login credentials (i.e., username and password) to access the multiple application that he/she has rights within a single established session. Thus, eliminating the need to switch to multiple sign-on dialogues prompts for login into multiple applications within the active session. When applied to enterprise level, SSO provides a path for business employees to use the same credentials to access public facing sites without the burden of additional login passwords. It eliminates the burden of managing user accounts, cost involved for strong authentication systems and reducing SaaS licensing cost by 30% (Rabinovich & Paul, 2015). There are many SSO solutions available for the business. The implementation solution can be chosen based on business requirements. Considering SAML is used for message communication in this project, then web-browser SSO profile implementation is best suited. Web browser SSO profile uses SAML assertion message, Authentication request protocol, along with bindings HTTP redirects and HTTP POST between the participating entities (i.e., SP and IdP).

- **Federated Identity**:

  The concept of federated identity is to establish an identity trust between a group of organizations/participating entities. The goal is to enable secure trust relationships between a group of organizations/SPs to share information about the identity of its user’s. The identity data is stored at one location i.e., IdP and multiple service providers can enable user authentication to access the required resources. In this way, organizations avoid the burden of managing user's identities. In this study, we are going to use a Federated identity management system WSO2(IdP) for managing user identities. The Federation of identities between the SP and IdP can be accomplished using many available open standards, such as SAML, OpenID, OAuth and so on. This study focuses on SAML and why it is used is described in the literature review Section 2.4.

### 4.4 OpenSSL

This section describes the OpenSSL tool. It is an open-source toolkit to enable HTTPS connections between the parties involved in communication or exchange of data. This connection can be built by generating a new self-signed certificate using OpenSSL commands. This includes generating a private key, certificate file and certificate format conversion (if a conversion is necessary). The certificate file represents a Common Name to validate the application. This tool is much helpful for testing purposes or internal use for the companies (Anon., u.d.).

In this study, OpenSSL is used for generating a certificate for the developed web application (i.e., Service provider). This certificate will be used by the service provider(SP) for establishing secure communication with an Identity provider (IdP). It was decided that the data exchange between the SP and IdP will be guided by the SAML protocol, hence the certificate is generated in X.509 supported format.
**Keytool:**

Another tool called ‘Java Keytool’ is used as a certificate management utility and by default available in JAVA software. It stores the keys and certificates in what is called a `.jks` KeyStore. Considering the web application is developed using JAVA, the above generated X.509 certificate format should be converted into a Java KeyStore file. This helps to solve issues with certificate incompatibility.

Below are the steps of certificate generation, conversion formats and importing:

1. Generating an X.509 certificate for service provider and converting the generated certificate into .jks JAVA format using JAVA key tool.

![Command Prompt](image)

*Figure 6 Using Java Keytool to convert the certificate into .jks format*

2. Printing the certificate file in x.509 format for supporting SAML communication. The certificate printed below can be copied into the SAML metadata files.
3. Importing the SP certificate into the Wso2 identity server(IdP) client-trust store:

**Figure 7 Printing certificate into X.509 format**

**Figure 8 Importing the SP certificate into WSO2 IS certificate store**
4.5 PAC4J library

This section described the PAC4J library.

It is a powerful Java security engine to protect any web application built using most available frameworks/tools (Anon., u.d.). The most supported frameworks/tools for web application implementation are J2E, Spring Security, Play 2.x, Java Spark, JAX-RS and so on. It also supports many of the authentication and authorization mechanisms used in today’s applications. In this study, play framework 2.6 is used for developing the sample web application and SAML as the client’s authentication mechanism. It also helps to return user profile attributes for authenticated client.

Components:

- **Client:** An SAML client performs the login process and returns a user profile. A user profile is the profile of the authenticated user, that return user identifier, profile name, client name, and some additional attributes.
- **Config:** It defines the security configuration via the client.
- **Security filter:** The security filter protects the URL by checking that the user is authenticated, according to the SAML client configuration.
- **Callback controller:** It helps to finish the login process of the user.
- **Logout controller:** It handles the application and identity server logouts for the users.

4.6 WSO2 Identity server

This Section describes the open source Wso2 Identity server used for authenticating the entities.

It is an identity and entitlement management server that enables IAM related activities for the enterprise architects and developers. It is based on open standards and open source principles. It comes with seamless, easy to use integration capabilities for connecting application or service providers, user stores, directories, and identity providers. It provides secure SSO, identity federation, strong authentication, identity administration, monitoring and so on (Xu & Yang, 2015). Refer (Anon., 2015-2016) for more detail explanation about the server architecture, installation, and configuration.

