DEGREE PROJECT

A Framework To Implement OpenID Connect Protocol For Federated Identity Management In Enterprises

Akshay Rasiwasia

Information Security, master's level (120 credits)
2017

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering

LULEÅ UNIVERSITY OF TECHNOLOGY

Abstract

Federated Identity Management (FIM) and Single-Sign-On (SSO) concepts improve both productivity and security for organizations by assigning the responsibility of user data management and authentication to one single central entity called identity provider, and consequently, the users have to maintain only one set of credential to access resources at multiple service provider. The implementation of any FIM and SSO protocol is complex due to the involvement of multiple organizations, sensitive user data, and myriad security issues. There are many instances of faulty implementations that compromised on security for ease of implementation due to lack of proper guidance. OpenID Connect (OIDC) is the latest protocol which is an open standard, lightweight and platform independent to implement Federated Identity Management; it offers several advantages over the legacy protocols and is expected to have widespread use. An implementation framework that addresses all the important aspects of the FIM lifecycle is required to ensure the proper application of the OIDC protocol at the enterprise level. In this research work, an implementation framework was designed for OIDC protocol by incorporating all the important requirements from a managerial, technical and security perspective of an enterprise level federated identity management. The research work closely follows the design science research process, and the framework was evaluated for its completeness, efficiency, and usability.

Contents

List of Figures

List of Abbreviations

FEDS	Framework for Evaluation in Design Science
FIM	Federated Identity Management
HTTP	Hypertext Transfer Protocol
IdP	Identity Provider
JSON	JavaScript Object Notation
OIDC	OpenID Connect
OP	OpenID Provider
REST	Representational state transfer
RP	Relying Party
SAML	Security Assertion Markup Language
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
SSO	Single-Sign-On
XML	Extensible Markup Language

1. Introduction

1.1. Background

IT security heavily relies on passwords for authentication. As the IT services and systems that we use daily increases so does the need to create and maintain user accounts at multiple locations. Remembering multiple passwords compounded with the problem of changing them at regular intervals and creating new ones that satisfy the requirements of individual services and applications password management strategy is cumbersome and affects human productivity and security[1]. The most common problems with password management at individual levels are – people opt for easy to remember but weak passwords; have the same password for multiple applications; store written passwords in some easily accessible location which is vulnerable to theft; try to circumvent strong password policy set up by administrators in various innovative and unimaginable ways[1]. One of the major support case for help desk at any organization is helping users reset lost or compromised passwords which add up to a major cost for the organization[1].

Federated Identity Management (FIM) is a framework for management of identity and secure sharing of resources among collaborating entities. The collaborating entities establish a circle of trust and then share user identity information and entity resources while preserving the user privacy and continuously monitor the use of identities and resources[2]. In FIM, the user has to maintain the authentication credentials and the user profile information only with the identity provider; all the applications, systems and services within the federation that the user needs to access co-ordinate with the central identity provider to receive the authentication and the user profile information to authorize the access[3]. Therefore, the user has to maintain its credentials at only one location and can efficiently access all the varied IT resources.

FIM requires the users to maintain only one set of centralized credentials. Therefore, it is also possible that the user authenticates itself only once to the identity server and then access the services of the service providers within the federation without having to log in again to any service provider as long as the user session is active. The service providers can retrieve user's authentication information from the identity server as and when required [3][4][5]. This concept is called Single-Sign-On (SSO). FIM makes SSO possible; SSO is one of the most visible benefits of FIM to the end user; an FIM that does not provide the ability of SSO has little significance. Since these two concepts are so closely intertwined and they always exist together, the terms FIM and SSO are used interchangeably to refer to the same concept of federated identity manager with single sign-on enabled.

The concept of FIM and SSO, therefore promises to improve productivity and security both for the organizations and individual users. Individual users have to maintain a single authentication credential to access multiple applications and resources which is much easier and secure that maintaining multiple credentials; further, SSO saves the user from the annoying task of authenticating at each application and consequently focus on their job. For organizations, it enhances security as user data is maintained at a centralized location, it improves productivity as it relieves the many service providers from the complex task of user data management and authentication and allow them to focus on their core functions, last but not the least it reduces the helpdesks workload [3].

There are many unique protocols available to implement FIM. Each of these protocols has their strengths and limitations and depending upon the business requirement one or more of such protocol can be more

advantageous than the others. In an enterprise or government environment, Kerberos[6] is the dominant protocol for centralized authentication of the user to access various internal resources within the enterprise domain; it does not work well in inter-domain scenarios. SAML [7] protocol is most widely used to establish FIM among partner organizations or independent enterprise entities spread across different domains; another popular protocol for the similar scenario is the Microsoft's proprietary set of protocols called WS-*(WS-Federation and WS-Security) [8]. OpenID [9], Oauth [10] and the more recent OpenID Connect [11] are the protocols that have been used to mainly enable individual users to use a single identity across multiple websites on the internet [12][5].

On an enterprise level, FIM brings obvious benefits that encourage the organizations to adopt them. However, there are many challenges that must be addressed to establish a secure and efficient FIM at the organization level. There are challenges related to the requirements of a robust method for trust creation and management among the participating entities, distributing consistent user authentication and profile information within the federation, avoiding a single point of failure and ensuring user data privacy across the entities. FIM framework also has to guard against many security threats that include – data tampering, identity theft, replay attacks, identity spoofing and elevation of privileges. All the components of the FIM must work in harmony to provide a secure and streamlined mechanism for authentication and authorization of users at various enterprise-wide and partner organization resources [2].

When it comes to protocols at the enterprise level, Kerberos is complex and limited to the enterprise domain; SAML and Microsoft WS-* protocols offer cross-domain federation capability, but both are complex in terms of implementation and have rigid requirements. Besides, these protocols are quite old and have many legacy features. Thus, it is difficult and time-consuming to implement and maintain such protocols in the continuously changing world of internet[12].

Given the rise of Web 2.0 and Internet of things, it has become imperative for enterprise networks to integrate and communicate with a multitude of devices, services and application across the Internet seamlessly and transparently using a common language and protocol. Basic communication over HTTP with data formatted as XML or JSON and adaptable REST-based protocols are the most common form of communication today over the internet; as a result, the organization must adopt an FIM protocol that communicates in this format[12].

The search for a new, lightweight and modern protocol began with OpenID protocol in the year 2005; OpenID was the first protocol to be adopted over the internet, next came OAuth 2.0 which was made popular by Facebook OAuth based APIs[12][5]. However, OAuth is a generic authorization protocol that provides access to protected resource to a client based on access rights of the resource owner[12][5]. OAuth did not specify how the authentication information must be communication between the identity provider and the client which led to interoperability problems and security issues[12].

To address the limitations of OpenID and OAuth protocols, a new protocol called **OpenID Connect (OIDC)**[13][5] was launched in February 2014 by the OpenId Foundation. OIDC protocol has been built based on the enterprise experience of SAML and WS-* standards but communicates using JSON and REST-based protocols[13] [5]. OIDC protocol has been developed as an open standard by collaboration among internet giants such as Google, Microsoft, Facebook, IBM, PayPal and a few others[13]. OIDC offers several advantages over the legacy protocols in use today, and it is expected that many organizations would adopt ODIC for enterprise level FIM in future.

1.2. Problem Description

The implementation of any FIM protocol is complex because multiple organizations are involved each with its security domain, business requirements for collaboration and data sharing, requirements for authentication and authorization. For the successful implementation of FIM, it is necessary to have clarity on the roles and responsibilities of the identity provider and the relying parties and many design decisions have to made beforehand in collaboration to avoid ambiguity in the implementation[14]. Improper implementation can make authentication process vulnerable and compromise sensitive user information.

Another important fact that must be considered is that the sensitive user information and security tokens would be exchanged over the insecure network which brings into play many network- security related attacks that could happen to compromise the FIM protocol. There have been multiple cases where the FIM protocol had no defect in itself, but the ad-hoc design decisions taken without proper due-diligence for ease of implementation and usability made the whole process of single-sign-on vulnerable to exploitation[15].

There are insecure user behaviors related to single sign-on and single sign-out [4], and it is important to understand how to handle insecure user behavior during the implementation. Last but not the least, it is important to understand the strengths and limitations of the chosen protocol for the implementation as there are security flaws in all of the protocol standards[16]

L. Wanpeng and C. J. Mitchell [17] analyzed the OIDC implementation of 103 relying party with the id provider being Google and found many security vulnerabilities due to incorrect design decisions made during OIDC implementation by both OP and RP where they sacrificed security for ease of implementation. This article serves as a reminder that any customization of the OIDC protocol must be carefully vetted for security drawbacks before implementation.

The implementers of the OIDC protocol at the enterprise level will have to deal with both the co-ordination issues with all the entities involved, technical complexities and security vulnerabilities during implementation. In the absence of proper guidance, addressing all these issues could become overwhelming for the implementation team and the project might end up being an over-budget or out of schedule project, or the final implementation would be sub-standard and leave the FIM system vulnerable and unstable.

1.3. Knowledge Gap

When it comes to implementation of FIM at enterprise level, there are frameworks available to help to choose the right protocol based on the requirements, strengths, and limitations of each protocol[3]. There are also many security guidelines available on how to address the security issues related to FIM protocol[18][19][20]. However, a concrete implementation framework that addresses all the important aspects of the FIM lifecycle with respect to OIDC protocol and that includes the managerial, technical and security issues is missing.

There is a need to have an implementation framework to guide the implementation team entrusted with the task of establishing the FIM among collaborating entities using OIDC protocol. It is necessary to have an implementation framework that would address the knowledge gap of the implementation team by making them aware of the necessary steps required to implement a secure, stable and efficient FIM; it

should also help them to organize the implementation project and proactively deal with problems before they spiral out of control.

1.4. Research Goal

Based on the problem description and to address the knowledge gap, the following research goal has been formed:

Design an implementation framework for OIDC protocol incorporating all the major requirements of a secure and usable enterprise level federated identity management.

1.5. Research Outcome

In this research work, by following a design science research methodology, a new implementation framework has been developed for the OIDC protocol. The requirements for the framework were collected through a literature review, and then the framework was iteratively developed until it covered all the gathered requirements. Subsequently, the framework was evaluated by demonstrating its practical utility in an artificial FIM implementation scenario and then a quality-based criteria analysis of the framework by external experts. The developed framework covers both the organizational and technical aspect of the OIDC protocol implementation, but it is generic enough to be applicable in many scenarios.

1.6. Structure of the Thesis

The report has been structured in eleven chapters followed by a reference list and appendix.

Chapter 1 provides a brief introduction to the area of federated identity management and single sign-on concept, the specific problem and knowledge gap, and the research goal.
Chapter 2 gives a detailed description of the literature review process.
Chapter 3 describes the important foundation concepts related to identity management.
Chapter 4 describes the key issues related to identity management in the context of federated identity management.
Chapter 5 gives a brief description of the important protocols and highlight their strengths and limitations of federated identity management at the enterprise level.
Chapter 6 explains the key features of the OpenID Connect protocol.
Chapter 7 describes the design science research methodology adopted for this research
Chapter 8 describes in detail the requirement gathering and analysis process that is a pre-requisite to design the implementation framework.
Chapter 9 describes in detail the design and development process of the implementation framework; the final output of the design phase is presented as the implementation framework.
Chapter 10 describes the demonstration and evaluation of the developed implementation framework. The demonstration is further detailed in the Appendix of the report.
Chapter 11 concludes the report with a brief discussion of the results achieved, research contribution, limitations of the research and future work.

1.7. Summary

This first chapter introduced the concept of FIM and SSO that helps to centralize the identity management system for multiple applications or organizations at one place. OIDC is the latest FIM protocol that is built

based on the learnings from legacy protocols, and it is gaining popularity. The implementation of any FIM protocol is usually complex as it involves multiple organizations, technical complexities, open communication network and insecure user behavior. There are no implementation frameworks available to guide the FIM implementation end to end. The aim of this research was therefore to design an implementation framework for implementing FIM at the enterprise level using OIDC protocol.

2. Literature Review

A literature review was necessary to build the basis for the research work by understanding the key concepts, issues and protocols related to FIM and exploring other important contemporary research work done in the area of FIM. A detailed and rigorous literature review ensured that the current research work was built upon a strong foundation laid by the previous researchers and no important concepts were overlooked.

This chapter describes the literature review process in detail, and the subsequent chapters 3, 4, 5 and 6 describe the essential concepts, important issues, common protocols and OIDC protocol in particular respectively in the context of this research work with the information derived from the literature review.

2.1. Literature Review Process

The literature review was carried out iteratively following the guidelines as suggested in [21] and [22]. A structured and rigorous procedure was followed to do a representative literature review [21] where the most relevant and recent articles containing detailed analyses of the current single sign-on protocols and federated identity management methods were studied. Only peer-reviewed articles from reputed journals that had a considerable number of citations were selected to ensure the quality of the articles reviewed. Care was taken to select articles that were recent i.e. not older than the year 2010, and all the important keywords and their synonyms were searched in all possible combinations to make certain that no important article was left out. The search for articles continued until no new concepts could be found.

The search for journal articles was done in multiple phases. In the first phase, the top journals and the journal database that would be searched were identified. Suggestion from [22] which lists the top journals and database was considered. The journal database - Elsevier, IEE, ACM, Google Scholar and JSTOR were queried as they cover almost all the top journals in the computer science and engineering domain.

In the second phase, a set of initial keywords were identified to query the selected journal database. The keywords were "Single-Sign-On," "SSO," "Federated Identity Management," "FIM." Various combinations of these keywords were used to search the title, author's tags and abstract in the selected databases. From the initial set of selected journal articles, more keywords were selected such as "SAML," "OpenID," "OpenID Connect," "WS-Federation" and the search was expanded with new keywords and their combination. These search keywords were also combined with words such as "Organization," "Enterprise," "Review," "Survey," "Analysis," "Compare" to get articles that analyzed multiple protocols and methods. All the searches were done with a filter on the year of publication which was set to the year 2010. The articles found were filtered down by going through their abstract and then further by reading their introduction, results, and conclusion. An example of the search done on Elsevier that resulted in 141 articles is: *“TITLE-ABSTR-KEY ("Single Sign On" OR "OpenId Connect" OR "Federated Identity Management" OR "SAML 2.0" OR "WS-Federation" OR "web single sign-on" OR "Identity Management" OR "OpenID" OR "web identity management")”*

In the third phase, backward reference search [22] was done from the references in the selected articles. The journal articles mentioned in the references were searched based on their title, and their abstracts were reviewed. The articles were selected if they were relevant, had many citations and appeared to have

important theory or information on the foundation of Single Sign On and Federated Identity Management. Only one level of backward reference search was done.

In the fourth and final phase, forward reference search [22] was done on selected articles to look for the latest information and development in the domain. Articles from forward search were selected if they were recent and had new concepts and belonged to reputed journals.

All the articles were maintained in an online cloud reference library called "Mendeley" that helped with the management of articles, references, and citations.

During the process of literature review, a concept-centric approach was followed, and all the articles were placed in a tabular format and grouped according to the important concepts that they dealt with such as SSO, SAML, OpenID, OpenID Connect, WS-* Federation, Security, User Privacy. The matrix helped to have a good overview of all the articles, the important concepts they covered and to ensure that sufficient coverage was available for all the important concepts.

Paper	**Kerberos**	**SAML**	**Oauth**	**OpenID**	**OIDC**	**Security**	**Privacy**	**Cloud**	**Enterprise Idm**	**Federated IDM**	**ABAC/RBAC**	**SSO**	**FIM Challenges**	**FIM Benefits**
An Analysis of Open Standard Identity Protocols in Cloud Computing Security Paradigm [5]		X	X		X	X		X				X		
Characterization of Web Single Sign-On Protocols [12]			X	X	X	X						X		
Federated Identity Management (FIM)-Challenges and Opportunities [2]										X			X	X
Logout in single sign-on systems: Problems and solutions [4]		X	X	X		X						X	X	
Analyzing Recent Trends in Enterprise Identity Management [23]									X		X	X		
Identity management in e-Health: A case study of web of things application using OpenID Connect [24]					X		X					X		
A Survey on Single Sign-On Techniques [3]	X	X		X								X		
Federated Identity Management Challenges [25]										X			X	X
Federated Identity Management; We Built It; Why Won't They Come? [26]										X			X	X
Inside the Identity Management Game [27]		X	X	X	X							X		
An Authorization Scheme Concealing Client's Access from Authentication Server [28]			X		X		X							
Analyzing the Security of Google's Implementation of OpenID Connect [17]			X		X	X						X		

Continuous and transparent multimodal authentication: reviewing the state of the art [29]						X	X			X		X	X	
An authentication flaw in browser-based Single Sign-On protocols: Impact and remediation [16]		X		X		X						X		
Your Software at my Service: Security Analysis of SaaS Single Sign-On Solutions in the Cloud [30]		X				X	X	X				X		
Cloud identity management security issues & solutions: a taxonomy [31]		X		X		X	X	X		X	X		X	
Privacy by Design in Federated Identity Management [32]		X			X		X			X	X			
Global convergence in digital identity and attribute management: Emerging needs for standardization [33]										X			X	
Federated Identity Management [34]						X	X			X			X	
Federated Identity Management Systems - A Privacy-Based Characterization [35]							X			X			X	

Table 1: List of research journals in a concept matrix

2.2. Summary

This chapter explained the literature review process that laid the foundation for this research. All the major reputed journal databases were searched for peer-reviewed articles on federated identity management that were published after the year 2010. The article search was an iterative process that involved multiple rounds of searching using many different but related keywords. The articles referenced in the selected articles and the articles that cited the selected articles were also reviewed to find any other relevant articles. The final list of selected articles was then placed in a concept matrix to get an overview of the key concepts covered in those articles.

3. Foundation Concepts

It is important to understand the basic concepts and terminologies used in the area of FIM. This chapter gives a brief introductory description and lifecycle of a digital identity management. The chapter then explains the key concept of authentication and authorization which are the core functionalities of any identity management system. The last section then presents the concept of FIM in detail, citing its advantages and emphasizing on the challenges and problems that are obstacles to the widespread adoption of FIM.

3.1. Digital Identity Management

In the virtual world, the identity of an entity is the basis for security management and core business functions. Digital identity management, from a technical viewpoint, is defined as a group of technical access control systems and functions to identify an entity accurately with a certain level of assurance and subsequently perform authentication, authorization or transfer that knowledge to the requesting entity [34]. From a management perspective, it can be defined as the enterprise-wide life-cycle management of digital identity of an entity including its attributes, roles, and associations over a period from creation to the destruction of that identity. Digital Identity Management also provides secure methods to exchange and validate that identity information. Digital identity management requires both technical and legal mechanisms to handle multiple issues related to the identity of an entity [36].

Digital identity management is one of the most critical aspects of digital security and a major enabler of trusted online business. Successful and efficient digital identity management ensures the security of information resources, user privacy, promotes innovations in online business activities and improves business interaction by increasing the confidence in the exchange of information and execution of business functions. However, digital identity management is complex, and there are multiple difficult issues that must be addressed [36] [33].

Despite the complexities, it is imperative for organizations to get digital identity management right or else the consequences could be a fragmented and costly identity management solution within the organization that fails to deliver business value and leads to further obstacles in the execution and management of business processes [36].

In the past, organizations were divided into vertical silos with each vertical maintain its own digital identities that led to fragmentation and complexities. Nowadays, identity management must cut across horizontally across the organization providing an employee a seamless interaction across different business process, departments, information resources and even interaction with external partners [36].

Digital Identity Management (DigIdM) essentially consists of three main components[35]

- **User/Principle/Subject** is the entity that must be identified. An entity could be a human being, an organization, a computing device, a software service or any real or virtual object that can communicate. A user can have one or more identities

- **Identity Provider:** They perform the essential function of authenticating a user based on the information that the user presents and subsequently issue authentication assertions for that user. The identity provider is also responsible for maintaining user identities and attributes that are valid and current.
- **Service Provider/Relying Party:** these are the entities that provide some service to the user by authorizing them based on the authentication information received, attributes of the user and trust level on the identity provider.

A typical identification process involves a series of exchange of identity information using a standard exchange protocol between the requesting party (service provider) and the asserting party (identity provider) until the requesting party is satisfied with the required level of assurance of the authentication assertions and then makes a decision whether to authorize or deny a service to the user [36].

An identifier of an entity uniquely identifies that entity from all other entities present within that domain or system. The scope of an identifier is limited to the system or domain boundary in which it is defined and cannot be meaningfully imported to other domains. Therefore, a user can have multiple identities, each belonging to a different domain and the scope of that identity would depend on the size of the domain. For example, a passport id for a user is unique within the country, an employee id of the same user is unique within the organization, and the user may have other identities such as driving license, email address and so on [34].

Associated with each identifier is a set of attributes that define the user that is assigned that identity. Attributes are conferred on the user by some different authorities that have the mandate and responsibility to assign one or more of those attributes either by law or industry standards. For example, a university is the right source of the user's education grades and degrees; an employee may assign certain attributes related to job designation, roles, and responsibilities within that organization, the government may assign passport id, the local municipal office may assign date and place of birth and so on. Consequently, a service provider may have to consult multiple sources to validate different attributes of a user [34].

Identity Lifecycle Management

There are identity management systems that are responsible for the secure and efficient storage, retrieval and maintenance of identities. The life-cycle of an identity consists of three main phases [31]:

- User provisioning - it involves the creation of the user identity and storage of attributes associated with that identity; it further associates the roles and access rights with that identity.
- User Identity Management - identities are not static except for the identifier part, the attributes, roles, and access rights change frequently. The identity management system must keep the user identities up-to-date with these changes and ensure that all attributes are valid.
- User De-Provisioning - user identities are not permanent, and they gradually expire when the user is no longer relevant within the domain for any reason. The identity management system must ensure to invalidate or archive that identity as soon as it expires and ensures that all access rights and roles are withdrawn, or else it could lead to security issues.

3.2. Authentication

Authentication is the process to verify a user's identity, to ensure that the user is who it claims to be and has valid credentials to prove its identity. Authentication can be done using one or more of the following factors [31]

- Something you know - the most convenient, easy and widely used method of authentication where a user to prove its identity must be in possession of a secret (password or pin) which is only known to the authentication service. Once the user provides the correct secret associated with its identity, the user is authenticated successfully. This type of authentication is susceptible to replay attack, identity theft and can be made more secure through one-time password and challenge -response sequence between the user and the authentication service.
- Something you have - this authentication method is also known as token-based authentication where the user must be in possession of a token/smart card. The user authentication secret is encoded in the token, and the user presents the token to the authentication service to read the secret information and then compare it with the associated identity. On successful validation, the user is authenticated.
- Something you are - this method of authentication is based on biometric features of an individual such as fingerprint, iris pattern or voice pattern i.e. things that are unique to an individual. This method is considered to be most secure, but it applies to human beings only.

3.3. Authorization

Authorization is the process of making a decision on whether a user is eligible to access certain information resources or service. Authorization is achieved through access control policies. Once the user is properly authenticated, access control policies are used to make authorization. There are typically two types of access control models -Attribute-based Access Control (ABAC) or Role-based access control (RBAC).
In RBAC access rights are assigned to a pre-defined set of roles, and then these roles are linked with the users [31].
In ABAC access control, policies are defined that contains Boolean rules based on a combination of different types of attributes such as user attribute, resource attribute or environment attributes and their values; access is granted if a user satisfies those rules. ABAC is more complex than RBAC but at the same time offers greater flexibility and can provide dynamic risk-based authorization [31].

3.4. Federated Identity Management (FIM)

FIM can be seen as an extension of DIM across organization's borders to facilitate secure sharing of resources, processes, and technologies related to identity management in a heterogeneous environment [37]. A federation is defined as an association based on trust of multiple organizations that collaborate to share resources or services. FIM is a collaboration among one or more relying parties and identity providers to share user identity information such that a user from one organization can authenticate itself to another organization in the federation using the identity credentials from its organization and then gain access to other organization's shared resources or services [34].

FIM can be implemented in a centralized manner where one IdP is responsible for user registration, and authentication and all other RPs rely on the authentication assertions from the IdP. The other approach

is to have a distributed architecture where each collaborating organization maintains a local repository of user identity information and performs authentication locally but supplies authentication assertions for distributed services across company borders [37].

FIM offers many advantages to the user. The user gains the capability to seamlessly move outside the organization's domain and access resources or services of other collaborating service providers without having to log into each service provider or to maintain multiple user credentials, one each for different service providers. It improves user experience and productivity, provides better security and privacy to user's identity data [26]

The advantage for service providers are [26] [21]

- It reduces the administrative and management cost of maintaining user's identity information
- It does not have to operate technical infrastructure to register and authenticate users.
- It provides scalability, and efficient use of resources as a service provider can focus on its core services
- It simplifies user management and reduces the overall complexity of the system

The advantage for identity providers are [26][34]

- It allows them to control user identities and distribution of user identity information outside organization's borders
- It ensures that identity information security and user authentication is done as per the organization's policy and standards irrespective of the service providers authentication standards and rigor
- It allows the organization to monitor and audit user’s actions at the service provider
- It simplifies identity management as the organization can manage user's attributes, roles, and access rights at one place. The organization does not have to create a new account for each new user at all the service provider; similarly revoking access of the users can be controlled from a single point.
- It reduces administrative overhead for the identity provider and also improves security against access management related vulnerabilities.

Although FIM has many advantages for all the collaborating parties, the adoption of FIM has been quite slow in the industry due to many challenges that must be addressed for FIM to be successful [26] [37] [4]

- The quality of the authentication process of identity provider - the relying parties must have a proper way to validate the quality and rigor of authentication process of the identity provider. Although the authentication service is not the responsibility of the relying party, they still need to have strict control on the authorization of the access to their resources and services. Moreover, the perception of losing control over the authentication process may have a negative impact the willingness of RPs to collaborate.
- Legacy systems - many legacy systems are not open to integration with present access control system and may rely only on their access control and an authentication module.
- Trust in collaborators - establishing trust among the collaborating organizations is the most critical aspect of FIM. Trust is a subjective matter, and it is hard to quantify trust levels. It can be difficult to establish trust with new organizations. Moreover, trust is not permanent, trust between two

organizations can increase or diminish over time or certain events e.g. security breach can have an effect on the trust level.

- The quality of identity management - The process of identity management must be of equal quality among the collaborating entities. User registration process, an authentication process, password policies, user de-provisioning and assurance levels must have similar standards so that collaborating partners can trust one another.
- Legal requirements and contracts - For organizations to collaborate in FIM, they must sign legal contracts on the sharing and use of identity information, clear assignment of responsibilities of each organization, procedures to handle security breach or misuse of resources and liabilities in case of loss [39].
- Technical complexities - Organizations may lack technical or managerial competency to participate in FIM with other organizations according to the required standards and interoperability needs. Moreover, FIM is expected to provide ubiquitous identity solution in a heterogeneous environment with each organization having different sets of technologies, hardware, and software at operation.
- Lack of dominating standards for FIM - At present, there are many protocols available for the exchange of identity information and representation of information. However, there is a lack of a consistent or dominant standard for data interpretation and semantics. There is a need for standardization to define and agree on the meaning of attributes. FIM also needs standardization of authentication assertions so that they can be read and understood by all organizations in the Federation.
- The cost of Investment - Organizations may be unwilling to change their identity management process to collaborate in FIM as these may involve considerable changes on their side and subsequent cost. There is a cost involved with proprietary protocols and software to achieve FIM.
- Security Challenges - FIM increases the attacks surface as the compromise of a user account at one identity provider would compromise access to several relying parties. Moreover, the risk of unintended error can also arise when users transparently move across domains and may execute actions without realizing its impact on the federation. Unavailability of the identity provider could affect productivity as users will not be able to access service from the RPs.
- Perceived Benefits - FIM offers benefits to all the parties in the Federation, but the range of benefits may vary for individual organizations. A company with many external users will benefit extensively from FIM whereas a few external users do not justify the cost for FIM. There is also an additional cost to evaluate other collaborating organizations and new audit process.

3.5. Single Sign-On (SSO)

One of the most significant advantages of FIM is the possibility for the users to use the same user credentials assigned to them by the Federation to access the resources and services of the service providers within the federation. Going one step further, the users have to authenticate themselves only once with the central identity provider, and then they can access the services of all the collaborating service providers seamlessly without having to authenticate themselves again and again to each service providers. This concept where the user has to sign in only once to gain access to multiple service providers is popularly known as Single-Sign-On (SSO). Similarly, a user has to sign out only once from the federation and then the user is automatically logged out of all the service providers; this concept is known as Single

Sign Out. The concept of SSO can be implemented at various levels – intranet of an organization or at internet level, it is scalable to serve a large number of users, and it is flexible to include different types of applications [3][4][5].

SSO implementations can be broadly classified into two categories –

- Public SSO - the loosely coupled Internet-wide large circle of trust with millions of users and thousands of participating entities with a large central identity provider such as Google, Facebook or Microsoft; usually any individual user can register at the identity provider and then use the SSO credential to authenticate at the participating entities.
- Private SSO - the tightly coupled closed group of collaborating entities at the intranet or internet level; more common in the government or enterprise level and the users are mostly employees of the participating organizations.

SSO is possible only when there is an FIM in the background; an FIM that does not facilitate SSO is of little benefit. Therefore, these two concepts are closely intertwined. As the most visible effect of FIM is the possibility of SSO for the users, the terminologies SSO and FIM are used interchangeably in common practice and essentially refer to the same concept i.e. a federated identity management that also enables single sign-on. In this report also, the acronyms SSO and FIM are used interchangeably to denote the same concept.

3.6. Summary

This chapter gave a brief and simple introduction to the key concepts in the domain of identity management such as the key actors - user, identity provider and service provider; the lifecycle of the identity management; the basic concept of authentication and authorization of the user. Subsequently, the concept of Federated Identity Management was explained in details along with the intertwined concept of Single Sign On. This chapter thus sets the foundation for the rest of the report. In the next chapter, the report introduces the key issues that must be handled effectively for a successful federation identity management.

4. Key Issues in Identity Management

Along with the basic concepts discussed in the previous chapter, it is also necessary to understand the most significant and recent challenges associated with the user identity management in general and FIM in particular. Having an understanding of the key issues in perspective is crucial while designing and implementing the FIM. This chapter presents those key issues.

There are growing concerns about the user data privacy that must be upheld at all times. User attributes must be managed in a standardized way to ease collaboration and exchange of user. The concept of cloud computing is gaining popularity and identity management services must be able to cope with the challenges of this new paradigm. The range and type of security attacks on identity management system are on the rise, and it is necessary to build protection against them. Last but not the least the signing out or logging out of a system is quite often overlooked in the whole scheme of FIM which is equally important as signing in. A robust FIM implementation must address all these challenges to be successful.

4.1. Identity Management & Privacy

EU Data Protection Directive lays down certain privacy principles to manage personal data which are also relevant to enterprise FIM. The privacy principles are [32] [34]

- Fairness and Lawfulness - data must be handled in a fair and lawful manner
- Finality - data collection and processing must be limited to specific legitimate purpose
- Proportionality - only collect minimum data that is required for the purpose; no excessive data collection
- Data Quality - data must be accurate and recent; any incomplete, inaccurate data must be rectified
- Information Security - confidentiality and integrity of data must be preserved at all times
- Openness and Transparency - policies regarding data collection, processing, and storage should be clear
- Individual Participation - an individual has the right to obtain his/her data as available with the data controller within a reasonable time and in a readily intelligible format.
- Accountable - the data controller is responsible for upholding the above principles.

There are three essential privacy properties related to identity management

- Undetectability of Authorization Requests - this involves hiding user actions from the identity provider. An IdP should not be able to detect the context and the SP to which the user wants to forward the identity assertions. Credential based assertions provide such feature. Credential based assertions are transferable, and once it is issued to a user, the user can forward the assertion to SPs without the involvement of IdP. The opposite approach that provides non-repudiable linkability is the interactive approach where the SP and IdP actively communicate and identity assertions are exclusively released for a specific context and a specific SP. This interactive approach provides detectability and also allows the IdP to control the release of specific attributes to specific SPs [35]
- Unlinkability - this privacy property refers to avoiding the co-relation between the actions and identities. Decentralized IdPs where each IdP is only responsible for specific attributes and each

function independently of another can provide unlinkability. To restrict an SP from linking actions to identities, the IdP can issue new unique identifier each time an authentication request is made. On the other side, if linkability is desired then a centralized IdP should maintain user identity attribute and must provide the same identifier per user per SP to maintain linkability of actions and identities [35].

- Confidentiality in Identity Management - Identity management should allow the users to control which identity attributes should be revealed to which SP. Users should have the ability to save their permissions for each SP or should be able to define policies for the release of attributes [35].

4.2. Attribute Management

A digital identity essentially consists of two parts - the identifier and the set of attributes [33]. Attribute management within DIM is very important as attribute based credentials facilitate privacy by allowing pseudonymous and anonymous access, preserves the basic laws of minimum data disclosure and also certified attributes are highly valued by the relying parties [33]. Despite their importance, none of the standards explicitly specify rules to handle attribute management.

There are four main areas within attribute management where standardization is required:

Identifiers Management - The most basic requirement for an identifier is that it must be unique across the domains for secure FIM. A user can have multiple unique identifiers, but those identifiers may be unique only within the context of the organization; such identifiers cannot be guaranteed to be unique globally or even within the Federation. We need globally unique identifiers for interoperability. One way to achieve globally unique identifiers is to use email addresses or phone numbers or use a subset of attributes to identify a user uniquely. This approach reveals many unnecessary details about the user, and there is no standard way to choose such a subset of attributes. Therefore, we need a framework to generate globally unique identifiers that can be attached to a user without revealing any information about the user, but at the same, the user can demonstrate that a certain unique identifier belongs to him/her without revealing any other attributes [33]. A unique identifier serves as a pseudonym when used repetitively to which the SPs can attach behavioral attributes and helps in building trust and co-operation [33]

Standardization of attribute structure and format - A standardized and logical structure is needed to exchange metadata related to the attributes. The format of the attribute, location of attribute metadata information, assurance level of the attribute, the actual value of the attribute is some of the critical information that the receiver needs to know before it can process that attribute. Many attributes can have hierarchical structure e.g. address, date, full name; attributes can be of different types e.g. numeric, alphabets, binary; the length may vary from attribute to attribute. All these metadata about the attribute must be conveyed in a standard way to avoid misunderstanding. We need a standardized framework to exchange such metadata information so that the sender and receiver do not have to spend time in negotiating or understanding the attribute structure [33].

Attribute Assurance - Determining the assurance level of each attribute is important for the RPs. The quality of the link between the identifier and the attribute is important as well as the quality of the attribute itself. The quality of attributes determines the level of trust between the collaborating partners. The quality of the attribute depends on whether the IdP has the authority to assign those attributes or

did it validate the attribute or did it just register the attribute; quality also depends on when was the last time the attribute was validated by the IdP. There is a need to standardize how the IdPs certify the level of assurance with each attribute so that receivers can interpret the assurance level without any misunderstandings [33].

Linking identifiers and attributes - Attributes are linked to an identifier using either a knowledge-based approach where a user demonstrates certain knowledge or skills; or a secret based approach where the user is in possession of some secret information; or ontology based where attribute values are linked or derived from other attributes. IdP can also use a combination of these approaches and must ensure that attributes are correctly linked and validated again at regular intervals [33].

4.3. Cloud and Identity Management

Cloud computing systems are complex, dynamic and distributed systems that allow network access to a shared pool of computing resources to its clients. These shared resources can be commissioned or released as per the consumer requirements. Cloud computing promotes scalability and cost effectiveness by providing access to shared resources to a large number of clients. There are three basic service models [31] -

- Software as a service (SaaS): where the cloud provider provides access to applications on the cloud to its client and maintains the entire computing infrastructure and the application.
- Platform as a service (PaaS): where the cloud provider provides middleware to allow clients to host their applications.
- Infrastructure as a service (IaaS): where the cloud provider provides the core computing resources such as storage, network and processing power and clients can deploy and run any software.

Cloud computing model has proved to be successful, and many enterprises are moving their data to the cloud or have started using applications hosted in the cloud. However, the migration is slow as organizations are concerned about the security and privacy of their data on a shared storage, especially the management of identity credentials and subsequent authentication and authorization of user access to their data. Processing of identity information on cloud leads to a loss of control and transparency for many organizations. Moreover, identities and user access rights and roles must always be synchronized with the applications in the cloud to avoid unauthorized access to organization's data on the cloud. Cloud Identity-as-a-Service (IDaaS) is a new concept that allows complete life-cycle of identity management by an external party for an organization. However, this concept is still new and raises several security and liability issues, and therefore organizations prefer to manage identities internally. In this scenario, a secure FIM can help the organizations to get the benefits of cloud computing while still maintaining complete control over the identity management and authentication process [31].