4.7 SAML metadata

Metadata is a configuration data used for negotiating agreements between the participating entities. The configuration data is presented in a nested tree structured XML-format. It contains information about the participating entities, bindings, and protocols used for communication. It even contains certificates, private/public keys, and cryptographic capabilities for establishing trust and secure exchanging of data. Metadata has the flexibility to be customized according to the use case. Exchanging metadata that contains specific information, determines that specification will be used for that business case.
Here in this study, SAML is used for establishing communication between the SP and IdP. The metadata configuration is structured to the SAML standard specification for efficient process deployment. The configuration parameters are defined in a syntactical format with elements and attributes; starting ‘<>’ and ending ‘</>’ tags. For example, the element within tags are defined as below:

```xml
<md: keyDescriptor>
  <ds: KeyInfo>... </ds: KeyInfo> //The element contains information about certificate keys.
</md: keyDescriptor>
```

**Metadata elements:**

The key building block for SAML metadata is the ‘EntityDescriptor’. It is a container element for describing the system entities such as an identity provider or service provider. It contains attributes as:

- **valid until and cacheDuration**: Helps in keeping the delivered metadata fresh.
- **entity ID**: It is a unique string or URL that distinguishes it from other entity.

This EntityDescriptor also contains two roles for defining the providers. Based on the provider used, the metadata file format is structured. For describing the roles, the `<RoleDescriptor>` elements `<IDPSSDescriptor>` for IdPs, `<SPSSODescriptor>` for SPs is used. This role descriptors also contains few data elements to describe the way parties/entities communicate. Formats of some of these data elements are common to both the providers, while some of those elements are used respectively to the roles. They are:

- **Algorithm Support**: These elements are used for supporting multiple secure algorithms to protect the data from cyber-attacks. It adds secure ciphers to the data communicated and provides flexibility. The attribute ‘alg’ supported element is `<Extension>`.

- **Discovery support**: These elements are used for provisioning an IdP or SP discovery service.

  The SP offers services of SSO protocols by including an Attribute consuming service named `<AssertionConsumerService>` endpoint element in their metadata. This element contains three main attributes ‘Binding’, ‘Location’ and ‘Index’. The location in which the IdP sends the authenticated user request to the SP. The binding defines the communication protocol used for profile redirection eg: HTTP-POST. The index is a small positive integer, that should be unique among the defined role.

- **Signing/encryption keys**: These elements are used for exchanging the signed and encrypted public key between the parties. It also helps in verifying and validating the entities in communication.

  The element `<KeyDescriptor>` is a wrapper around the `<ds: KeyInfo>` elements. The KeyDescriptor defines the use of a key for signing or encryption. The key info is used for describing the keys. Additionally, SAML metadata specification contains `<ds:X509Certificate>` element. It is used to include the certificate in an X509 format for establishing the trust relationship between the entities.
• **Endpoints and NameIDs:** The Endpoint type describes a protocol binding endpoint. These bindings define the type of SAML assertions messages for sending between entities. Other attributes they are bounded to this type are ‘Location’, ‘ResponseLocation’.

The protocol specific binding used in IdP, are `<SingleSignOnService>`, `<SingleLogoutService>`.

The NameIDs element is used to support name identified formats, that are used for authentication exchange. The elements are in format ‘urn: oasis: names:tc:SAML:2.0:nameid-format:unspecified’.

• **Additional elements:** The element `<organization>` specifies basic information about an organization responsible for a SAML entity or role. The use of this element is always optional. Its content is informative in nature and does not directly map to any core SAML elements or attributes.

In summary, the SP and IdP role typically includes the following information as shown in Table 10.

<table>
<thead>
<tr>
<th>IDPSSDescriptor (Identity Provider)</th>
<th>SPSSODescriptor (Service provider)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public keys are used by IdP for authentication and encryption. Eg: <code>&lt;Extension&gt;</code>, <code>&lt;KeyDescriptor&gt;</code></td>
<td>Public keys are used by the SP for authentication and encryption. Eg: <code>&lt;Extension&gt;</code>, <code>&lt;KeyDescriptor&gt;</code></td>
</tr>
<tr>
<td>Various types of endpoints used for communicating between the parties. Eg: <code>&lt;SingleLogoutService&gt;</code>, <code>&lt;ArtifactResolutionService&gt;</code></td>
<td>Various types of endpoints used for communicating between the parties. Eg: <code>&lt;AssertionConsumerService&gt;</code></td>
</tr>
<tr>
<td>Explicitly supported named identifier formats, if any. Eg: <code>&lt;nameIDFormat-unspecified&gt;</code></td>
<td>Explicitly supported named identifier formats, if any. Eg: <code>&lt;nameIDFormat-emailAddress&gt;</code></td>
</tr>
<tr>
<td>Explicitly supported attributes, if any. Eg for SAML; protocolSupportEnumeration.</td>
<td>Explicitly supported attributes, if any. Eg for SAML; AuthRequestSigned, WantAssertionsSigned.</td>
</tr>
</tbody>
</table>

*Table 11 Summary of SP and IdP role descriptors used in metadata files*
4.8 SAML message exchange

This Section describes the SP initiated SAML message exchange

In the Service Provider(SP) initiated the process, the user accesses the webpage without passing to the IdP first. The service provider will take the responsibility to issue the SAML request back to IdP on behalf of the user. The detail message exchange steps are presented in the below Figure 3 an explanation as follows:

1) The user arrives at the webpage of the company, but without SAML assertion.
2) The SP redirects the request to the appropriate IdP login webpage. This is done with the help of exchanging metadata files with attributes data, such as entity ID.
3) The SP returns the IdP SSO page to the user browser.
4) The user enters the login information. Then the SP initiates the actual SAML authentication process by sending the SAML AuthRequest to the IdP.
5) The IdP identifies the user by verifying the details of AuthRequest assertions of SP and its internal user store.
6) The IdP provides a SAML response to the SP.
7) Based on the response, the SP provides returns the requested resources or an HTTP error.
Figure 9 Sequence diagram of SAML communication between User browser, SP, and IdP
The developed web application for this thesis study is as below:

*Figure 10* Developed web application index page
5. Output Evaluation

Evaluation of the developed IT artifact using suitable design frameworks is one of the key activities in any Design research methodology. This evaluation results in achieving the methodology stated goals and objectives. It also helps to provide feedback, at each phase of thesis development. In this study, the Framework for Evaluation in Design Science (FEDS) is chosen for evaluating the developed artifact. This framework consists of four steps to evaluate (Venable, et al., 2016):

(1) **explicate the goals of the evaluation,**

The goals for the designed application were that the user’s authentication must be accurate, application security, design usable and less cost for implementation. The goals met were discussed in the below output evaluation.

(2) **choose the evaluation strategy,**

This thesis project is built on a laboratory setup, works on a specific technology and with less cost. The strategy that best suits, is a Technical Risk & efficacy evaluation strategy. This strategy starts with formative evaluation whether this specific technology may work as perceived at the start of the design. Then proceeds to artificial evaluation with a laboratory experiment to clarify the boundaries of the technology including the cost of setup.

(3) **determine the properties to evaluate, and**

The parameters chosen for output evaluation are related to web technology used in this thesis. The parameters details are listed below Figure 11 SAML logger output.

(4) **design the individual evaluation episode(s).**

The design of the individual evaluation episodes is presented in Figure 8. This thesis is carried out in two iterations. Starting with Ex-ante evaluation for deciding, whether to opt for the chosen products and technology. Followed by artificial evaluation for implementing the system according to the BIE methodology stage.

The output of the designed application needs to be validated to prove its accuracy and security, along with the strength of SAML messages while authenticating the users. In this project SAML web SSO profile is implemented using the developed application, therefore a series of SAML message assertions exchange over the HTTP-POST between the participating entities. Now, it’s time to validate the correctness of sent attributes in the assertion by using a SAML tracer tool program. It can be installed and added as an Add-on to the Mozilla Firefox browser. It helps in analyzing the SAML data and its corresponding attributes that are communicated between the parties, i.e., SP and IdP. This output also helps to learn the actual authentication process by sniffing the SAML response and request messages. This learning is necessary to make sure that the right user is authenticated to the right resources.

Once the user starts accessing the webpage, the SAML logger can be activated by clicking its icon on the browser. Once it is activated, all the user actions are logged in the SAML tracer window. The logs not only show the normal HTTP requests of the designed application but also highlights and decodes SAML messages transmitted between the entities. The HTTP tracing lines that
include the SAML data are highlighted with a SAML orange logo on the right side of the URL in the window, as shown in Figure 10.

Selecting a request gives you up to three tabs. Each tab has any of the below trace information:

- **HTTP**: A quick overview of the request, with request and response headers.
- **Parameters**: GET and POST parameters included in the request.
- **SAML**: Decoded SAML message found in the request.

![Figure 12 SAML Logger output](image)

The output lists a series of requests between the participating entities. Each request can be validated to ensure proper attributes are passed. The POST action shows the hidden SAML Request and Responses fields in XML format. The output is validated according to the below SAML assertions (i.e., elements and attributes) to verify its message integrity, Security, and accuracy in communication.