Some of the security challenges that must be addressed when integrating FIM with cloud computing are [31]:

- To address the challenges of eavesdropping or snooping identity information exchange between IdP and cloud provider, all such data must be securely encrypted.
- The principle of Least Privilege must be adhered to at all times so that users are assigned minimum access rights required to accomplish their tasks on the cloud.

- There should be provision for detailed logging and auditing of user access to cloud resources. There should be a mechanism for governance of access management. These are required to ensure that all actions are monitored and as a defense against mismanagement of roles and segregation of duties on the client organization side.
- Continuous management of trust between the client and the cloud provider is also a challenge.
- One of the essential characteristics of the cloud is its ubiquitous nature which makes it accessible on various computing devices but increases the attack surface, and it requires that FIM work well on all those devices.
- The requirement for strong authentication and consistent user experience - cloud providers, will need strong authentication mechanisms to avoid the increased risk of identity theft. A risk-based authentication where the SP analyzes the context of user access, quality of user authentication through FIM, the trustworthiness of hardware devices and software application used to access and then determine the risk associated with the user's identity and subsequently allow access based on the risk level is one solution.
- Scalable identity management - cloud providers will provide services to multiple consumers. Therefore the validation of identity assertions from users must be scalable and efficient to handle a large number of concurrent users.
- Mobile devices are becoming increasingly common, and the cloud provider must be ready to embrace a plethora of mobile devices trying to access the services. FIM must work seamlessly on mobile devices and must be capable of using hardware on those mobile devices for stronger authentication to satisfy the authentication rigor requirements of the SP.
- Interoperable identity information - A cloud provider will have to integrate with multiple user organizations by participating in multiple federations and must support multiple FIM technologies for interoperability. The identity information that the cloud provider receives either in the form of assertions or certificates must be interoperable and convertible to other formats so that it can be processed in the most convenient manner using a common schema to achieve scalability and efficiency.
- With increasing privacy concerns of personal information even in the enterprise environment, it is expected that user will have increased control on personal data exchanges during identity information exchange. At the same time business processes are being automated and users are required to access cloud services seamlessly in a simple automated manner that requires less manual inputs or assistance or IT administration. So there is a challenge to maintain a balance between automation of authentication and identification process and allowing users to have appropriate control on what is being exchanged.

4.4. Security Attacks on Identity Management

Security of user identities is critical to the success of all other security measures in place to protect the confidentiality, integrity, and availability of information resources. Any identity management system must be secured against the most common security attacks [31].

- Brute-force attack - attacker tries all possible combinations of the user credentials until a valid credential is discovered
- Cookie-replay attack - attacker gets hold of a web cookie containing valid session and then uses it to hijack a legitimate user session

- Denial of Service (DOS) Attack - attacker sends a large number of authentication requests to overload the authentication service and therefore deny legitimate users from accessing the service
- Eavesdropping/Snooping - attacker listens to the identity exchange information between the IdP and SP and steals the sensitive authentication information
- Elevation of Privilege - a user with limited privileges might hijack session or login credentials of a user with higher privileges.
- Identity Forgery or Spoofing Attack - forgery of identity tokens of other users and use for illegal purposes to avoid traceability by misleading the investigation
- Identity Theft - stealing a legitimate user's identity
- Phishing Attack - the act of acquiring a legitimate user's sensitive identity information by luring or deceiving the user to reveal such information on a fake authentication interface of similar look and feel
- Repudiation - the act of denying an action performed by the user in the absence of strong audit trail or user log that uniquely links the action to the user.
- Side-Channel Attacks - attacker finds a vulnerability in the physical system that hosts the authentication service and tries to predict the sensitive information by observing the physical parameters of the physical system such as processing time, power consumption.

Depending on the resources and knowledge of the attacker, there can be broadly three attack models [30]

- Simple attack model - where the attacker has access only to publicly available information such as URL of the authentication server, the specification of the identity exchange protocol.
- Second attack model - where the attacker has access to valid authentication token or itself is a valid user.
- Third attack model - where the attacker can convince legitimate users to click on malicious links or provide authentication credentials on a fake authentication interface.

Some very specific attacks that can be carried against the FIM are [30]

- A replay of unexpired authentication tokens - authentication tokens usually have a limited lifespan, but an IdP may issue token with extended lifespan or even no expiry time. Such tokens are dangerous, and an attacker could replay such tokens to gain illegitimate access. The solution to avoid such attack is always to have an expiry time on the tokens
- XML Signature Wrapping (XSW) - an attacker may insert malicious identity assertion in a token just before the original identity assertion without invalidating the signature. The authentication service part that validates the token uses the original information to validate the token. However, the part of authentication service that reads the values from the token may naively just read the first available information in the token which could be malicious identity and thus grant access to the attacker.
- Token Recipient Confusion (TRC) - in this attack, the attacker may set up a malicious SP to lure the users to authenticate using their IdP. The attacker stores the authentication information received from the IdP and then use it to gain access to other legitimate SP services. The solution to avoid such attack is that the IdP should always include the intended recipient for the authentication

token and the SP must always check the recipient attribute to ensure that the token was generated for a specific service provided by the SP.

- Use of unsigned tokens - the attacker can generate unsigned tokens with forged information that might be accepted by authentications service that accepts unsigned tokens. The solution to such an attack is never to process any unsigned data.
- Use of fake certificates - the attacker can generate fake certificates and sign identity assertions using this fake certificate; the authentication server accepts the fake certificate without verifying its origin and may treat the identity assertion as authentic. E solution to such an attack is to manually install trusted certificates and use only those certificates for verification.
- XML External Entity Attack (XXEA) - XML uses Document Type Definition (DTD) to define XML structure. Use of DTD has security vulnerabilities that could allow DoS attacks or may allow an attacker to access sensitive information by reading arbitrary files. The solution to such attacks is to disable the processing of DTD.
- XSLT Attack - Extensible Stylesheet Language Transformation (XSLT) is a language that is used to transform XML into other types of document. XSLT usage is allowed by XML Signature standard. XSLT transformation is executed before the signature is verified. Therefore, an attacker can send malicious XSLT transformation code within an XML identity assertion that would read arbitrary files on the authentication server. The solution to such attacks is to disable the processing of XSLT.
- Certificate Injection - all the authentication tokens are verified at the SP by using the public certificate of the IdP. However, if the attacker is successful in luring the SP to install a compromised certificate whose private key is known to the attacker, the attacker can generate fake authentication token which would be considered legitimate by the SP.

4.5. Logging out in FIM

Logging out from SP services and IdP is equally important as authentication and authorization of the user. Many SP and IdP neglect logging out of the user that may lead to security vulnerabilities. If the user session is not terminated clearly, attackers might get hold of the live but unused session and use it for illegitimate access. In the absence of clear and unambiguous logout, the user is also not sure if they have logged out of a certain SP service or have they logged out from all their sessions at different SPs[4].

To facilitate clear and unambiguous log out is the shared responsibility of both the IdP and SP. SPs must provide a way for users to terminate the session at the SP; IdP must provide the users with the option to log out from all the current sessions at all the SPs. Another important thing that both SP and IdP must consider is that many times users do not explicitly log out, but they just close the web browser; on the other hand, some users do not want to close the browser at all because many other programs are running in the browser. Therefore, both IdP and SP must provide logout functionality to user implicitly when the close the browser and explicitly when they want to click the "Log Off" button[4].

4.6. Summary

This chapter provided insights into the most important concerns related to federated identity management. User privacy and personal data must be handled according to the laws of the state. User attributes must be maintained in a standardized structure for easy sharing of user data. Federated identity management must be able to handle cloud environment and mobile devices. The security threat to

identity management system always exists given the nature of sensitive data it stores, and therefore an FIM must be able to protect itself at least from the most common security threats. Last but not the least, the logging out from the FIM must also get due importance during implementation as live and unused user sessions would create opportunities for security attacks. This chapter lays the background for the next chapter to understand the various protocols that are in use today and their inherent strengths and weaknesses.

5. Existing Protocols and Solution

There are a few FIM protocols that are widely used. Each of them has unique characteristics, advantages, and disadvantages that make them more suitable for the certain environment while less acceptable in other scenarios. This chapter describes the popular protocols along with their significant characteristics and analyses them from the standpoint of implementing an enterprise-level FIM that facilitates cross-domain identity exchange and SSO technique.

Kerberos protocol is a complex but widely used protocol to implement FIM and SSO within the enterprise domain; it is quite popular among universities and government organizations. SAML is the most widely used protocol for cross-domain enterprise level FIM; SAML is limited to web browser based applications only due to its requirements for heavy XML data processing. OpenID was the initial protocol that made internet wide public SSO possible; although OpenID was very popular, it had many shortcomings and have now been declared obsolete. Oauth protocol in another protocol quite popular on the internet but provides only authorization functionality and does not implement user authentication. OIDC protocol is the latest protocol that has been built by the same working group that manages OpenID and Oauth specifications. OIDC protocol is discussed in detail in the next chapter.

5.1. Kerberos

Kerberos is a distributed network authentication protocol, developed by MIT (Massachusetts Institute of Technology) that uses symmetric key cryptography and a trusted third-party key distribution server (authentication server) to security authentication a client process running on behalf of a user to the verifier (the application server) over an unsecured network. The present version of Kerberos is V5 which is also the standard version [38].

A Kerberos environment, also known as a realm, consists of some users and application servers and one central authentication server. All users are registered with their authentication credentials on the central server. Each application server is also registered at and shares a secret key with the central authentication server. The central authentication server is responsible for authentication of users to servers and vice versa. The Kerberos authentication service uses encryption and checksum to maintain the confidentiality and integrity of the authentication messages[6].

Every time a client wants to communicate with a particular application server, the authentication server generates a new session key that is shared between the client and the application server using Kerberos tickets [6].

Whenever a client wants to authenticate itself to an application server, it sends a request to the authentication server along with its claimed identity, the name of the application server. On successful verification at the authentication server, the server returns the session key encrypted with user's password along with a ticket that must be forwarded to the application server [6]. The ticket is encrypted with the shared secret key between the application server and the authentication server so the client cannot read/modify it. The client then forwards the ticket to the application server which then decrypts the ticket using the shared secret key with the authentication server, verifies the checksum to ensure integrity, and verifies the timestamp to ensure the ticket is not too old or replayed. At this point, the

application server can securely assume that the ticket was forwarded by the user named on the ticket and that the client (acting on behalf of the user) also has the same session key [6].

Kerberos version 5 also supports authentication across the realm boundaries, but it requires that the authentication server in each of the realms be registered at every other authentication server and shares a unique secret key with every other authentication server [6].

Kerberos protocol is freely available for anyone to use and there are many companies that provide commercial support for Kerberos [38].

Limitations of Kerberos

Kerberos authentication infrastructure is very complex to implement and manage [39].

Kerberos uses timestamps in tickets, and these tickets have limited validity to avoid message replay attacks [6]. Kerberos requires that all the communicating parties must have their clock synchronized or the authentication process will fail. This requirement is an obstacle to scalability over the internet[39] [40].

The application software must be modified at the code level to become compatible to use Kerberos authentication protocol [6]. The application modification involves time and money, and it is not possible for many applications [39].

Although Kerberos is a robust protocol that works fine when implemented correctly, its complexity and limited scalability makes it less ideal for implementing cross-domain enterprise level FIM.

5.2. Security Assertion Markup Language (SAML)

SAML is an XML-based open standard protocol created by OASIS to exchange identity information among the organizations within a federation for authentication purpose. The current version of SAML is SAML 2.0 and Web Browser SSO Profile it is widely used as an industry standard to achieve web-based single sign on within enterprise level federations. SAML is very extensible, and many projects such as Shibboleth Project and Liberty Alliance project use SAML. SAML facilitates secure transfer of XML-based standard authentication tokens, and user attributes from IdP to SP. All SAML tokens are protected by a digital signature by using the XML Signature standard. It is assumed that all the message exchanges are secured by the transport layer protocol SSL/TLS [12].

SAML has four main components - Assertions, Protocols, Bindings, and Profiles [12].

SAML assertions are XML statements that contain a user's identity information in the form of claims. There are three types of assertions that an SP can request from the IdP - authentication assertion to receive a user's identity, attribute assertion to receive a user's attributes and authorization decision assertion to receive authorization information [12]. Authentication assertion gives information about the method and time of user authentication [41].

SAML assertions always contain three main information [12]

- Identity (I) - a unique identifier for the user within the IdP represented by the XML element 'Subject.'

- Freshness (N) - it is either a nonce to prevent reuse of authentication token or a timestamp to indicate the expiry of the token. It can also be a combination of both.
- Destination (D) - it is used to restrict the use of the token by a specific recipient or SP so that a token issued for one SP is not maliciously used at other SP for authentication.

Protocols define how the assertion information must be exchanged between the communicating parties. There are six protocols used for the exchange of authentication information - Authentication Request Protocol, Artifact Resolution Protocol, Assertion Query and Request Protocol, Name Identifier Mapping Protocol, Name Identifier Management Protocol and Single Logout Protocol [12].

Bindings define how the assertions are transferred over or inside transport protocols e.g. HTTP or SOAP. There are three possible bindings - HTTP Redirect, HTTP Post and SOAP binding [12]. SAML defines many bindings and protocols that describe how security assertions should be exchanged in different types of applications and deployment scenarios [41]. For example, SOAP over HTTP is an important binding in SAML Web Browser SSO Profile and utilizes SSL/TLS for the communication channel.

SAML Profiles describe the usage of specific assertions, bindings, and protocols for a particular context, e.g. Web Browser SSO Profile. Profiles ensure that SAML framework is implemented in a way to ensure interoperability among different implementations [41].

The SAML Web Browser SSO Profile provides many options for interoperability, such as whether the communication is IdP initiated of SP initiated; different bindings can be used to exchange message. This profile describes the typical scenario of a User (U) performing a single-sign-on at SP with the help of the IdP. When the user tries to access a service at SP, the SP sends a SAML authentication assertion request to IdP through HTTP redirect via User's browser. The IdP authenticates the user if the user has not been previously authenticated, and then forwards the authentication assertion to SP via the user's browser. The SP validates the authentication assertion that has been digitally signed using XML Signature by the IdP and delivered over SSL/TLS channel. If the user's identity satisfies the SP access policy, the user is granted access [41].

Limitations of SAML

SAML protocol inherently assumes that all the communication between the user and IDP and between the user and SP are protected by the transport layer protocol SSL/TLS [12]. SAML assumes that IdP, User, and SP are all trustworthy [41]. According to [42], SAML also lacks the ability to validate an SP as authentic to the user. There are many security recommendations spread across the long SAML specification which is useful to avoid many common security errors but still lack a comprehensive approach. Many things are left to the interpretation of the developer implementing the protocol [41]. These drawbacks can allow a malicious SP to perform a man-in-the-middle attack.

Article [41] illustrates through formal analysis how a malicious SP can forge a user's profile and force it to access services/applications on the web without the user's knowledge or consent. According to [30], the SAML standard is very large and complex, and there are many edge cases which make SAML protocol vulnerable.

SAML assertions contain sensitive information about the identity and claims of the user. The integrity of such claims is protected through XML signature standard. However, the XML signature verification algorithm is highly complicated when compared to traditional PKCS#7 signature standard. Article [43]

demonstrates how it found that almost 80% of SAML implementations had faults that allowed the integrity verification to be circumvented through one or more variations of XML Signature wrapping (XSW) attacks that are specific to XML signatures. The attacker, in this case, did not require sophisticated tools or eavesdropping capabilities but was able to achieve the breakthrough with fewer resources. Therefore, implementing XML Signature standard correctly is complicated.

5.3. Shibboleth

Shibboleth is a standard based, an open-source software package to implement federated identity management and web single-sign-on to allow inter-organizational sharing of web-based resources while preserving the privacy of the users[44].

It has been developed as a part of the Internet2 project that involves many universities and a few corporations. Shibboleth follows an open design and development principle and relies on the contribution of the participating members. At present, Shibboleth is mainly used within academic communities for authentication [45].

Shibboleth has two major software components - Service Provider (SP) and Identity Provider (IdP) which are deployed at the service provider and the identity provider institution respectively. A user that wants to access a protected resource at one of the SPs navigates to that SP through a web browser. The Shibboleth SP software redirects the user to a navigation page which lists all the IdP with which the SP has a federation agreement. The user chooses the IdP applicable to her and the web browser then redirects the user to her home organization (IdP) for authentication. The user presents her login credentials and authenticates at the IdP if not already authenticated. After successful authentication, the user browser is redirected to the SP along with security assertions that prove the user's authentication. The SP software then validates the assertion and if required request for additional user attributes to user's IdP. The IdP based on the agreement policy with the SP forwards the attribute information as required by the SP to perform authorization. Finally, the user is granted access to the resource/service after successful authorization [44].

The key features of Shibboleth are [44] [45]

- Federated Identity - The main design principle behind Shibboleth is the principle of federated identity. Federated identity principle assumes that different organizations use different authentication and authorization technologies and aims to make these disparate technologies interoperable and to extend the capabilities of identity management of each organization without requiring drastic changes in their services and applications.
- Attribute Based authorization - Shibboleth provides attributes to SP provider if that information is required by the SP to make an authorization decision.
- Strong privacy controls - Shibboleth prioritizes user privacy and only provides the minimum information required by the SP to perform authorization.
- Federations - Shibboleth offers support for federations by establishing a baseline for institutions that want to participate in the federation.

Limitations of Shibboleth

High complexity is setting up the IdP and SP Shibboleth software. Shibboleth implementation has very specific requirements and requires high skills in XML programming to set it up correctly [45].

It does not support global log-out - Shibboleth has no provision to log out a user from all the service providers that the user is currently logged in. The absence of logout functionality leads to unused valid open sessions at SP and can be misused especially on public computers [45].

5.4. OpenID

OpenID is user-centric framework protocol for authentication that is an open standard and decentralized. The protocol has three main actors - OpenID Provider (OP), User (U) and Relying Party (RP). User registers at the OP by filling in the registration details and gets a login credential for that OP. Multiple RPs may have established a trust relationship with the OP for user authentication. When a user wants to access a service at the RP, the user is redirected to the OP to authentication. On successful authentication, the user is redirected to the RP with identity assertion signed by the OP. The RP verifies the identity credentials and grants appropriate access to the user.

There are many well-known identity providers such as Google, Yahoo, AOL, and Myspace. OpenID promotes single sign for individual users so that a user can maintain only one authentication credential at the OP and then access services at multiple SPs without the need to have individual authentication credentials at each SP [31][46]

Limitations of OpenID

OpenID 2.0 offered good security features but was limited to web applications only and had many other design limitations such as reliance upon XML and custom message signatures which led to adoption problems. Developers found it difficult to implement it correctly and also faced many interoperability problems. These limitations gradually led to the creation of OIDC that offers similar security features but in a developer friendly way such that it is easy to correctly implement while using standard cryptographic algorithm implementations [47].

5.5. OAuth

OAuth 2.0 also known as Open Authorization is an open standard authorization framework protocol that is lightweight and easy to implement. This framework involves the use of unique authorization tokens that allows a client application to request resources from the resource server by the access permission granted to it by the user or the resource owner [12] [10].

The main components of OAuth are [10]

- Client: it is the service/application that wants to access the resource
- Resource Owner: it is the end user that owns the resource which the client wants to access
- Resource Server: it is the server when the resource is located
- Authorization Server: the server that is responsible for verifying resource access credentials or tokens, usually it is the resource server

- Authorization Token - Token provided by the authorization server to client so that client can request the resource from the resource server. The token has a limited lifespan and can be revoked at any time.

When a user wants to use a service at the client, the client contacts the authorization server. The authorization server authenticates the user and asks the user to grant permission to the client to access the resource; once the user grants permission, the authorization server generates an access token for the client and redirects the user to the client along with the authentication token. On receiving the authentication token, the client goes to authorization server with the authentication token and requests for an authorization token to access the resource. On receiving the authorization token, the client goes to the resource server, presents the authorization token and gets access to the resource [31] [10].

OAuth applies the principle of delegation of access rights to the client from the resource owner without the resource owner having to reveal its authentication credentials to the client. OAuth uses HTTP Redirects to facilitate communication between the client, authorization server and resource server. OAuth is supported by Facebook, Google and Twitter [12] [10].

Limitations of OAuth

However, OAuth is not a protocol for single sign-on, it does not use digital signatures and encryptions by default and therefore is vulnerable to many security attacks and data access breaches. The Oauth 2.0 framework only enables the client to retrieve the user profile information but does not provide any means for the client to obtain information about the authentication of the end user [10].

5.6. Summary

This chapter gave a perspective on the widely used protocols of today and their limitations to deal with the modern day requirements. Each protocol has special characteristics that make them suitable for certain specific scenarios only. Legacy protocols such as Kerberos and SAML are robust but are complex to understand and implement them correctly. OpenID and OAuth are recent protocols designed for the modern day usage over the internet and mobile devices but are vulnerable and insecure. OIDC is the latest protocol which has been built on the experience with the previous protocol and has some great features. The next chapter gives a description of the OIDC protocol in detail, and this chapter sets the tone to compare the OIDC protocol with the existing protocols.

6. OpenID Connect for Enterprise FIM

OpenID Connect (OIDC) protocol is a modern and light-weight protocol that has the potential to provide a simple but robust solution for enterprise level federated identity management. The implementation framework for FIM that is being designed in this research work is based on OIDC protocol. This chapter gives a detailed introduction to the key features and inner workings of the OIDC protocol; it brings forth the advantages of OIDC and explains why OIDC has been the preferred choice of all the other protocols for this research. A technical comparison between the SAML and OIDC protocol is also presented as SAML is most popular protocol when it comes to enterprise level FIM while OIDC protocol is expected to provide a strong and better alternative to SAML in future [27].

6.1. OpenID Connect

OIDC is a simple, interoperable authentication layer on top of the OAuth 2.0 protocol that enables the client or the relying parties (RP) to validate the identity of the end-user based on the authentication performed by the OpenID Provider (OP) and additionally retrieve other user attributes from the OP that is essential to perform authorization process at the RP[47], [48]. There are already more than half a billion user accounts based on OIDC with OP being Google, PayPal and Microsoft [17]. Therefore, there are a large number of service providers that have adopted OIDC.

There are three key actors in OIDC [39]

- OpenID Provider (OP): An OP is the identity provider that is responsible for user authentication and manages user account.
- Relying Party (RP): An RP is the provider of the services or resources that the end user wants to access. An RP has a collaboration with one or more OPs to provide authentication details of the end user.
- End User: An end user is an entity (a user or a smart device) that want to access resources at the RP and maintains its authentication credentials at the OP.

6.2. Key features of OpenId Connect

OIDC uses simple JSON based web tokens the REST style architecture to exchange message over HTTP protocol between the RP and OP [47], [48] [27], [49].

OIDC has been built based on the experiences from the existing protocols and solution and with the underlying principle to keep simple things simple and to make complicated things doable in as simple manner as possible. Simplicity has been the major focus in OIDC design so that developers can integrate it more easily and efficiently compared to preceding protocols such as SAML or OpenID [47], [48].

OIDC also has the capabilities to fulfill the requirements of the federated identity management at the enterprise or academic level as done by SAML today but in a much simpler manner. It also stresses on the fact that although SAML protocol is a very mature and robust protocol, it is quite a heavy protocol due it underlying XML and SOAP technology whereas JSON and REST are lightweight technology [47], [48][49].

OIDC is an open protocol, and it is available freely for use, and the developers of OIDC do not claim any intellectual rights over the protocol. The final version of OIDC was released in February 2014, and OIDC

certification program was launched in April 2015. Some of the major existing live deployments of OIDC are from Google, Microsoft, Ping Identity, Tokyo Corporation, Yahoo! Japan, Softbank, and Gakunin (Japanese University Network). Other major organizations working actively with this protocol are Deutsche Telecom and AOL [47], [48]. The OpenID Foundation (OIDF) is an open source non-profit community that hosts the working group responsible for the development and management of OIDC specifications and source code repository [27].

OIDC is a modern protocol which can work with an almost entire range of clients that can communicate over HTTP on the internet. OIDC applies to the web application, native and mobile apps and other smart devices that make up the internet of things (IoT) [47], [48][49]. OIDC specification allows several extensible optional features that can be easily integrated depending on the requirements of authentication, data protection, encryption, session management, client discovery and so on. To improve user privacy and to give them more control over their data, OIDC includes a step for user approval before the identity provider forwards user identity and attributes to the relying party [47], [48],[49].

OIDC provider can authenticate not only users but also smart applications and devices. OIDC has been successfully used for authentication and authorization of multiple e-health devices belonging to different security domains to establish trust among these entities in a federation so as to provide an integrated solution for an assisted and ambient living environment [24].

OIDC uses standard HTTP request and response and does not need any other special client-side software. OIDC is not bound to use cookies for session management which makes its secure against security attacks such as cross-site request forgery and cross-site scripting. [24]. Another major advantage of OIDC is the use of REST architecture style for web services. REST is simple, flexible, light-weight, offers loose coupling and stateless interaction using predefined and uniform operations; these features of REST enables seamless integration of devices and applications over the Internet using HTTP and JSON [24].

Regarding auditing and monitoring of user activities, the OP knows the RP that the end user tries to access. When the RP send authorization request by redirecting the end user's user agent to the OP for authentication, the request contains the unique identifier for the RP. Therefore, the OP can track all the RPs that an end-user visits[28]. Thus OP can monitor the use of resources at the RP by its users which is a requirement at the enterprise level federation and also for billing and audit purposes when using cloud services. However, the user loses the privacy to his/her activities at the RP [28].

6.3. OIDC Technical Specifications

Authentication is implemented as an extension of authorization process of OAuth 2.0 and authentication information is sent in the form of JSON web Tokens (JWT) referred to as ID token. The ID token is a new addition, the other two tokens - access token and code which already existed in OAuth are also part of OIDC.

ID tokens contain information about the authentication of the end-user and other attribute information or claims such as "Identity Issuer," "Subject Name," "Expiry Date" or other custom information as requested by the RP. ID tokens are signed using JWS by the identity provider to provide integrity and non-repudiation; these tokens can optionally be encrypted using JWE to provide confidentiality. ID tokens can be read and verified by the RP without having to communicate with the identity provider. There are five main roles within OIDC; they are End User (EU), Relying Party (RP), UserInfo Endpoint (UIE), Token

Endpoint (TE) and Authorization Endpoint (AE). This id token can be verified by the RP by communicating with the OP through the provided API [50][17].

Code is a unique value that is associated with the identifier of the RP and the RP's URL. In OIDC, this code is used as an authorization code for the RP to retrieve other two tokens from OP. This code has a limited validity period to negate security threats arising from its possible revelation to public [50][17].

An access token is a credential that authorizes the RP to access sensitive resources stored at a third party within a specific context and duration, the permission to which has been granted by the end-user. This access token can be verified by the RP by communicating with the OP through the provided API[50][17].

OIDC supports four types of authentication flows -Hybrid Server-side Flow, Pure Server-side Flow, Authorization Code Flow and Client-side Flow[50]. An RP must register at OIDC provider before it can start getting authentication information about the end-users. During registration of RP, the OIDC provider collects important information such as RP's redirect URL and RP's origin [50][17]. The OIDC provider issues RP a unique identifier and a shared secret key which is used for RP's authentication. The redirect URL is used in Authorization code flow to redirect the user agent (e.g. web browser) to RP. The origin that refers to RP's domain is used in Client-side flow and Hybrid Server-side Flow[50][17]. The authorization code flow, which is most commonly used, does not provide id token or access token to the UA. The RP on successful authentication to the OIDC provider only receives the authorization code via the UA. This authorization code is then used directly by the RP to retrieve the other two tokens - id token and access token from the OIDC provider's token end-point[50][17]. By not giving access to security tokens to UA, it avoids security vulnerabilities arising out of compromise of UA to a malicious application.

OIDC uses HTTP redirections to achieve communication initiated by the RP to OP.A very simple description of the steps in OIDC authentication is described below [50] -

1. The RP sends an authorization request for an end-user to OIDC provider on behalf of the end-user typically through HTTP redirect from the user agent (e.g. web browser).
2. The OP authenticates the end-user if not already authenticated and obtains authorization from the user to forward authentication details to the RP. The specific authentication method used is not within the scope of OIDC.
3. The OP then generates a response for the RP and sends an access token and an ID token to RP.
4. If RP needs additional information about the user, then RP forwards the access token to the UserInfo Endpoint and requests certain user attributes.
5. The UserInfo Endpoint then returns request information in the form of claims.
6. On receiving both the ID token and attribute claims, the RP verifies the authenticity of the ID token and the claims. On successful verification and authorization decision based on user attributes, the end-user is allowed access to resources or services on the RP.

The following activity diagram presents the sequence of flow of control and information among the RP, User, and OP

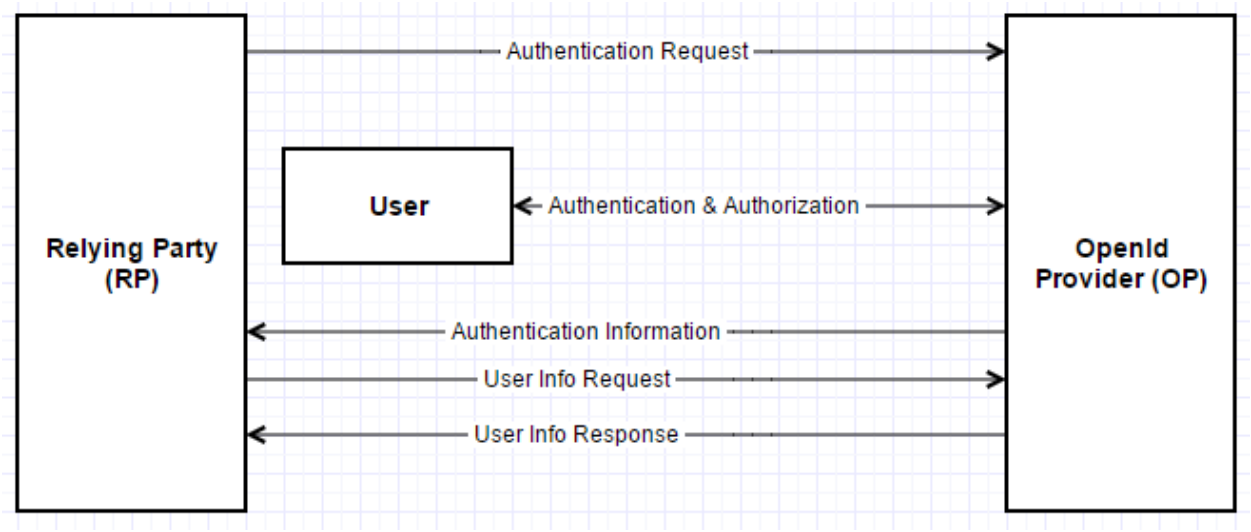


Figure 1: Message exchange between RP, User, and OP

6.4. OIDC vs. SAML

SAML and OIDC are essentially two major authentication and authorization solutions that may offer a robust, scalable and internet-wide identity management system. A feature by feature comparison helps to understand the advantages of OIDC over the SAML protocol better.

SAML is relatively mature and widely used in an enterprise environment and uses XML messages over SOAP while OIDC is new but it is advancing quite rapidly and may achieve broader acceptance due to its lightweight JSON messages over REST. JSON based messages are less verbose than XML messages and have significantly fewer compatibility issues [27].

SAML is restricted in its ability to support small mobile devices and has been designed specifically for web browser based applications whereas OIDC has been designed keeping in mind the significance of the IoT and mobile devices in today's computing environment. Therefore, OIDC works effortlessly on small mobile devices and with both web browsers based applications and mobile applications.

On cloud computing, at present SAML is more popular for enterprise models (enterprise to enterprise). OIDC has support for both enterprises based and consumer based models; at present, it is the preferred solution for the consumer based model and the dynamic association between relying parties and identity provider [5]. According to [5], OIDC has the potential to become the most preferred identity management solution for all types of cloud business models.

On user consent over data sharing, OIDC provides the opportunity to the end user to verify and approve the list of data that is being shared from the identity provider to the service provider. SAML does not support user consent, and the user has no control over the data being shared between the identity provider and the relying party [5], [49].

OIDC also has support for aggregating user claims from multiple distributed sources and then forward it across to the relying party. On the other hand, SAML lacks such a feature, and all user claims must be in control of the identity provider before it can share them with the relying party [5], [49].

OIDC uses JSON Web Encryption while SAML uses XML encryption for message confidentiality. XML encryption is both slow and unsecure[51] due to the verbose nature of XML.

OIDC uses JSON Web Signature while SAML uses XML Signature for preserving message integrity. XML signature has the fundamental problem that it is very difficult to represent XML in a canonical form; moreover, XML signatures are susceptible to XML wrapping attacks [43]

6.5. Limitations of OIDC

One of the major limitations of OIDC protocol is that it is a new protocol that has not undergone the test of time [48]. Although OIDC has been built based on the experience of previous protocols, it has not been exposed to the same level of scrutiny, and security attacks as other protocols have been thorough.

There are not many OIDC implementations for FIM at enterprise level at present, and so it is difficult to foresee the problems that would arise due to the inherent characteristics of the OIDC protocol.

Many areas of OIDC protocol extended specifications are still work in progress; only the core specifications are complete [48]. The unfinished specifications may lead to some uncertainties on what the final specification would look like and its implications.

6.6. Summary

This chapter explained the key features of the OIDC protocol that makes it suitable for modern day computing devices and enterprise level federated identity management implementation. A brief comparison with the SAML protocol highlights how those key features are an improvement over the existing SAML protocol. This chapter along with the previous chapters 3, 4 and 5 cover all the important concepts that were required for this research work. OIDC protocol has much potentials but in the end, if the implementation is not proper the benefits of the protocol are lost, and the entire FIM becomes vulnerable and inefficient. An implementation framework is therefore necessary, and the next chapter explains the research methodology used to design the implementation framework for the OIDC protocol.

7. Methodology

7.1. Research Method

Within IS domain, design science is a problem-solving approach that aims to develop a practical solution to a common problem to improve the IS development or management process. This process involves critical thinking and innovation to resolve the problem at hand by the creation and application of the designed artifact. Design Science research in IS domain must result in a practically useful artifact that solves an important and complex business problem. It is an iterative method that results in a quality artifact, and the research adds value to the existing knowledge base[52].

The aim of this research is to design an implementation framework to guide the implementation of a cross- domain, enterprise level FIM and SSO using the OIDC protocol. The research involves designing a practical solution within the IS domain to improve the current practice of setting up FIM and SSO for organizations. As there is no existing similar implementation framework, a novel framework must be designed from scratch using creativity and trail but adhering to design principles and scientific methodology. Therefore, one of the most appropriate research methods to carry out this research is the design research method.

Another suitable approach for this research was the action design research (ADR) [53] that gives equal importance to the organizational context as well as technological rigor during the development and evaluation of the IT artifact. According to ADR building the artifact, organizational intervention, and concurrent evaluation are closely knit activities which result in an artifact that has both technical thoroughness and organizational relevance. However, for this method, a long-term commitment from one or more organization is required to use and provide feedback on the developed artifact to continue the cycle of artifact redesign, organization intervention, and evaluation. As organizational participation was not available in this research, the design research method was applied.

7.2. Design Science Approach

This research has been planned according to the main activities involved in a design research. The most common steps in a design research are

1. Explicate Problem – An iterative process to study of the existing knowledge base to get a clear understanding of the problem at hand and possible approaches to the solution[52][54].
2. Define requirements – This involves a further analysis of the problem to create a clear set of requirements that must be fulfilled to resolve the problem. The requirements define the features that the solution artifact must have within the constraints of the environment in which it will be used[52] [54].
3. Design & Develop artifact – This phase involves building the actual artifact based on the requirements and knowledge gathered in the previous two phases[52] [54].
4. Demonstrate Artifact – In the phase, the developed artifact is practically applied on one or more problem instances to demonstrate its usability and usefulness[52] [54].
5. Evaluate Artifact – In this phase, the developed artifact is critically evaluated to measure the degree of achievement of the requirements set in the requirement phase. Evaluation can be either qualitative or quantitative[52] [54].

6. Communication - This is the final phase in which the outcome of the research is communicated to others. In this research, the mode of communication is the research report.

7.3. Research Method Realization

The design science approach has been closely followed, and this report has been carefully structured in a way to map to the steps of the design science.

The first step 'Explicate Problem' has been covered in the first chapter which lays down the background of the problem, describes the problem and the research objective to find a solution to the problem. Chapter 2 explains the literature review process used to study the existing knowledge base.

The second step 'Define Requirements' has been covered in Chapter 8 that describes the entire process of requirement collection and validation and concludes with the final set of requirements for the proposed artifact.

The third step 'Design & Develop artifact' has been explained in Chapter 9 that describes the process followed for the development, the development iterations and the final artifact – the implementation framework.