The output is inspected by copying the SAML data into a text editor. The below items need to be validated according to the value of each assertion. Some of these items need to be cross verify with the application metadata. The application metadata consists of two sets of XML files, one is created metadata at the service provider and other metadata generated at the identity provider respectively. Figure 11 shows the below items:

- **The Signature and Digest method hashing levels:**
  The integrity of the messages must be preserved from attacks such as man-in-the-middle, spoofing. In dealing with this scenario, SAML assertions can be signed digitally using various supported signature algorithms. The selection of the algorithm depends on the type of data and sensitivity required for the application. Here it uses RSA-sha256, sha512 algorithms and so on in the request/response transmissions.

- **Validate the X.509 Certificate being passed between Idp and SP:**
  The SAML standard allows using an X.509 certificate for establishing a secured back channel connection between the participating entities. This helps, the encoded SAML messages to be transmitted in the same trusted channel. The author uses the public key of
the certificate to verify that the content of the SAML response matches with the user key in the metadata of application code. The response from the asserting provider is valid and not been tampered if only the asserting provider has the matching private key to the public key in the message. If the response is from a valid trust, the application accepts the user details supplied in the SAML attributes for successful authentication.

There are two main reasons for using the public key in the SAML assertions:

- The tampering check is quicker and can take precautionary actions if the sending public key is unknown.
- To know in advance from which identity provider the assertion has come.

c. The Issuer URL / Entity ID:
   The entity ID that is displayed in the SAML output needs to be the same as the entities specified in the metadata of the application code. This helps us to know that the passing assertions of authentication are with the right entity ID.

d. The SAML attribute assertions including their format and value
   The entity specifying the right attribute value is taken for authenticating the client.

e. The Assertion Consumer Service (ACS) URL (Reply URL), SingleSignOn URL and SingleSignOn Logout URL:
   The specified destination, identity server access control URL’s for sign-in and sign-out must be specified correctly for the right redirection to take place for completing the user’s desired actions.

f. Use of SAML attribute NotBefore and NotOnOrAfter:
   The use of these condition attributes in SAML compromises the reply attacks. The attacker can intercept the valid assertions and saves the data for impersonating at a later time. To avoid such a scenario, these condition attributes for a specified subject (IDP/sp) is set to a time until which the assertions are valid. The period is set as short, around 5 minutes. With these conditions even if the attacker compromises the data, those assertions are not valid.
Figure 13 Captured SAML data while authenticating user
### SAML Authentication Request

**AuthnRequest**

```xml
<?xml version="1.0" encoding="UTF-8"?>

Service provider SAML calls the specified assertion URL in the Identity Provider.

- User's SAML authentication request calls the "method": "POST" on port 9000, for redirecting the page to the Identity Provider.

- The Identity server runs on port 9443 and verifies the below value of
  `<AssertionConsumerServiceURL>` exactly matches the `md: AssertionConsumerURL` in the metadata corresponding to the service provider. Once verified the user login page is redirected to the "URL":
  `https://iamhost.local:9443/samlsso` of the Identity server "value": "WSO2 Carbon Server".

- Once the Authentication request is called, there establishes a session with a unique ID. The user will be active throughout the communication on this ID
  `_imwmjpurd0qefnxuurb9v5l4szwq9i7upbhs0g` . The request responses are sent on this ID.

- The issue of the SAML request date and time is also mentioned below.

```xml
<saml2p: AuthnRequest
 xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
 AssertionConsumerServiceURL="http://iamhost.local:9000/callback?client_name=SAML2Client"
 Destination="https://iamhost.local:9443/samlsso"
 ForceAuthn="false"

ID="_imwmjpurd0qefnxuurb9v5l4szwq9i7upbhs0g"
 IsPassive="false"
</saml2p: AuthnRequest>
```

### SAML Authentication Response

**Response**

```xml
<?xml version="1.0" encoding="UTF-8"?>

Identity provider SAML calls the destination URL of the service provider.

- The user is authenticated and the "method": "POST", redirects to the Destination service provider page on port 9000, on "URL":
  `http://iamhost.local:9000/callback?client_name=SAML2Client`,

- The requested user profile information is sent in the form of SAML Response to the user application.

- The response is sent on the same established ID of SAML AuthRequest.

- The issue of the SAML response data and time is also mentioned below

```xml
<saml2p: Response
 xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
 Destination="http://iamhost.local:9000/callback?client_name=SAML2Client"

ID="_40f19f3054ce160e2a922b36afbdce12b"
 InResponseTo="_imwmjpurd0qefnxuurb9v5l4szwq9i7upbhs0g"
 IssueInstant="2018-04-30T13:10:57.882Z"
 Version="2.0"
</saml2p: Response>
```
SAML Authentication request messages of service provider name 'pac4j-saml' use an HTTP user agent for binding the request with HTTP-POST message. This same user agent format is used for exchanging messages between the parties.