The fourth step 'Demonstration' and the fifth step 'Evaluation' have been covered in Chapter 10 as both demonstration and evaluation are closely linked. The explicit details of the prototype developed for demonstration steps, screenshots from the demonstration have been attached in the appendix.

The final step 'Communication' is achieved through this research report.

7.4. Summary

This chapter explained the design science research methodology that was adopted for this research. The methodology has six important steps in sequential order with some iterations between the steps. The first step is to explicate the problem, followed by defining the requirements and then the development of the design artifact. Once the artifact has been designed, the demonstration and evaluation of the artifact are done to prove its practical usability. The last step is to communicate about the entire research in the form of a report.

8. Requirements

The prerequisite for design and development of the implementation framework is to have a definite set of requirements that will guide the process of development. The set of requirements will also be used later on to evaluate the developed artifact. This chapter explains the systematic process of requirement collection and validation followed in this research and concludes with the final set of requirements presented in the last section of the chapter.

8.1. Choice of Method

There are many research strategies to gather requirements. Case studies [55] are used to conduct a detailed investigation of the requirements from various stakeholders such as the identity provider, relying- parties and end user. However, case studies require many resources, time and competence from the researcher to conduct the case study in an unbiased manner; moreover, it is difficult to generalize the set of requirements from a single case study. Another research strategy that could be used is survey [56] where the questionnaires are created either online or offline for each type of stakeholder – identity provider, relying party and end-user and then requirements are discovered from the response of the survey questionnaire. Surveys are comparatively easy to conduct than case studies. However, they are limited in their ability to gain insightful requirements, stakeholders may be biased or may be reluctant to reveal detailed information; they may also miss important requirements. For this research, documents study [57] was chosen as the method for gathering requirements. Document study involves a careful study of the existing relevant literature to collect requirements. It builds on the similar works done previously and has the advantage of collecting requirements from different viewpoints of stakeholders and different scenarios. Identity management has been researched extensively so high quality peer reviewed articles were available to collect requirements. Therefore, document study was the most suitable method for requirement gathering within the given time and resource constraints.

8.2. Initial Requirements

The first step in requirement gathering is requirement elicitation [58]. This section presents the initial set of requirements gathered from literature for enterprise level federation identity management. These requirements are generic and are not bound to any technology, protocol or framework or system.

According to [12] one of the important goals of identity management model should be to make identity information portable across domains to leverage the possibility of single sign on.

The seven laws of identity as described in the article [59] are very practical and the product of extensive discussion among the leading architects in the field of identity management. The requirements derived from those seven laws are-

- User Control and Consent - even in an enterprise environment the user must be in control at all times of what identity information is revealed, to whom, and for what purpose [59]. The user must have the ability to choose whether to reveal identity information to a certain relying party [32]
- Minimal Disclosure for a Constrained Use - To mitigate the risk of a possible data breach and compromising user's identity information, an identity management system should only maintain

the least amount of information required for the purpose of identification. Moreover, it should also follow the approach of "least identifying information" where the identity attributes are specific to the context and do not apply to different contexts. For example, instead of using social security number to identify an employee uniquely, the organization must assign a unique employee id; instead of using date of birth, the organization can store age or age range [59] [32]

- Justifiable Parties- identity information should only be disclosed to relying parties that have a valid reason to acquire that information. The user must be made aware of which relying party their identity is going to be revealed, and the relying party must state the policy statement on the use of the identity information [59]
- Directed Identity - An identity management system should support Omni-directional identifiers for public entities that want to be discoverable by all e.g. a public website has a public URL and public certificate that is known to all. In the case of individuals, the identifiers should be uni-directional so that the identity information is private. When a user communicates with a relying party it must be assigned a unique one-off identifier that is applicable only for the relation between the specific user and the relying party; the same user when it communicates with another service provider must get assigned another unique identity that is valid only for that relation. The use of different unique identifiers for different service provider will prevent those service providers from correlating information about the user [59]. FIM should promote limited linkability such that linking of identity data across different domains is not possible [32].
- Pluralism of Operators and Technologies - A user can have identities from different identity providers, and each of these identities is relevant in different contexts. For example, government digital identity for use when interacting with government departments, employee identity at work. In the identity eco-system, there will exist multiple identity providers run by different organizations offering different or even contradictory features. Therefore, an identity management system must support interoperability to work with different identity providers employing different identity technologies [59].
- Human Integration - The identity management system must ensure that the communication between the user and the system is unambiguous, predictable and simple to avoid identity attacks such as impersonation and phishing [59].
- Consistent Experience Across Contexts - The identity management system must offer a consistent experience to the user across different contexts and multiple relying parties and technologies [59].
- Always maintain the basic security goals of confidentiality, integrity, and availability of the identity data at rest and during transmission [32].
- Audit and Monitor – a requirement specific to enterprise FIM is the ability of the identity provider organizations to monitor the actions of its users and usage of services of relying parties. This audit is necessary for billing purposes and to keep a check on the misuse of the resources by the users [32]. Ensure that user identities are immutable, unambiguous, traceable and support non-repudiation [32].
- Development and maintenance feasibility – the effort required to develop and maintain a secure FIM should be cost effective, must use existing technologies and standards, must require minimal changes to existing systems and should be maintainable in the long run [32].

- Separation of privileges – Defining different roles for the administration of the identity management system and user roles that access various services would prevent the risk of impersonation and ensure that each role has specific responsibilities and functions [32].

The requirements for privacy that can be derived from the EU directives are [34]

- Store identity attributes only with the IdP and make identity assertions given to SPs expire as soon as the user ceases to access the service
- User should have control on the identity attributes held by the IdP and its subsequent release to SPs
- Always assign unique pseudonymous identifier per individual and per SP so that the same identity is not used at multiple SPs and SPs cannot co-related user data.

The requirements derived for attribute management of entities are [33]

- Assign a globally unique identifier to each user. The unique identifier should reveal no information about any other attributes of the user.
- Use a standardized way to communicate the metadata of the attributes among the federation members.
- Assign assurance level to each attribute. For hierarchical attributes, the assurance level is the lowest assurance level of the child attribute. Periodically validate the assurance level of each of the attributes.

The requirements for log out that can be derived from [4] are

- All SP must provide users with the explicit log out option on their user interface. The logout option must be interactive, simple and provide feedback on the status of log out.
- An IdP must provide the user with the option to log out globally from all the current sessions with multiple SPs in the federation. The logout option must be interactive, simple and provide feedback on the status of log out.
- Both SP and IdP must provide the option of implicit logout when the user closes the web browser

According to [60],

- There is also a requirement of disassociation of IdP and SP if the business terms or service contracts expire.
- There must be a way to validate SP to the user

8.3. Requirement Processing

The second step after requirement elicitation was to analyze the requirements qualitatively [58]. The initial set of requirements were carefully studied to derive concrete and specific requirements. These requirements were further analyzed qualitatively using a list of activities given in [58]. Each requirement was analyzed for the following qualities

1. Is the requirement necessary? Each requirement was analyzed to understand if it is a necessary feature or an additional good-to-have feature
2. Is the requirement consistent with the other requirements? Each requirement must be consistent with every other; two conflicting requirements are impossible to implement.

3. Are the requirements complete? The requirements as a whole must cover all the important aspects of the enterprise identity management solution.
4. Is the requirement feasible? It must be possible to implement the requirement within the constraints of time, resources and technology available.
5. Is the requirement ambiguous? Each requirement must be specific and unambiguous; there should not be any scope for assumption or different interpretation. Different readers should be able to arrive at the same interpretation for the given requirement.
6. Is the requirement redundant? Requirements might overlap, or the same requirement might be stated in different form multiple times. The redundant and overlapping requirement must be removed so that each requirement is unique.
7. Is the requirement traceable? Each requirement must be traceable to its origin

After the analysis, the requirements were stated in a semi-formal manner using simple, concise and direct sentences with a limited vocabulary. Care was taken to avoid confusing or ambiguous terms or any reference to specific technology. The requirements were stated in a result-oriented manner such that they are testable in the evaluation phase.

The final step in requirement processing was requirement validation [61]. In this step, the requirements were assessed to ensure that they are defined in a standard manner and represent an acceptable description of the identity management system that will be implemented during the development phase. The focus here was to answer the question “Have I got the requirements right?” The validation process involved getting the requirements reviewed.

8.4. Final requirements

The final set of requirements are

- RQ1: Identity information should be portable across security domains to enable single-sing-on.
- RQ2: Identity management system must store only the minimum required personal information of the user that is necessary for authentication & authorization purpose.
- RQ3: Identity management system must define identity information that is specific to the context of the organization. They must not store identity information that is valid outside the context of the organization.
- RQ4: Identity provider must share identity information only with relying parties with which they have established trust and are part of the federation.
- RQ5: Identity provider must share only the minimum identity information required by the relying parties to perform authorization.
- RQ6: Identity provider must assign unique identifiers applicable per user and per relying party so that multiple relying parties cannot collaborate to link user information based on the same identifier of the user across multiple relying parties.
- RQ7: The unique identifiers assigned by the identity provider
    - Must be globally unique
    - Must not reveal any information about the user.
    - Must be completely random and not be derived from other user attributes
- RQ8: Identity management system must maintain a separate role for administration and user role. Each role must have only the minimum privileges to carry out their responsibilities.

- RQ9: Identity management system must be interoperable with different relying parties employing different technologies.
- RQ10: Identity management system must work with different users employing different computing devices to connect to services of different relying parties.
- RQ11: The communication between the end user, relying party and identity provider must be simple, predictable and unambiguous to avoid identity-related attacks.
- RQ12: Identity provider must use a standard way to communicate the metadata information about the user attributes within the federation that is understood implicitly by all the members.
- RQ13: Identity management system must offer consistent user experience across different contexts and multiple relying parties and technologies.
- RQ14: All communication related to identity exchange between the relying party and identity provider must be encrypted using one of the standard encryption algorithm ton ensure confidentiality.
- RQ15: All communication related to identity exchange between the relying party and identity provider must be digitally signed using one of the standard digital signing algorithms to ensure integrity and non-repudiation.
- RQ16: Enterprise identity provider must be able to audit and monitor the activities of the user and resource usage at the service provider. All user identities must be unique, unambiguous, traceable and non-repudiable.
- RQ17: Identity provider must assign assurance level to each attribute shared with the relying party. The assurance level of these attributes must be updated periodically.
- RQ18: End user must always have the option to allow/deny the relying party from receiving identity information.
- RQ19: End user must always be in control of what information is shared with which relying party.
- RD20: Identity provider must provide the user with the option to log out globally from all the live sessions with one or more relying party in the federation.
- RQ21: The relying party must provide the user with the option to log out explicitly from their services.
- RQ22: The logout option must be simple, interactive and the provide feedback to the user on the status of the session.
- RQ23: The user must be implicitly logged out when the user closes the browser.
- RQ24: Identity assertions issued to the relying party must expire as soon as the user logs out of the relying party system or ceases to use the resources offered by the relying party.
- RQ25: There must be a simple and reliable process to disassociate a relying party from the federation.
- RQ26: There must be a simple and reliable process to disassociate a user from the identity provider.
- RQ27: All access to resources/services at the relying party must be denied as soon as the user credentials are revoked by the identity provider.

Some non-functional requirements [62] that are not directly related to the core functionalities of the system but are concerned with the overall quality attributes of the federation identity management system are-

- NFRQ1: Usability – the system should be user-friendly providing a consistent user experience that is aligned with their mental model of how federation identity management system works.
- NFRQ2: Reliability (availability) – since federation identity management systems are mission critical systems. Therefore they should be available all the time, and the failure rate must be minimal
- NFRQ3: Efficiency – the user must be able to authenticate to relying parties within an acceptable response time, and the resource required to process such requests should be reasonable.
- NFRQ4: Capacity – the process of authentication at RPs must be scalable so that many users can simultaneously authenticate to the external services
- NFRQ5: Development and maintenance feasibility – the development and maintenance effort should be cost effective, must use existing technologies and standards, and must require minimum changes to existing systems.

8.5. Summary

This chapter explained the entire process of requirement gathering and validation. In the end, it presented the final set of requirements that must be fulfilled by the designed artifact. High-level requirements were collected from the selected research articles. These high-level requirements were broken down into simple and concise requirements which were then analyzed for their necessity, consistency, completeness, feasibility, ambiguity, redundancy and traceability. After analysis, the requirements were further validated to ensure that they represent a complete set of acceptable requirements for an enterprise level across domain federated identity management. At the end of the chapter, the final set of requirements was presented.

9. Design Implementation Framework for OIDC

Based on the requirements collected in the previous chapter, the implementation framework was designed and developed. This chapter describes the development process and the main iteration phases that shaped and refined the implementation framework. The final version of the OIDC implementation framework has been presented in the last section of this chapter.

9.1. Choice of Method

The aim of the design phase was to design an implementation framework that would provide the roadmap to implement single sign-on protocol securely by considering all the major requirements and the key decision points that must be considered. As there was no such existing implementation framework and the framework was being built from scratch, the design process required innovative thinking and improving iteratively by following a trial and error approach.

The implementation framework has been designed by following a repetitive process of refining the initial basic framework based on the design rules and practical knowledge that emerged from the implementation of OpenId Connect protocol between an identity provider server and a client application (relying party). The method can be described in the following steps:

1. An initial framework designed on the basis of the requirements gathered in the requirement phase.
2. The framework was then put to the test to achieve the single-sign-on and single sign out between the collaborating entities by closely following the framework.
3. The result of implementation was analyzed against the requirements set to understand the limitations or drawbacks of the framework
4. The framework was updated based on the analysis at step 3
5. Steps two, three and four were repeated until a comprehensive and robust implementation framework is achieved

9.2. Development Process

The development of the framework was an iterative process, and there were three major iterations followed a few smaller iterations to fine tune the framework. The framework was developed on a Windows 10 platform using Microsoft Visual Studio 2017 Community Edition[63] as the development environment. Web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Opera were also used during the development.

The framework was tested at the end of each iteration by a simple instantiation of the OIDC protocol by following the implementation framework guidelines; the requirements set in section 8.4 was taken as a checklist, and each requirement was verified against the instantiation. The requirements that were not fulfilled were noted down as a list and then analyzed to form the objectives to improve the framework in the subsequent iteration.

A timeline of the development phase has been presented below followed by the details of each iteration.

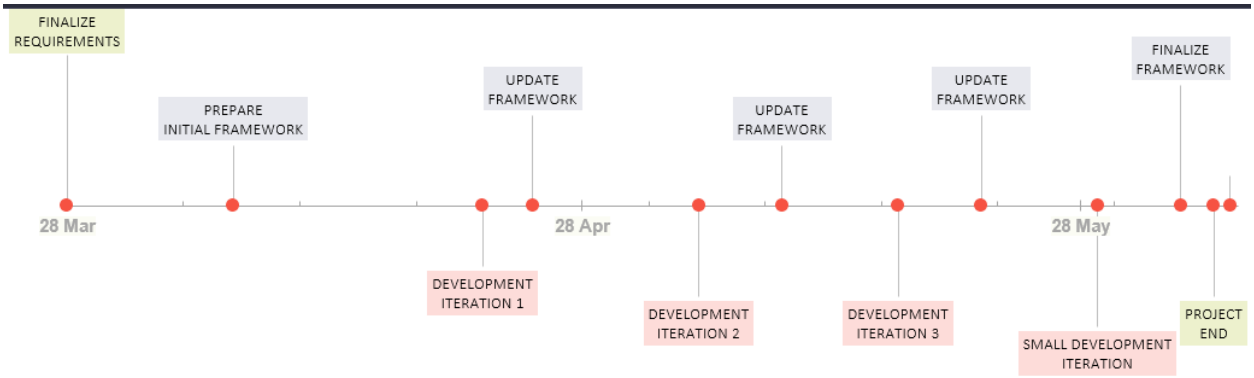


Figure 2: Timeline of the development phase

9.3. Development Iteration 1

During the development, a few of the research articles that focused on the case study related to the practical implementation of SSO protocol were referred. A case study[14] on SSO implementation at a private Swiss bank highlighted the important lessons learnt during the implementation. Another case study[15] that investigated the implementation errors of major Oauth identity providers gave useful recommendations to reduce network attack surface of IdP and RPs. Similarly a research[18] on SSO vulnerabilities due to insecure communication channel between the IdP and RP provided additional inputs for securing the communication channel. Another useful article[20] on automated testing of SSO implementation for vulnerabilities brought into attention the few careless mistakes that developers commit during implementation that might make the whole SSO implementation vulnerable and leak sensitive information. These articles complemented the requirements gathered in the previous phase to help in designing a robust implementation framework.

The first iteration was difficult and time-consuming. There was a learning curve on understanding the OIDC protocol and its nuances; the implementation framework was raw and untested. The result was a working but ad-hoc implementation of the single-sign-on protocol.

It was found that many of the requirements were not fulfilled due to the limitations of the framework. The framework was too simple and missed out on many key decision points that were necessary –

- It did not consider the requirement to have a strong digital certificate necessary for encryption and signing of access tokens;
- It did not consider the requirement to display consent page to the users before sharing the user data; it did not consider the requirement to make all communication between the servers over SSL encryption to avoid network spoofing attacks;
- It also missed the requirement to protect the cookies from being read by the client-side scripts such as Java scripts.
- It did not consider the requirement for a consistent user interface

9.4. Development Iteration 2

The second iteration involved updating the implementation framework and then using it again to implement the OIDC protocol. The result was compared against the requirements set in the same manner

as in the first iteration; the comparison revealed that many requirements that were missed in the first version of the implementation framework were covered; the result was encouraging, and the process revealed a few more things that must be considered during implementation.

It was found that the process of logging out was more complicated than logging in. According to the requirements, there are three options available to the end-user to log out –

A. The user can sign-out from the identity provider which in turn should ideally sign-out the user from all the logged in applications;
B. The user can choose the option of global sign-out on one of the client application which should log-out the user from the identity provider and other client applications;
C. The user can also choose to log-out only from one specific application and expect to remain logged into other applications.

Option A requires that the identity provider keep a track of all the logged in applications and it must have a way to inform each client to log out the user from their respective application once the user has logged out of the identity server. This approach would require each client to collaborate to provide such functionality within their application. Option B is easier to implement as identity server provides means to initiate log out from the client applications but again these clients must implement such a functionality within their application. Option C is practically redundant as the user when tries to visit the application again after logging out from that specific application, the application would automatically fetch the user's authentication and authorization information from the identity server because the user is already logged into the identity server, so the user will never be prompted to log in.

Another major issue that was revealed in this iteration was the difficulty to control the user information shared with the relying party. A relying party application needs access to personal information such as user's first name & last name, email address and similar attributes to offer personalized service to the user. Moreover, the identity provider has no control over the user information that has been shared with the relying party. Therefore, a strong contract is required between the identity provider and the relying party on the usage of user information.