<table>
<thead>
<tr>
<th>ProtocolBinding</th>
<th>&quot;urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST&quot;</th>
<th>ProviderName</th>
<th>&quot;pac4j-saml&quot;</th>
<th>Version</th>
<th>&quot;2.0&quot;</th>
</tr>
</thead>
</table>

These issuer elements contain the entityID of the Service provider used for establishing SAML request assertions communication:

- `<saml2:Issuer xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" sp.storigo.se></saml2:Issuer>`

The elements in this metadata are digitally signed for metadata integrity and authenticating the metadata by a trusted signer. The data can be signed with flexible format choices and algorithms. The algorithms for chosen methods are shown below:

| <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"> |
| <ds:SignedInfo> |
| <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/> |
| <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more:rsa-sha256"/> |
| <ds:Reference URI="#_imwmjpurdoqefnxcuurbv5l4szwq9i7upbhs0g"/> |

These provider used for establishing SAML request assertions communication:


The elements in this metadata are digitally signed for metadata integrity and authenticating the metadata by a trusted signer. The data can be signed with flexible format choices and algorithms. The algorithms for chosen methods are shown below:

| <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"> |
| <ds:SignedInfo> |
| <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/> |
| <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more:enveloped-signature"/> |
| <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/> |
| <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/> |
| <ds:Transforms> |
| <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha12"/> |

These provider used for establishing SAML request assertions communication:


The elements in this metadata are digitally signed for metadata integrity and authenticating the metadata by a trusted signer. The data can be signed with flexible format choices and algorithms. The algorithms for chosen methods are shown below:

| <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"> |
| <ds:SignedInfo> |
| <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/> |
| <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more:enveloped-signature"/> |
| <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/> |
| <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/> |
| <ds:Transforms> |
| <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/> |

These provider used for establishing SAML request assertions communication:


These provider used for establishing SAML request assertions communication:

The SAML metadata entity imports certificates in the form of X509 certificate format for secure transporting of SAML messages in the back channel. Below is the certificate in x509 format and contains the public keys, whereas the private key is kept secret with the entity. These keys are used for message-level signing. The below certificate match with the certificate in the metadata file. This proves the same clients are participating in the communication.
The completion of the Authentication request to the Identity provider from the service provider.

<saml2p:AuthnRequest>
  
  The Identity provider is satisfied with the user authentication request and sent a <response> message with appropriate status code 'SUCCESS'.

  <saml2p:Status>
  </saml2p:Status>

  The assertion ("_16cca76f6b2ddff3359fa92b8345617b") was issued at time "2018-04-30T13:10:57.932Z" by identity provider (idp.storigo.se)


    The elements in this metadata are digitally signed for metadata integrity and authenticating the metadata by a trusted signer. The data can be signed with flexible format choices and algorithms. The algorithms for chosen methods are shown below

    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
        <ds:Reference URI="#_16cca76f6b2ddff3359fa92b8345617b">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
          </ds:Transforms>
          <ds:DigestValue>CijTYGm7a5WCEoxzoRfrOsEnSSg="/ds:DigestValue">
        </ds:Reference>
      </ds:SignedInfo>
    </ds:Signature>
  </saml2p:Assertion>
</saml2p:AuthnRequest>
The SAML metadata entity imports certificates in the form of X509 certificate format for secure transporting of SAML messages in the back channel. Below is the certificate in x509 format and contains the public keys, whereas the private key is kept secret with the entity. These keys are used for message-level signing and encryption.

The principal identified in the <saml2:Subject> element is a user at this company. The user 'Admin' is authenticated at a time "2018-04-30T13:10:57.932Z" by means of the password sent over a protected channel. The relying party verifies the request of this user profile with the associated subject of the assertion.
The `<saml:Conditions>` element, which gives the conditions under which the assertion is to be considered valid. The conditions are 'NotBefore' and 'NotOnOrAfter'

a. **NotBefore** - Assertion isn’t valid prior to the time specified = "2018-04-30T13:10:57.932Z"

b. **NotOnOrAfter** - Assertion is not valid on/after time specified = "2018-04-30T13:15:57.882Z"

The response assertions are sent over 13:10. After time period 13:15 these assertion are not valid.

The SAML response assertion with authentication issued at a time "2018-04-30T13:10:57.952Z" and over the session ID "f39cdd07-84ba-490c-9335-9838a2d07038" is returned to the profile page of the output.

The completion of the Authentication response to the service provider from the Identity Provider.

---

**Table 12 SAML Request – Response output evaluation**
6. Limitations and Future research

This section discusses the limitation and future research for this thesis study.

- The SaaS application presented in this thesis was demonstrated in a testbed environment. With that ‘Interoperability’ is one of the main concerns. It was discussed in the reflection stage of section 3.5, that the application development was switched on to the Unix platform due to the local domain issue of windows platform. This limits the application to Unix platform as shown in the thesis. However, this problem can be solved once the application is loaded into the live environment with actual domain names of windows without any additional changes in code.

- No end user involvement, as it is at practitioner stage.

- Another limitation of this study is that certificates used are self-signed certificates. If the application is going to be working on a live environment, for better security it needs to be signed by a trusted certificate authority (CA).

- The Technical Risk & efficacy evaluation strategy is used for evaluating the output and it resulted to be the best suitable strategy for this thesis. This strategy starts with formative evaluation whether this specific technology may work as perceived at the start of the design. Then proceeds to artificial evaluation with a laboratory experiment to clarify the boundaries of the technology including the cost of setup.

- The current method output analysis was done up to 10 more users to verify the integrity of the application. However, the identity server has a capability to integrate database applications at the backend for holding ‘n’ number of user accounts. After successful authentication, the user profile is returned to the webpage for the only testing purpose.

- As a Future work, deploying it to a production environment and testing will result in choosing the Purely Technical Artifact strategy for evaluating the output.

- This thesis shows that the developed application is uploaded to the cloud with no additional changes. The Cloud output evaluation is not shown.

In the current study, the scope was on implementing a secure authentication mechanism that works both on-premises and cloud without any additional changes in the application code. This scope can be further expandable to implement fine-grained authorization module for the authenticated users.
7. Discussion and Conclusions

The contribution from this thesis to research is practically demonstrated that the SP application and IdP server can be connected and transmit data identities together easily and effectively while maintaining security. The contribution to practice was mentioned in the following text. The solution presented in this thesis was able to authenticate users securely and correctly both on-premises and cloud. This is achieved by integrating the WSO2 identity server with SAML authentication module of SaaS application. The SaaS application was developed in a JAVA play framework, which supports PAC4J security library. The library is developed in JAVA and supports integration with any SAML providers and play web applications. It has a very powerful security engine to protect the web application and ships with several authentication modules. The current solution uses this library to implement web browser SAML SSO profile for authenticating users. This profile provides functionality for the two scenarios: SAML- Service provider and SAML-Identity Provider. The developed SaaS application acts as a service provider while WSO2 identity server acts as an Identity Provider. The integration of SP and IdP is done with the help of certificates and Metadata files. The trust channel is established by importing the SP certificate into the IdP trust store. Both providers have their metadata files with attributes, that can be distinguish using the entity ID. It also includes the certificates in X509 format to establish a secure channel for user session communication.

When the user requests to access the SP application, a series of communications takes place as shown in Figure 8. When the authentication request starts, SAML request-response assertions messages flow between the SP and IdP with the help of data found in the metadata files.

The detail output evaluation of ‘SAML Request’ and ‘SAML Response’ assertions shows that the SAML SSO profile can be used to authenticate users due to its correct data transfer and security. The integration of the SAML module with an identity server suggests that additional security and reduction in application cost can be obtained. The additional security refers to well-established X509 certificates and security module of the Pac4j library. This integration further showed that the generated certificates are difficult to understand or decode, which is an extra advantage of this methodology.

The previous research in this area showed that applications using the SAML are demonstrated only for cloud models but not for the on-premises. The developed SaaS application in this thesis contributes to the research gap in theory and proves that it works for on-premises and can be easily uploaded to the cloud with no additional changes in the code. The identity servers so far used in such implementations results in an additional extra cost to the companies. However, adapting to Wso2 server within this implementation can be cost-effective, easy to install, customizable as per company needs, run on a browser without additional hardware support, used as a developer platform, and, finally independent platform.

In our research process, we have chosen to use Action Design Research methodology and it helped the organization management to achieve the desired outcome. During the first stage of the ADR process, the initial artifact problem has been identified and then the research gap was formulated in accordance with the gained theory principle. The second stage added more knowledge in building the artifact in accordance with the practice-oriented principle. This stage follows a series of iterations of testing and evaluating the artifact output. The use of IT-Dominant BIE helped in
solving the technical problems occurred at each alpha stage. The third stage helped in reflecting the technical, non-technical challenges faced while developing the artifact. The fourth stage helped in formalizing the learning at each stage. Then how the final artifact can be deployed into the organization structures by generalizing the outcomes in the form of learned design principles. These stages are explained in detail at Section 3.2.

The contribution of using this research methodology helped in achieving the desired outcome. The results seem promising, but additional testing of the finished artifact is achieved when deployed it into the production environment. If I were to redo my thesis, I would use the same methodology. The IT-Dominant of BIE stage helps in changing the implementation part without disturbing the baseline structure of this thesis. It can be carried out in two phases of cycles Alpha and beta. The Alpha stage is were the thesis ended up, with the help of beta stage it can be deployed to production and test it with real customers.