9.5. Development Iteration 3 and further

Iteration 3 started after the framework had been updated from the feedback in iteration 2. Single logout functionality was successfully implemented. The requirements set for section 8.4 was compared against the implementation in the same manner. Most of the requirements from requirement phase were covered in the framework. There were a few missing aspects of the implementation that were the focus of this iteration.

The relation between the identity server and the user data repository was scrutinized with the help of the security principle of "separation of privileges." It was realized that it is very important to separate administrative role that manages user data at the user repository from the role that reads user data from the identity server for authentication purpose. It was concluded that the identity server should only have read rights on the user data. Moreover, identity server must fetch data only for active users.

Disassociation of relying party from identity server was also focused upon in this iteration. Many scenarios related to disassociation were tried e.g. users trying to authenticate at the relying party after

disassociation; handling authentication requests received from the disassociated party; active user session at the relying party at the time of disassociation. Based on these experiments, the framework was updated on how to handle disassociation of a relying party.

Another aspect that needed attention was the user attributes that were forwarded to the relying parties. As per the requirements, the attributes must carry information about its metadata – such as its purpose or description, data length, and assurance level. This information must come from the user data repository. Moreover, there is no out of the box support for specifying the metadata of the attributes and its corresponding assurance level. The only way possible was to design customized attributes with these properties about metadata and assurance level. Nonetheless, the implementation framework was updated to provide metadata and assurance level information about the attributes from the identity provider to the relying party.

One more aspect that came out during the development of implementation was the issue of logging events during the process of single-sign-on and single-sign-out. It was not part of the requirements set, but during the development, it was revealed that logging essential to maintain records of the events for future analysis and troubleshooting errors.

The framework was updated again from the findings of iteration 3. The framework had now covered almost all the important aspects of the implementation.

There were a few more small iterations to fine tune the framework, update the security guidelines and consolidate the implementation rules in one coherent framework. The final framework has been presented in section 9.7

9.6. OIDC Implementation Framework

The final version of OIDC implementation framework has been divided into nine phases. Each phase has been designed to encapsulate one key aspect of the FIM lifecycle. The phases have been arranged sequentially so that each phase lays the groundwork for the next phase. Nevertheless, the phases are not tightly coupled and allow for flexibility depending on the business requirements.

The framework has been presented in two parts – A and B. Part A gives an overview of the entire framework by succinctly defining each phase along with their salient features. Part B describes each phase in a bit more detail and list downs the implementation tasks that must be accomplished within that phase.

Implementation Framework Part A

1. Initial Preparation	•identity user data repository, methods and roles required to fetch the data from the repository and assigning unique identifier to each user
2. Negotiate user data requirements	•negotiate and identify the least amount of mandatory and optional user attributes required by the relying party
3. User attribute Management	•prepare the user attributes for sharing by assigning standard names to attributes, specify a way to communicate metadata and assurance level information of each attribute
4. Establish trust relationship	•verify the relying party's credibility, exchange cryptgraphy related digital certificates and secret keys and assigning a unique identity to the relying party
5. Establish clarity on the protocol	•ensure that all stakeholders understand and agree on the specific message flow, message format, probable use-case scenarios, error handling and exceptions
6. Design consistent user interface	•specifies the key features that the user interface for signing-in, giving consent to data sharing and signing-out must have
7. Implement network security measures	•specifies security measures that must be implemented to protect sensitive data sent over the communication network
8. Implement log out functionality	•specifies the steps required to correctly implement single log out such that user log out behavior is consistent across all the members of the federation
9. Disassociation of relying party	•lists down the steps required to properly disassociate a relying party from the federation.

Implementation Framework Part B

Phase 1 Initial Preparation
This phase involves the basic tasks to identify user data repository, methods and roles required to fetch the data from the repository and prepare the user data for sharing by assigning a unique identifier.
➢ Identify the data repository that stores the user information. ➢ Identify the data repository interface & methods that would be used to fetch user information ➢ Create a custom unique identifier for each user in the context of the specific relying party. The identifier must be globally unique, random and unrelated to any user specific attributes. ➢ Create a role with read-only rights on the user data repository that will be used to fetch user information on behalf of the identity server. The administrative role that manages user data must be completely separate from this role. ➢ Identity server should only fetch active user's information from the data repository. Users that are marked deleted/disabled/revoked must never be allowed to authenticate

Phase 2 Negotiate user data requirements
This phase involves the task of negotiating and identifying the least amount of mandatory and optional user attributes required by the relying party and negotiating user data handling and usage agreement with the relying party.
➢ Identify user data mandatory for authentication ➢ Identify user data mandatory for authorization ➢ Identify user data that is optional but requested by the relying party ➢ Identify additional resources, services or data required by the relying party. For example, any web interface that provides some data or service to reliable partners and the specific relying party is interested in receiving ➢ Negotiate an agreement on how the user information data would be processed, stored and used at the relying party's application server.

Phase 3 User Attribute Management
This phase involves preparing the user attributes that will be shared with the relying party by assigning standard names to attributes, specify a way to communicate metadata and assurance level information of each attribute to the relying party.
➢ Assign standard names for each user attribute that would be shared with relying parties. The user data repository may have internal attribute names or conventional custom name that would make little sense outside the context of the identity provider. ➢ Assign assurance level to each attribute that will be shared. The assurance level must communicate the rigor with which the attribute value was validated by the identity provider

- Recommended providing metadata information on each attribute. The metadata information could include data-type, purpose or description, display name, the expiry date of the information.

Phase 4 Establish trust relationship

This phase involves steps to institute a relationship of trust with the relying party by verifying the relying party's credibility, exchanging digital certificates and secret keys for message encryption and integrity and assigning a unique identity to the relying party.

- Mutually agree on a globally unique and random unique id for the relying party with which the relying party would identify itself to the identity provider server.
- Negotiate the encryption and digital signature algorithm that would be used to protect sensitive data exchange. Choose algorithms that are standard and have proven cryptographic strength. Avoid custom or deprecated algorithms.
- Exchange information about the necessary endpoints/ network addresses that would provide the relying party with user data. Acquire the relying party endpoints form where the user authentication/authorization requests would arrive at the identity server.
- Mutually agree on the period for which the security tokens would remain valid and the procedure to refresh or prolong the token expiry
- Identify the person responsible at both the ends (identity provider & the relying party) to co-ordinate implementation and future management.

Phase 5 Establish clarity on the communication protocol

This phase involves steps to ensure that all stakeholders understand and agree on the specific message flow, message format, common use-case scenarios, error handling and exceptions to avoid confusion and ambiguity in real time.

- Identify the specific communication flow that would be used to achieve the authentication and authorization. A single-sign-on protocol may provide multiple types of flow each suitable for a specific set of scenarios and communication environment
- Design and exchange use-case scenarios and message flow diagram with the relying party to avoid any ambiguity in the understanding of the communication flow
- Implement logging of important events such as successful or failed sign-on or sign-out attempt, attempt to fetch additional user information, expiry, and replacement of access tokens. Log the request source IP address, date & time of the event, input and output parameters involved and any other relevant information.

Phase 6 Design consistent user interface

This phase specifies the key features that the user interface for signing-in, giving consent to data sharing and signing-out must have to ensure that the end user's interaction with the FIM aligns with the user's mental model of authentication and authorization and the users get proper feedback for their actions.
➢ Design & implement a consent screen for users to approve data sharing. Specify the mandatory & optional data requirements of the relying party. ➢ Design and implement the user login interface with the ability to allow users to recover lost/forgotten password. ➢ Design and implement the user log out interface which presents the user the option to log out from all client applications and the identity server. The logout interface must inform the user of the status of log out. ➢ Ensure the user interface is consistent across devices.

Phase 7 Implement network security measures
This phase specifies security measures that must be implemented to protect sensitive user data and security tokens sent over the communication network i.e. internet.
➢ Specify that all communication over the network must be encrypted and digitally signed. ➢ Acquire a private/public key pair certificate from a reputed certificate authority that will be used to provide keys for the cryptographic algorithms ➢ Ensure that all communication happens over HTTP protocol using SSL encryption. ➢ Protect communication between identity server and the relying party from cross site request forgery ➢ Protect authentication cookies from being exposed to any client software on user's device

Phase 8 Implement logout functionality
This phase specifies the steps required to implement single log out correctly and clear user session from all relying parties. This phase ensures that user logout behavior is consistent across all the members of the federation and the user session is properly terminated.
➢ Identity provider must provide a way for the end user to log out both from the identity server and from all the logged in relying parties' application ➢ Ensure that each relying party application provides information about the logged in status of the individual user ➢ Ensure that each relying party application provides an interface to allow identity provider server to trigger log out event on the relying party application ➢ Each relying party must provide the option to log out from the relying party's application and subsequently from the identity server. ➢ At the log out from the identity server, all user sessions and security tokens of that specific user must be revoked immediately.

➢ The relying party application must revoke the access security tokens as soon as the user closes the application

Phase 9 Disassociation of relying party
This phase lists down the steps required to disassociate a relying party from the federation properly. During the lifetime of an FIM, many relying parties would join and subsequently leave the federation as per their changing business environment, and so it is necessary to have a disassociation phase in the implementation framework.
➢ All the security access tokens issued for that relying party must be immediately revoked ➢ Identity provider must trigger log out for all its user from the relying party's application ➢ All configuration related to the relying party must be erased/disabled/revoked. e.g. Relying on party's unique id, shared secret key, public key certificates of the relying party and registered IP address range. ➢ The relying party must remove the option to log-in/ log-out via the identity provider from all its user interface. ➢ The relying party must be asked to erase all user data as per agreement acquired from the identity provider ➢ Identity provider must send out communication to all its user about the disassociation of that specific relying party

9.7. Summary

This chapter presented the details of the entire process of the design and development of the OIDC implementation framework that was the main design artifact of this research. The development process was highly iterative, and each step ended with an evaluation of the artifact against the set of requirements; the evaluation from the previous step set the objectives of the next step. After three major iterations and a few minor iterations, the OIDC implementation framework was ready. The final version of the developed framework has nine phases in sequential order with each phase mentioning the specific tasks that must be achieved to build a robust and secure enterprise level federated identity management.

10. Evaluation of Artifact

The OIDC implementation framework developed and presented in the previous chapter was evaluated by demonstrating its practical utility and subsequently analyzing the implementation framework for its completeness, efficiency, and usability. This chapter describes the evaluation process and the outcome of the evaluation.

10.1. Choice of Method

Evaluation of the designed artifact is a necessary step to demonstrate its quality and practical utility. The pre-requisites for the evaluation are the definition of proper metrics on which the artifact would be evaluated and a well-defined evaluation method that is most suitable for the developed artifact and the chosen metrics [52].

For evaluation, Framework for Evaluation in Design Science (FEDS) [64] has been followed as it provides a detailed guideline on how to conduct an evaluation of a design science research project. The FEDS has four steps. The first step is to set the goals of the evaluation; the second step is to decide upon the evaluation strategy; the third step is to establish the properties to evaluate, and the fourth step is to plan and execute the one or more evaluation episodes.

The main evaluation goals are rigor and efficiency. The OIDC implementation framework must be evaluated to establish that the use of the artifact would certainly help in the secure and efficient implementation of the OIDC protocol. The evaluation process must be efficient and must be conducted within the constraints of the time and resources available. The summative evaluation would provide a high degree of rigor and reliability, while artificial evaluation method would be most cost effective and efficient [64].

Technical Risk & Efficacy strategy is the most appropriate evaluation strategy as the designed artifact is technically oriented, and the evaluation goal is to establish the utility of the artifact. The Technical Risk & Efficacy strategy involves multiple artificial formative evaluations during the development of the artifact and concludes with one or more artificial summative evaluation [64].

The designed artifact is an implementation framework that guides the complex process of real implementation of FIM using OIDC protocol. The evaluation, therefore, requires measuring the quality of the implementation framework. The quality attributes which are most suitable to analyze this artifact are usability, efficiency, and completeness. Usability attribute measures the ease of use of the artifact, it highlights the difficulties in understanding or applying the artifact in a practical scenario by the end users[62]. Efficiency attribute measures how the artifact helps to accomplish the task optimally without the loss of too much time and resources [62]. Completeness attribute measures the extent to which the artifact meets the broad set of requirements determined in the requirement phase [62]. The other attributes that were also included for evaluation were reliability, testability, and comprehensibility. A few other quality attributes that were also considered but not included were consistency, accuracy, and performance. Measuring these attributes would certainly add rigor to the evaluation but measuring them requires multiple implementations of the federated identity management using the implementation framework in different scenarios which was not possible within the constraints of time and resources.

There were multiple evaluation episodes for the artifact. Formative artificial evaluations [64] have already been conducted during the development of the artifact. The artifact development was iterative where each iteration ended with an evaluation to find actions for improvement in the next iteration. Following the Technical Risk & Efficacy strategy, multiple summative artificial evaluations [64] were done after the artifact development was complete. A laboratory experiment was set up to use the artifact to implement FIM with SSO among one IdP and two SP and demonstrate its practical utility; informed arguments were used after that to reason about the usability, efficiency, and completeness of the artifact. A couple of criteria-based quality analysis in the form of the expert reviews were also done with the help of software professionals who had previous experience in implementation of the FIM. As the framework involved both managerial and technical aspects, one review was conducted from the managerial perspective by an experienced product manager while the other review was conducted from the technical perspective by an information system architect.

The other evaluation method that was considered but not used was to conduct a case study to use the artifact in a real business environment to implement FIM with SSO in one or more projects to get a naturalistic summative evaluation [64]. This method was not feasible as they required much time, resources, and collaboration with organizations which were not available during this research.

10.2. Evaluation Episode with Artificial Implementation

Use Case Scenario

A fictitious organization name Able Consultancy (A) provides IT consultants to its various customers. Able Consultancy has made an agreement with a company called Learn Online (E) that provides online education on various IT related subjects, to allow its employees to have unlimited access to all the training modules at Learn Online website. Able Consultancy has also made a similar agreement for its employees with another company called Books Online (B) that provides e-books in IT domain. Able Consultancy wants to establish an identity management federation with the two service provider so that its employees can access the services at Learn online and Books Online using the same user credentials as provided by Able Consultancy. Able consultancy will act as an identity provider and maintain complete control over the user information, access rights, and user authentication; while the organizations – Learn Online and Books Online will focus on providing the service and do not have to implement user management modules at their end.

Implementation Groundwork

The implementation framework developed in the development phase would be used to implement a federated identity management among the three collaborating entities using the OIDC protocol. The main focus would be the establishment of the federation, exchange of identity information post user authentication, single-sign-on and single-sign-out.

The software used for the OIDC implementation are Microsoft Asp.Net core framework[65] to implement the client application websites – Learn Online and Books Online; IdentityServer4[66] to implement the OIDC identity provider; Microsoft Visual Studio 2017 Community Edition[63] was the integrated development environment within which all these applications were developed and tested. A third party software library SeriLog[67] was used for logging. All of this software was available to download for free for education and non-commercial purpose.

The website for Learn Online was built first. It was a simple web application that provided the users with a list of online courses to choose from; users could then register for the courses they wanted to take. The user could visit his/her profile page to see the registered courses. The website for Book Online was similarly built. The user could visit the site, browse the list of books and then order books online. The user could visit his/her profile page to see the list of ordered books.

The identity server of the Able Consultancy was built next. The identity server provides the functionality for the users to register themselves and provide basic details. All the user details are persisted in the Microsoft SQL Server Express[68]. At this stage, there was no trace of OIDC in any of the three applications, and all the applications were independent.

OIDC Implementation

The first phase in the implementation framework was 'Initial Preparation.' The data repository for user information was the Microsoft SQL Server, and Entity Framework [69] was used as the object-relation mapping framework to fetch user data from the SQL Server. The deleted user accounts were removed from the database to prevent deleted users from gaining access.

The second phase was to identify user data requirements of the relying parties i.e. the service providers – Learn Online and Books Online. Only user id and email id were identified as required; the other information about the user such as first name, last name, address and phone number were identified as optional. It was decided that only user id and email will be stored at the relying party's end to help them uniquely identify the users and communicate with them; the other user information will only be fetched from the identity provider when required and only if the user has given consent to it.

The third phase was about user attribute management. The user attributes were assigned standard names as per OIDC protocol, so it was understood within the federation. OIDC does not have specific support for defining the assurance level on each attribute; however, the specification mentions about attributes such as "email_verified" and "phone_verified" which indicates if the identity provider validated the email and phone number respectively at the time of registration. The only way available to provide assurance information was to create a custom attribute for each user attribute to denote the assurance level of that attribute. Similarly, OIDC provides no specific way to convey the metadata properties of the user attributes; every attribute is sent across as a string in JSON object, and it is up to the relying party to interpret the data accordingly.

The fourth phase was to establish a trust relationship with the identity provider and relying parties. Following the framework guidelines, a globally unique identity was assigned to each relying party; a unique and secure private key was randomly generated and shared with each relying party. Relying party's website URLs where the user would be redirected after successful login and log out were collected. The identity provider's URL from where the relying party can fetch the OIDC discovery document was shared with the relying parties. The OIDC communication flow most suitable for user interactive login on server side application is the hybrid flow, and this flow was agreed upon between the relying party and the identity provider. Logging of all operation flow and important events was implemented at the identity provider to maintain the record of all the transactions with the relying parties. Third party software library SeriLog was used for logging, and another software tool Seq[70] was used to read and filter information from the log data.

The fifth phase was to design a consistent and understandable user interface at the identity provider to allow users to sign in, sign out and give consent to information sharing. A consent page was implemented that informed user about the mandatory and optional data required by the relying party. The logout page provided feedback to the user on successful logout and presented them the link to go to the home page of the identity provider. The implementation guideline helped in ensuring that the user interaction was easy to use and intuitive.

The sixth phase was focused on network security aspects of the user identity information exchange. For the purpose of demonstration, a self-signed certificate was created and assigned to the identity provider. This certificate was used to sign all the identity information being sent from the identity provider to relying party. The digital signing algorithm used was RSA with SHA256. All the communication between the identity provider and relying party was using HTTP and SSL protocol. SSL protocol was set as mandatory within the federation, and therefore all communication was end-to-end encrypted. The authentication cookies at the relying party were set to 'HTTP Only' to forbid any application on the user's computing device from reading the cookie. Measures were also taken to avoid cross-site request forgery by attaching validation token with each response.