The findings from this thesis can be helpful for startup companies for the initial development of user’s authentication methods and looking for the same application running both on-premises and cloud. This thesis further suggests that the implementation of the authorization module for user authentication can be considered as future work.
8. References


Anon., 2005. OASIS SAML-metadata-V 2.0. [Online]
Available at: https://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

Available at: https://docs.wso2.com/display/IS530/WSO2+Identity+Server+Documentation

Anon., n.d. OASIS - Saml-profiles-V2.0, OASIS - Saml-Bindings-V2.0. [Online]
Available at: https://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf ;
https://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf

Available at: https://www.openssl.org/

Anon., n.d. PACJ. [Online]
Available at: http://www.pac4j.org/


Available at: https://msdn.microsoft.com/en-us/library/aa480030.aspx

Christiansen, C. A. & Laura, S., 2016. Identity as a Service on the Journey to the Cloud. [Online]


Riege, A., 2003. Validity and reliability tests in case study research: a literature review with “hands-on” applications for each research phase. *Qualitative market research: An international journal*, 6(2), pp. 75-86.


Vom Brocke, J. et al., 2009. Reconstructing the giant: On the importance of rigour in documenting the literature search process. *In ECIS*, Volume 9, pp. 2206-2217.


Appendix

Experiment screenshots

The metadata files contain the necessary information for establishing the SAML data exchange between the Service provider web page and Identity server. The below metadata files are created and deployed to the web application.

Metadata file of the service provider (SP):

```xml
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor
  ID="fa0f0a32de3c9747e2f3a2f14sa1e2" entityID="sp.storigo.se"
  validUntil="2038-03-21T00:58:28.639Z">
  <md:Extensions>
    <SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="true"
      <md:Extensions>
      <md:KeyDescriptor use="signing">""</md:KeyDescriptor>
      </md:Extensions>
      </SPSSODescriptor>
    </md:Extensions>
  <md:EntityDescriptor
    ID="fa0f0a32de3c9747e2f3a2f14sa1e2" entityID="sp.storigo.se"
    validUntil="2038-03-21T00:58:28.639Z">
      <md:Extensions>
        <SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="true"
          <md:Extensions>
        <md:KeyDescriptor use="signing">""</md:KeyDescriptor>
        </md:Extensions>
      </SPSSODescriptor>
    </md:Extensions>
  </md:EntityDescriptor>
</md:EntityDescriptor>
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

![XML Document](image)

[Image of XML document]

This image is a screenshot of an XML document related to SAML metadata.
</ds:X509Data>
</ds:X509Certificate>

</md:KeyInfo>

</md:KeyDescriptor>

</md:NameIDFormat>

</md:NameIDFormat>

</md:NameIDFormat>

</md:NameIDFormat>

</md:NameIDFormat>

</md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="http://iamhost.local:9000/callback?client_name=SAML2Client" index="0"/>
</md:SPSSODescriptor>
</md:EntityDescriptor>
Metadata file for Identity provider (IdP):

```
<EntityDescriptor entityID="idp.storigo.se">
  <!DOCTYPE IdPSSEntityDescriptor[ EXPRESS ]>
  <IdPSSEntityDescriptor>
    <KeyDescriptor use="signing">
      <X509Data>
        <X509Certificate>
          MIICCTCAZgAwIBAgIESA33qtJgANBgkqhkiG9w0BAQUFADBVMQswCQYDVQQGEwJb
          EVUxELMAkGA1UEBAoIbG9jYVwoeDQELMAcGBh0JacobtouSHAxOCAQAxg1
          AQAwDQYJKoZIhvcNAQEFMQgAQCgQYMWR2oZYt6233r9cTLzJz/wgSfT98V
          R57uTn53t3aBoP6v6IvI8rPa+D7k/nXxR54HrK5TzaAcC1U70QDl
          AqBhObxwEDAOBgNVHQ8BAf8EBAMCBwAwgggDAgEBPzAfBgNVHSMEAgIw
          DQYJKoZIhvcNAQEFMQgBQCgQYMWR2oZYt6233r9cTLzJz/wgSfT98V
          R57uTn53t3aBoP6v6IvI8rPa+D7k/nXxR54HrK5TzaAcC1U70QDl
          AqBhObxwEDAOBgNVHQ8BAf8EBAMCBwAwgggDAgEBPzAfBgNVHSMEAgIw
          DQYJKoZIhvcNAQEFMQgBQCgQYMWR2oZYt6233r9cTLzJz/wgSfT98V
          R57uTn53t3aBoP6v6IvI8rPa+D7k/nXxR54HrK5TzaAcC1U70QDl
          AqBhObxwEDAOBgNVHQ8BAf8EBAMCBwAwgggDAgEBPzAfBgNVHSMEAgIw
          DQYJKoZIhvcNAQEFMQgBQCgQYMWR2oZYt6233r9cTLzJz/wgSfT98V
          R57uTn53t3aBoP6v6IvI8rPa+D7k/nXxR54HrK5TzaAcC1U70QDl
          AqBhObxwEDAOBgNVHQ8BAf8EBAMCBwAwgggDAgEBPzAfBgNVHSMEAgIw
          DQYJKoZIhvcNAQEFMQgBQCgQYMWR2oZYt6233r9cTLzJz/wgSfT98V
          R57uTn53t3aBoP6v6IvI8rPa+D7k/nXxR54HrK5TzaAcC1U70QDl
        </X509Certificate>
      </X509Data>
    </KeyDescriptor>
    <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
      Location="https://iamhost.local:9443/saml/mso" />
  </IdPSSEntityDescriptor>
</EntityDescriptor>
```

Redirected to the WSO2 identity server login page:
After entering the user’s credential, the code returns the user profile information. This helps in testing the web application code integrity and its functionality. It means, whether the right user profile has authenticated and additionally helps in identity verification.

Once the user clicks on log out, then the previously returned profile information should vanish from the index page.

SAML code from Security module file:
Cloud Deployment

Application deployed to the cloud:
```bash
lamhost@lamhost:~/Documents/Service provider AppS
$ sbt stage deployHeroku
```