The seventh phase was about implementing the single-log-out functionality both at the identity provider and the relying party. A global logout button was implemented at each relying party. This button allowed the user to log out from the identity server and also from each of the logged in relying party's application. Similarly, a global logout button was present at the identity server application to allow the users to log out from all the sessions at all the relying party application and also from the identity server. As soon as the user logged out, all authentication cookies and tokens were destroyed so that no further access was possible.

The last phase was the disassociation of the relying party from the identity provider. The relying party was disassociated by removing all the configuration of the relying party at the identity provider. This ensured that the relying party could no longer request authentication information or refresh current tokens. However, already logged in users could continue to use the relying party's service. At the relying party's end, all configuration of the identity provider was removed so that users can no longer authenticate using the identity provider credentials. Moreover, all user authentication cookies were manually destroyed to force log out all the users at the relying party. As per the initial agreement, no user information was stored at the relying party except the user id, so no user data delete was required at the relying party. The implementation framework guidelines helped to disassociate the relation between the identity provider and the relying party in a systematic manner.

Once the implementation was complete, functional testing was done to check if the entire implementation worked correctly. A few minor bugs were found due to errors in writing the code, but, overall the test cases gave a positive result; all the essential features of the federated identity management were working as expected.

A detailed description of the service provider's website and identity server along with the screenshots of important web pages have been attached in Appendix A.

Communication Flow Diagrams

As per the guidelines in phase five of the implementation framework to establish clarity of communication flow among the relying parties and the identity provider, two activity sequence diagrams were created.

Single Sign On

The following activity diagrams provide the sequence of activities for a successful single sign-on of the user.

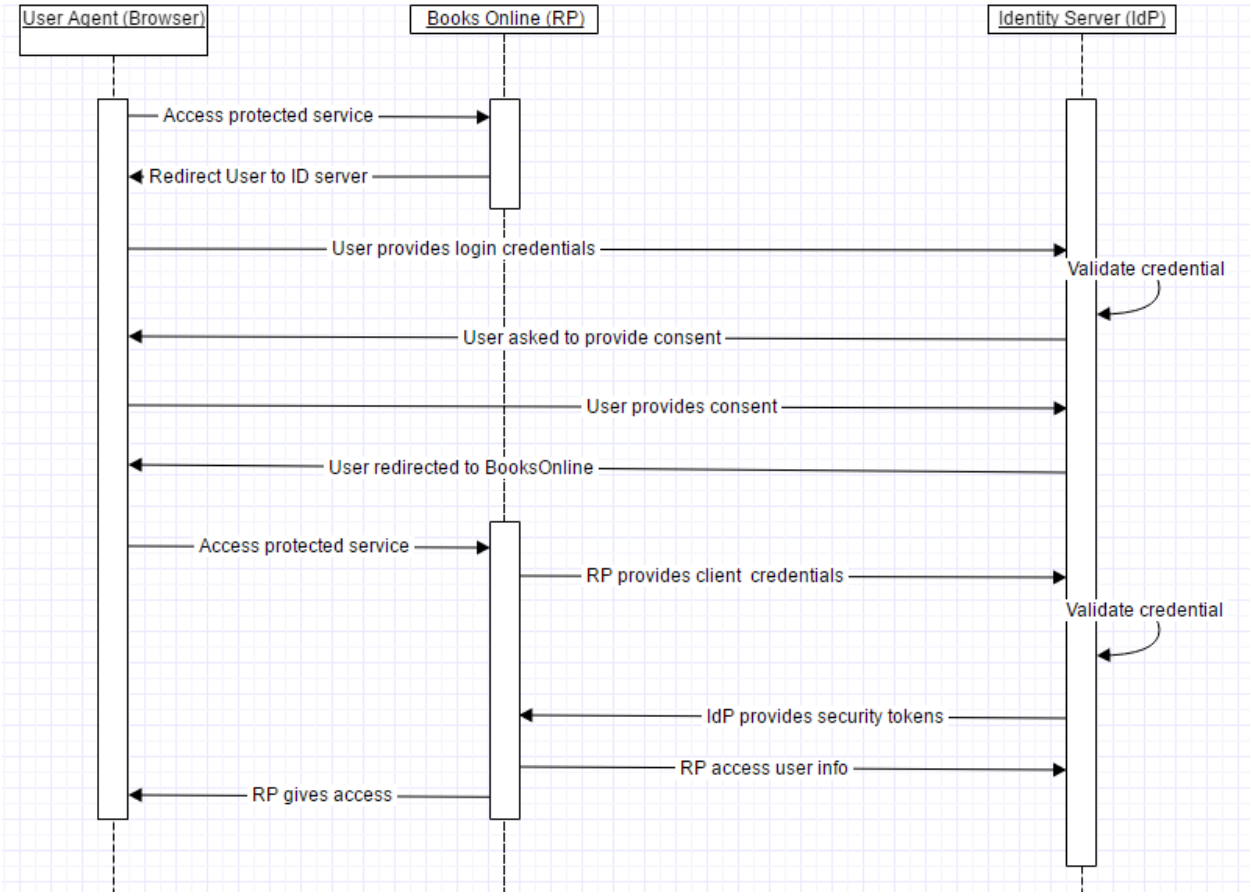


Figure 3: Single Sign-On with One RP

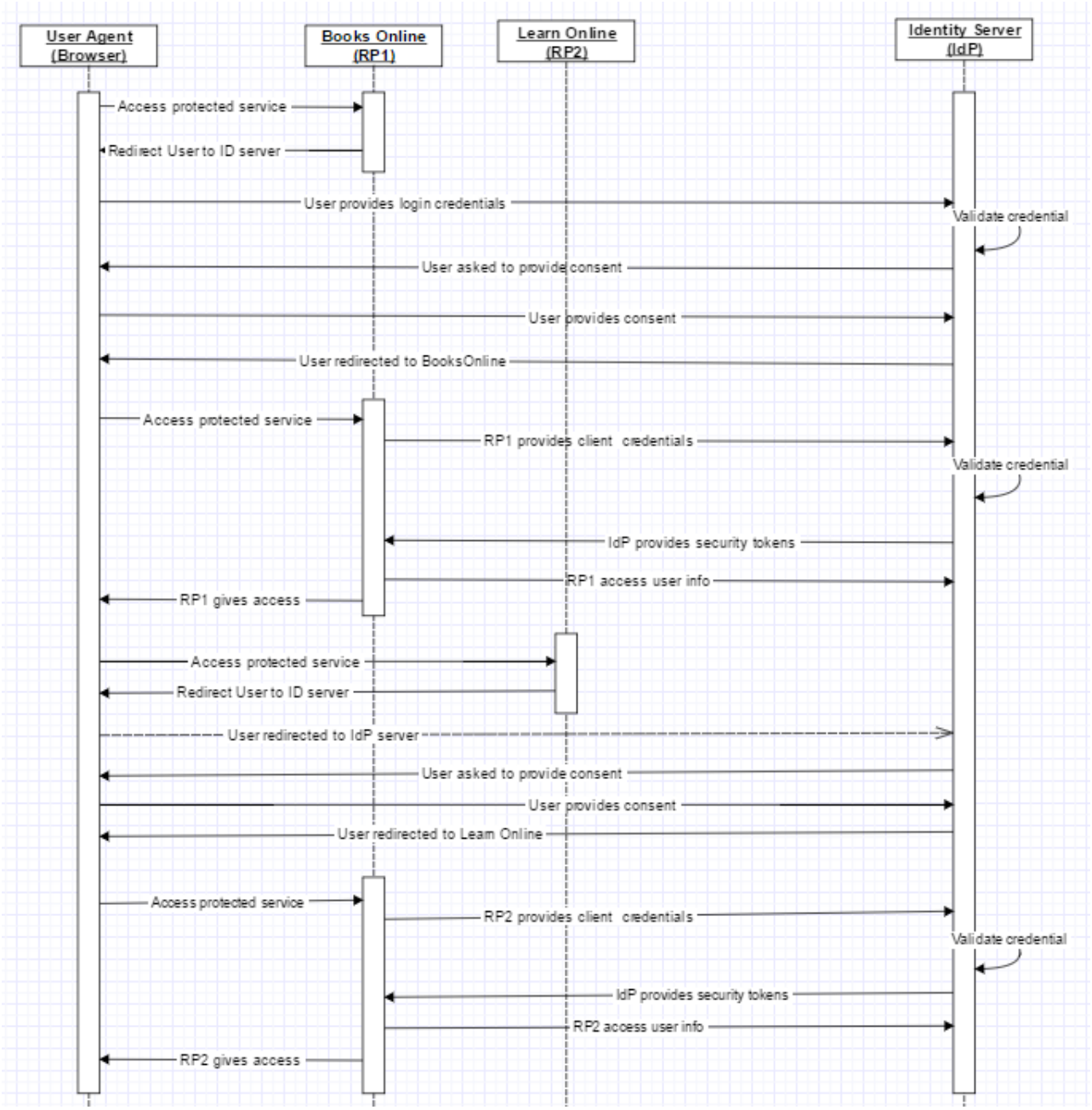


Figure 4: Single Sign-On with Two RPs

Single Sign Out

The following activity diagrams provide the sequence of activities for a successful single sign out of the user.

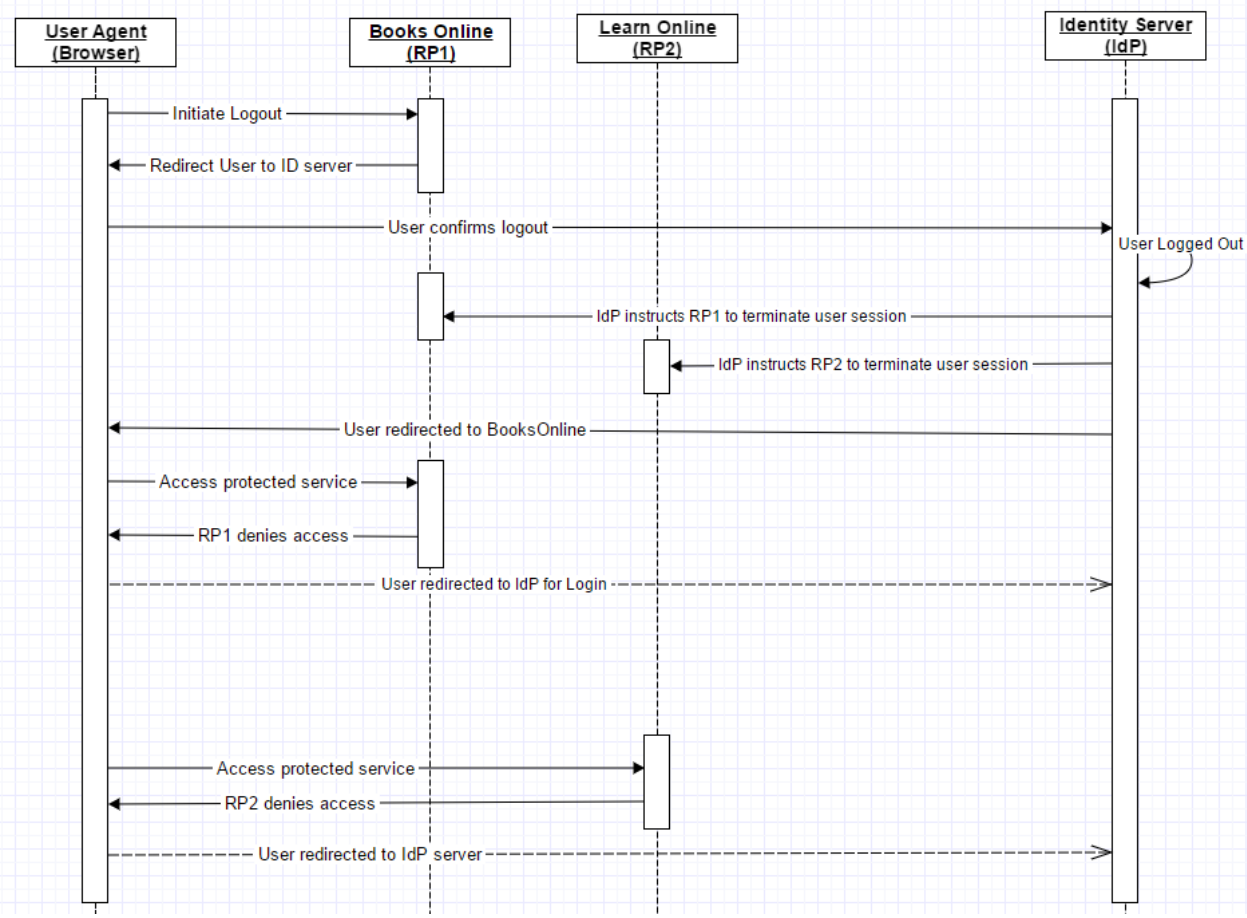


Figure 5: Single Sign Out

Appraisal of the Implementation Framework

The implementation framework has covered all the requirements set at the end of the requirement phase either implicitly or explicitly. REQ1, REQ9, and REQ10 are implicitly enabled by the OIDC protocol. REQ2, REQ3, and REQ5 are covered in phase 2 of the framework which proposes to negotiate user data requirements with the relying party so that the only minimum, required and context specific data is shared. REQ4 has been covered in phase 4 of the framework that provides guidelines on how to establish trust with the relying party. REQ6, REQ7 and REQ16 related to globally unique and completely random yet traceable unique user id has been covered in the first phase of the framework that instructs on how to create the user's unique id to shares across with the relying party. REQ8 is covered by the guideline for separate administrative roles in the framework's first phase. REQ11 is covered in the fifth phase of the framework that provides clear instructions on how to establish clarity of protocol message flow and use case scenarios among the collaborating entities. REQ12 and REQ17 are covered in the third phase of the framework that provides guidelines on how to deal with user attribute exchange in a standardized way along with the metadata and assurance level information associated with the attribute. REQ13 is covered in the sixth phase which is all about designing a consistent user interface irrespective of the relying party's application and user's computing device. REQ14 and REQ15 that are related to strong encryption and integrity of messages exchanged are covered in phase 4 and 6; in phase 4 the identity provider negotiates the encryption and integrity providing cryptographic algorithms and phase 6 describes how network

security should be implemented by applying those algorithms. REQ18 and REQ19 related to user consent and control over information shared are covered in phase 6 which instructs that user must be provided with the opportunity to see and decide upon what information to share with the help of the consent screen; OIDC protocol also supports such a feature implicitly. REQ20, REQ21, REQ22, REQ23 and REQ24 related to global log out either from the identity provider or one of the relying party's application is covered in the eighth phase of the framework which is all about single log out implementation. REQ22 is also covered in the sixth phase that deals with the implementation of the consistent and interactive user interface. The last phase of the framework i.e. The ninth phase provides detail implementation guidelines on how to disassociate a relying party from the federation. REQ26 on revoking a user credential from the identity provider is a typical administrative task for identity provider management and is independent of the protocol used for federation. REQ27 which is about denying access to deleted user is partially covered in the first phase where one of implementation guideline is to allow user authentication only for active users. REQ27 is also fulfilled by the eighth phase that clearly states that security tokens should be immediately revoked when a user is logged out; it is assumed that the user is logged out before it is deleted from the identity provider. As all the requirements can be clearly mapped to the implementation framework, it is evident that the framework satisfies the evaluation criterion of "completeness." A simple mapping between the phases in the framework and the requirements they cover is shown in the following table.

Phase No.	Phase Name	Requirements Covered
1	Initial Preparation	REQ6, REQ7, REQ8, REQ16
2	Negotiate user data requirements of the relying party	REQ2, REQ3, REQ5, REQ26
3	User attribute Management	REQ12, REQ17
4	Establish trust relationship with the relying party	REQ4, REQ14, REQ15
5	Establish clarity on the protocol, communication flow, and specific message exchanges	REQ11
6	Design consistent user interface	REQ13, REQ18, REQ19, REQ20, REQ22
7	Implement security measures to protect the data exchanged over network	REQ14, REQ15
8	Implement logout functionality	REQ20, REQ21, REQ22, REQ23, REQ24, and REQ27
9	Disassociation of relying party from identity provider	REQ25

Implementation of a federated identity management can easily become an overwhelming task due to the many aspects of the federation related to collaborating parties, technical requirements, security requirements and ease of user interaction. Each phase of the framework focusses on a specific aspect related to federated identity management; each phase covers the most basic and essential requirements in the form of implementation guidelines that must be fulfilled for that aspect to be implemented

correctly. By following the framework, the entire project of establishing the federation is streamlined and divided into phases and then into small, unambiguous and manageable tasks. These tasks can then be planned and assigned to individual members or sub-groups within the implementation team. Since these tasks are clear to-do objectives, they can be easily monitored, and the result can be validated against the task goals to ascertain if the task has been accomplished. The phases within the framework are not strictly sequential, but they can also be carried out in parallel depending on the implementation team size, thus further improving the efficiency. Therefore, it is evident that the framework would certainly improve the efficiency of implementing a federated identity management. However, the improved efficiency has to be evaluated further by using the framework for real implementations and with teams of varying sizes.

The framework offers straight-forward and practical implementation guidelines that are easy to understand and follow. The framework has been sub-divided into phases that can be followed sequentially or can be done in parallel depending on the implementation team size; therefore, a team of any size can use this framework. The implementation framework can be used both by software engineers to do the technical implementation by deriving low-level technical requirements from the framework, or it can also be used by the project managers as a checklist to track the progress of implementation. The framework is not only useful for new implementations but can also be used to evaluate the existing implementations and identify parts that need more attention or have missing functionalities. The framework phases are relatively independent of one another, and therefore these phases can be used individually as a guideline to enhance specific parts of existing implementations. It can be concluded that the implementation framework is usable in multiple scenarios to serve different purposes. Last but not the least, the framework has been carefully designed to encapsulate all the latest requirements gathered from the recent peer-reviewed journal papers; therefore, it can be said that the framework is relevant and applicable to the practical scenarios of recent times. Nevertheless, the usability aspect also needs further evaluation by using it for real project implementation and gathering feedback from different implementation engineers or managers to get the user's perspective on the framework's usability.

10.3. Evaluation Episode with Expert Review

Two external evaluators were selected to evaluate the framework based on the six quality criteria – Usability, Efficiency, Completeness, Reliability, Testability, and Comprehensibility. One of the experts evaluated the framework from the project management perspective while the other expert evaluated from the technical perspective. A feedback form was prepared to enable the evaluators to give their feedback in a structured format. The feedback form had one row each for the six quality attributes. Each row had a column to rate the framework on the corresponding quality criteria on the scale of 1 to 5 (5-Excellent, 4-Very Good, 3-Good, 2-Average, and 1-Poor) with 5 being the best score. There was an adjacent column to write comments or suggestions from the evaluator.