```text
| INFO | Executing in batch mode.
| WARN | For better performance, hit [ENTER] to switch to interactive mode, or consider launching sbt without any commands, or explicitly passing 'shell=true'.
| INFO | Loading project definition from /home/lamhost/Documents/Service provider AppS/project
| # ms |
| INFO | downloading https://repo.scala-sbt.org/scalasbt/sbt-plugin-releases/com.horokou/sbt-heroku/sca... sbt-heroku jar...
| INFO | [SUCCESSFUL] com.horokou#sbt-heroku;2.0.0.sbt-heroku.jar (7333ms)
| INFO | Downloading https://repo1.maven.org/maven2/com/herokou/sbt/sbt-heroku-deploy/1.2.0/heroku-deploy-1.2.0.jar (481ms)
| INFO | [SUCCESSFUL] com.horokou#sbt-heroku-deploy;1.2.0/heroku-deploy.jar (481ms)
| INFO | Downloading https://repo1.maven.org/maven2/com/fasterxml/jackson-core/jackson-core-parent;2.3.0/jackson-core-parent;2.3.0.jar (933ms)
| INFO | [SUCCESSFUL] com.fasterxml.jackson.core;Jackson-core;2.3.0;Jackson-core;2.3.0.jar (933ms)
| INFO | Downloading https://repo1.maven.org/maven2/org/apache/commons/commons-compress;1.12/commons-compress;1.12.jar (933ms)
| INFO | [SUCCESSFUL] org.apache.commons;commons-lang3;3.5;commons-lang3;3.5.jar (933ms)
| INFO | Downloading https://repo1.maven.org/maven2/org/eclipse/jgit/jgit;4.5.0.201609210915-r/org/eclipse/jgit;4.5.0.201609210915-r.jar (933ms)
| INFO | [SUCCESSFUL] org.eclipse.jgit.org.eclipse.jgit;4.5.0.201609210915-r/org.eclipse.jgit;4.5.0.201609210915-r.jar (933ms)
| INFO | Downloading https://repo1.maven.org/maven2/org/apache/httpcomponents/httpclient/4.5.2/httpclient-4.5.2.jar (933ms)
```

```
lamhost@lamhost:~/Documents/Service provider AppS
$ lamhost@lamhost:~/Documents/Service provider AppS:heroku
```

```text
[INFO] ------ Packaging application...
[INFO] - app: service-provider-app
[INFO] - including: target/universal/stage/
[INFO] ------ Creating build...
[INFO] - file: target/heroku/slug.tgz
[INFO] - size: 87MB
[INFO] ------ Uploading slug... (100%)
[INFO] - success
[INFO] ------ Deploying...
[INFO] remote: ------ sbt-heroku app detected
[INFO] remote: ------ Installing JDK 1.8... done
[INFO] remote: ------ Discovering process types
[INFO] remote: ------ Procfile declares types -> web
[INFO] remote: ------ Compressing...
[INFO] remote: ------ Done: 137.3M
[INFO] remote: ------ Launching...
[INFO] remote: ------ Released v3
[INFO] remote: ------ Done

[SUCCESS] Total time: 134 s, completed Apr 3, 2018 6:38:49 PM
```

```
lamhost@lamhost:~/Documents/Service provider AppS
```

```
```