The evaluators were selected based on their overall experience in the IT domain and their practical experience with FIM implementation. There were a few prospective evaluators who were easily accessible during the research. These prospective evaluators were contacted through email or Skype or phone. Two of the most experienced IT professionals who showed a willingness to participate in the research work were chosen as evaluators.

The evaluator selected for the review from the management perspective had an overall IT domain experience of 15 years, and in her role as a product manager, she had managed a few implementation

projects for integration of service providers into the FIM of the organization. Therefore, she had a good understanding of the practical issues and challenges that must be tackled during an FIM implementation. The other evaluator who was selected for the technical evaluation had an overall IT domain experience of 9 years, and in his role as a system architect, he had actively worked on a couple of FIM implementation projects. Therefore, he had the practical experience of handling the technical challenges of an FIM implementation. Both of these evaluators, therefore, had the necessary experiences and skills to evaluate the implementation framework.

The evaluators were contacted through Skype to request for a short meeting in person to discuss the research work and to convince them to participate as an evaluator. During the meeting, the research objective and how the implementation framework had been developed was explained. The prospective evaluators were also explained the evaluation process, the time frame to do the evaluation and the expectations from the evaluators. The evaluators showed interested in the research work as they had practical experience with FIM implementation and were eager to see how the implementation framework could be used in their future projects. So they agreed to participate in the evaluation.

The implementation framework and the feedback form were sent in an email to the evaluators after they had agreed to participate in the evaluation. The evaluators were also encouraged to write their comments or suggestions for improvement of the framework on each of the quality attributes. The feedback from the evaluators was also received through the email. The filled in feedback forms from the evaluators have been attached in Appendix B.

Management Perspective

The evaluator graded the implementation framework as excellent (score 5) on the quality attributes - Usability, Efficiency, Reliability, Testability, and Comprehensibility. On usability, the evaluator commented that the framework has easy to follow step-by-step scenarios and they would most likely use this framework for their upcoming FIM/SSO project. The completeness attribute was graded as very- good (score 4) and the evaluator suggested that the framework could be extended to include "Testing" and "Maintenance" phase where the testing phase would guide the verification of the whole implementation and the maintenance would guide the long-term maintenance procedure for the identity management server and relying party integration. This suggestion has been accepted and added to future work on the framework extension.

Technical Perspective

The evaluator graded the implementation framework as excellent (score 5) on the quality attributes - Completeness, Testability, and Comprehensibility. On completeness, the evaluator commented that the framework covered all the essential aspects of SSO and that the framework was flexible and broadly applicable in many scenarios. On testability, the evaluator confirmed that the implementation framework guidelines are testable in real scenarios. The other quality attributes - Usability, Efficiency, and Reliability were graded as very good (score 4), and many practical suggestions on improvement were mentioned. The evaluator suggested adding more technical details on the minimum security requirements that must be fulfilled by both the identity provider and the relying party before they can establish a trust relationship. The evaluator further suggested going deeper into the technical nitty-gritty and challenges to be considered when implementing the FIM project.

Expert Evaluation Summary

To summarize, the feedback from both the evaluators were very positive, and both of them have rated the framework highly on all the six quality attributes; none of the attributes were rated below 4 by any of the evaluators. Their comments and suggestions for improvement were insightful. One of the important suggestions was to broaden the scope of the framework by including the testing and maintenance phase while the other important suggestion was to go deeper into the technical details of the implementation. Both of these suggestions were accepted and added to the future work to enhance the framework.

10.4. Summary

This chapter explained the evaluation process of the designed OIDC implementation framework and its outcome. Technical Risk & Efficacy evaluation strategy was adopted from the Framework for Evaluation in Design Science (FEDS). The was one episode of artificial summative evaluation using a laboratory experiment and then two episodes of expert review using criteria-based quality analysis. The laboratory experiment involved using the framework to set up FIM among two service providers and one identity provider and then analyzing the framework for its completeness, efficiency, and usability. The expert review analyzed the framework based on six chosen quality criteria - Usability, Efficiency, Completeness, Reliability, Testability, and Comprehensibility. The overall feedback from these evaluations was positive with a few suggestions for further improvements.

11. Discussion

The final chapter concludes the research work by presenting the final comments on the result, research contribution, limitations of this research and the probable future work.

11.1. Results

Based on the demonstration and evaluation, it is clear that the implementation framework covered all the major requirements for the establishment of the Federated identity management lifecycle process. In addition to the basic implementation requirements, the framework also focused on the security aspects as well as the end user usability aspects of the federated identity management. Last but not the least important is the fact that the framework could be used efficiently to do a practical implementation of the federated identity management using OIDC protocol among the collaborating identity provider and the relying parties. The output of this research is a well-defined and usable implementation framework that successfully achieved the objective set at the beginning of this research.

11.2. Research Contribution

As OIDC is a fairly recent standard, it will take some time for the enterprises to grow in confidence and accept OIDC as an alternative to SAML[5]. However, it is expected that the burden of legacy features in SAML and its inability to work with mobile devices and mobile applications will accelerate the adoption of OIDC at the enterprise level. OIDC has the necessary features to fulfill the major requirements of federated identity management at the enterprise level and as it evolves many new capabilities will be added to strengthen the protocol further. However, as the experience with SAML has shown that the protocol documents tend to become very large and complex over time and it becomes almost impossible to follow the documentation to implement the protocol. Such lengthy documents lead to a faulty and vulnerable implementation where many important requirements are missed.

This research work has led the foundation step to design an implementation framework to aid the software developers to realize a secure federated identity management system that covers all bases. The framework has been designed by collecting the requirements from all the important aspects of an identity management system and extracting a set of implementation guidelines which is then presented as a framework with sequential phases to follow in an unambiguous manner. Although the focus of this research was on OIDC protocol, the resulting framework is generic and covers the most basic requirements which are true for implementations with any other existing or future protocols. This framework can be easily extended either by broadening the protocols that it covers or by going in depth of OIDC protocol and adding more technical information to it.

Therefore, the research contributes to the existing knowledge base on federated identity management with the focus on the correct implementation of the single sign-on protocols.

11.3. Improvement over existing solutions

At present, there are no implementation frameworks available that cover all the areas of the FIM implementation using OIDC protocol. The implementation teams have to rely either on case studies of previous implementations or follow multiple guidelines from scattered sources on different aspects of the

FIM. The other alternative for the implementation team is to adopt an ad-hoc process of implementation which may risk the project spiraling out of control in terms of time and budget and severely affect the efficiency. The implementation framework presented here covers the entire lifecycle of federated identity management - from the initial establishment of the federation to typical use case scenarios such as single sign on, single sign out, user interaction, and finally the relying party disassociation from the federation. The framework can thus guide the FIM implementation at every stage. As the framework offers a broad set of guidelines without going too deep into the technical details or setting any rigid pre-requisites, it is flexible enough to be applied in various scenarios. These aspects of the implementation framework are a major improvement over the existing solutions and therefore fulfills the critical gap of a lack of a complete and flexible implementation guide at least for the OIDC protocol implementation.

11.4. Limitations

The requirements for the implementation framework have been primarily derived from the literature review focusing on the technical aspects of the federated identity management. Other methods such as case studies at organizations using federated identity management and interviews with users who use it on a regular basis could have revealed the user-centric behavior and issues related to federated identity management. These additional methods would have certainly brought out new requirements and contributed to a better design of the framework. Regarding the evaluation, a rigorous analytical evaluation focusing on the architectural properties of the framework would have enhanced the confidence on the framework.

There are certain limitations on the implementation framework developed during this research. The framework has been subjected to limited testing only during development and evaluation. It would require extensive testing at the hands of the professional developers and experts to assess its practical utility and efficiency better before it can be put to widespread use. The framework has not been tested in a real implementation of federated identity management in a production environment. Multiple federated identity management implementations in real scenarios with different types of applications using this framework would have established the reliability and performance of the framework.

Another aspect of the framework that could be argued as its limitation is that the framework does not go deep into the technical details of the OIDC protocol elaborating on the various possible flows and how to deal with the unique attributes of different kinds of participating applications. However, the driving principle behind the design of the framework was to keep it as generic as possible so that framework is flexible and broadly applicable in different scenarios; adding too many technical details would have made it unnecessarily complex and rigid.

11.5. Future Work

The various limitations highlighted in the previous section provide many opportunities for future work to improve and extend the framework and also make it more robust. An immediate future work could be to gather users' perspective and issues they face related to the usage of federated identity management and update the framework to handle those issues. The framework must also be tested in one or more real scenario implementations and then further improve the framework with the insights gained from practical experience.

There are a few aspects related to real-time implementations which have not been considered in the implementation framework such as - testing, deployment of the identity management services in a distributed environment, long -term maintenance of identity servers and relying party integration, scalability and performance issues in very large scale environments with thousands of users spread across the globe. However, as a future work, the framework can be broadened in scope to include these aspects. The other interesting area of research would be to apply the framework on another protocol such as SAML to generalize the framework and broaden its scope.

The framework can also be enhanced to go deeper into the technical details of the various phases of the FIM implementation, or the framework can be specialized to focus on the specific types of interactions such as machine to machine authentication or user authentication through browser based client side applications and mobile applications. The present framework can be used as a foundation to design such extensions.

Another important future work but not directly related to the implementation framework is to investigate the challenges of replacing the SAML-based FIM with OIDC within an enterprise. OIDC is a potential upcoming protocol that could be an alternative to SAML, and there will be business requirements to replace legacy SAML protocol with a more light-weight and future ready protocol.

To conclude, there are many interesting and important research areas related to OIDC implementation framework, and it would be exciting to observe how the implementation framework evolves in the future.

11.6. Summary

This final chapter presented the concluding remarks on the overall research. The research work was able to meet the research objective of designing an implementation framework for using OIDC protocol to establish enterprise level FIM. The implementation framework is new and has not been used in the real environment which is one of the limitations of this research. However, the implementation framework is a significant improvement over the existing ad-hoc approach of establishing FIM at the enterprise level. The implementation framework presented here could be developed further in multiple ways either by going deeper into the technical specifics of the implementation or by getting broader to include areas such as long-term maintenance and deployment.

12. References

[1] S. Mansfield-Devine, “Single sign-on: Matching convenience with security,” *Biometric Technol. Today*, vol. 2011, no. 7, pp. 7–11, 2011.

[2] A. A. Malik, H. Anwar, and M. A. Shibli, “Federated Identity Management (FIM): Challenges and opportunities,” *Proc. - 2015 Conf. Inf. Assur. Cyber Secur. CIACS 2015*, no. 1, pp. 75–82, 2016.

[3] V. Radha and D. H. Reddy, “A Survey on Single Sign-On Techniques,” *Procedia Technol.*, vol. 4, pp. 134–139, 2012.

[4] S. Suoranta, K. Manzoor, A. Tontti, J. Ruuskanen, and T. Aura, “Logout in single sign-on systems: Problems and solutions,” *J. Inf. Secur. Appl.*, vol. 19, no. 1, pp. 61–77, 2014.

[5] N. Naik and P. Jenkins, “An Analysis of Open Standard Identity Protocols in Cloud Computing Security Paradigm,” *2016 IEEE 14th Intl Conf Dependable, Auton. Secur. Comput. 14th Intl Conf Pervasive Intell. Comput. 2nd Intl Conf Big Data Intell. Comput. Cyber Sci. Technol. Congr.*, pp. 428–431, 2016.

[6] B. C. Neuman and T. Ts’o, “Kerberos: An Authentication Sevice for Computer Networks,” *IEEE Commun. Mag.*, vol. 32, no. September, pp. 33–38, 1994.

[7] OASIS, “Security Assertion Markup Language (SAML) V2.0 Technical Overview,” 2008. [Online]. Available: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html. [Accessed: 19-May-2017].

[8] Microsoft, “Understanding WS-Federation,” 2007. [Online]. Available: https://msdn.microsoft.com/en-us/library/bb498017.aspx. [Accessed: 19-May-2017].

[9] OpenID Foundation, “OpenID Authentication 2.0 - Final,” 2007. [Online]. Available: http://openid.net/specs/openid-authentication-2_0.html. [Accessed: 19-May-2017].

[10] B. Leiba, “OAuth web authorization protocol,” *IEEE Internet Comput.*, vol. 16, no. 1, pp. 74–77, 2012.

[11] OpenID Foundation, “Welcome to OpenID Connect,” 2016. [Online]. Available: http://openid.net/connect/. [Accessed: 19-May-2017].

[12] V. BELTRAN, “Characterization of web single sign-on protocols,” *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 24–30, 2016.

[13] P. MADSEN, “OpenID Connect 1.0 for Enterprise,” *White Paper*. Ping Identity Corporation, pp. 36–39.

[14] A. Volchkov, “Revisiting Single Sign-On,” *J. IT Prof.*, vol. 3, no. 1, pp. 39–45, 2001.

[15] S.-T. Sun and K. Beznosov, “The Devil is in the (Implementation) Details: An Empirical Analysis of OAuth SSO Systems,” *Proc. 2012 ACM Conf. Comput. Commun. Secur. - CCS ’12*, pp. 378–390, 2012.

[16] A. Armando, R. Carbone, L. Compagna, J. Cuéllar, G. Pellegrino, and A. Sorniotti, “An authentication flaw in browser-based Single Sign-On protocols: Impact and remediations,” *Comput. Secur.*, vol. 33, pp. 41–58, 2013.

[17] L. Wanpeng and C. J. Mitchell, “Analysing the Security of Google’s Implementation of OpenID Connect,” in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2016, vol. 9721, pp. 357–376.

[18] Y. Cao, Y. Shoshitaishvili, K. Borgolte, C. Kruegel, G. Vigna, and Y. Chen, “Protecting web-based single sign-on protocols against relying party impersonation attacks through a dedicated bi-directional authenticated secure channel,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8688 LNCS, pp. 276–298, 2014.

[19] H. Lee and N. Wang, “The implementation and investigation of securing web applications upon multi-platform for a single sign-on functionality,” *Int. J. Adv. Comput. Res.*, vol. 6, no. 23, pp. 39–46, 2016.

[20] Y. Zhou and D. Evans, “SSOScan: Automated Testing of Web Applications for Single Sign-On Vulnerabilities,” *23rd USENIX Secur. Symp. (USENIX Secur. 14)*, pp. 495–510, 2014.

[21] J. von Brocke, A. Simons, B. Niehaves, K. Riemer, R. Plattfaut, A. Cleven, J. Von Brocke, and K. Reimer, “Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process,” *17th Eur. Conf. Inf. Syst.*, vol. 9, pp. 2206–2217, 2009.

[22] Y. Levy and T. J. Ellis, “A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research,” *Inf. Sci. J.*, vol. 9, pp. 181–212, 2006.

[23] M. Kunz, M. Hummer, L. Fuchs, M. Netter, and G. Pernul, “Analyzing recent trends in enterprise identity management,” *Proc. - Int. Work. Database Expert Syst. Appl. DEXA*, pp. 273–277, 2014.

[24] M. C. Domenech, E. Comunello, and M. S. Wangham, “Identity management in e-Health: A case study of web of things application using OpenID connect,” *2014 IEEE 16th Int. Conf. e-Health Networking, Appl. Serv. Heal. 2014*, no. iii, pp. 219–224, 2015.

[25] J. Jensen, “Federated Identity Management Challenges,” *2012 Seventh Int. Conf. Availability, Reliab. Secur.*, pp. 230–235, 2012.

[26] J. Jensen and M. G. Jaatun, “Federated Identity Management — We Built It ; Why Won ’ t They Come ?,” *IEEE Secur. Priv.*, vol. 11, no. April, pp. 34–41, 2013.

[27] L. Lynch, “Inside the identity management game,” *IEEE Internet Comput.*, vol. 15, no. 5, pp. 78–82, 2011.

[28] T. Saito, Y. Tsunoda, D. Miyata, R. Watanabe, and Y. Chen, “An Authorization Scheme Concealing Client’s Access from Authentication Server,” *2016 10th Int. Conf. Innov. Mob. Internet Serv. Ubiquitous Comput.*, pp. 593–598, 2016.

[29] A. Al??Abdulwahid, N. Clarke, I. Stengel, S. Furnell, and C. Reich, “Continuous and transparent multimodal authentication: reviewing the state of the art,” *Cluster Comput.*, vol. 19, no. 1, pp. 455–474, 2016.

[30] C. Mainka, V. Mladenov, F. Feldmann, J. Krautwald, and J. Schwenk, “Your software at my service : Security analysis of SaaS single sign-on solutions in the cloud,” *Proc. ACM Conf. Comput. Commun. Secur.*, vol. 2014–Novem, no. November, pp. 93–104, 2014.

[31] U. Habiba, R. Masood, M. Shibli, and M. Niazi, “Cloud identity management security issues & solutions: a taxonomy,” *Complex Adapt. Syst. Model.*, vol. 2, no. 1, pp. 1–37, 2014.

[32] R. Horbe and W. Hotzendorfer, “Privacy by design in federated identity management,” *Proc. - 2015 IEEE Secur. Priv. Work. SPW 2015*, pp. 167–174, 2015.

[33] M. Talamo, M. L. Barchiesi, D. Merella, and C. H. Schunck, “Global convergence in digital identity and attribute management: Emerging needs for standardization,” *Proc. 2014 ITU Kaleidosc. Acad. Conf. Living a Converg. World - Impos. Without Stand. K 2014*, pp. 15–21, 2014.

[34] D. W. Chadwick, “Federated Identity Management,” in *Foundations of Security Analysis and Design V: FOSAD 2007/2008/2009 Tutorial Lectures*, A. Aldini, G. Barthe, and R. Gorrieri, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 96–120.

[35] E. Birrell and F. B. Schneider, “Federated Identity Management Systems : A Privacy-Based Characterization,” *IEEE Secur. Priv.*, vol. 11, no. October, pp. 36–48, 2013.

[36] G. Ben Ayed, “Digital Identity Management,” in *Architecting User-Centric Privacy-as-a-Set-of-Services: Digital Identity-Related Privacy Framework*, Cham: Springer International Publishing, 2014, pp. 57–95.

[37] J. Jensen and A. A. Nyre, “Federated identity management and usage control - Obstacles to industry adoption,” *Proc. - 2013 Int. Conf. Availability, Reliab. Secur. ARES 2013*, pp. 31–41, 2013.

[38] MIT, “Kerberos: The Network Authentication Protocol.” [Online]. Available: https://web.mit.edu/kerberos/. [Accessed: 28-May-2017].

[39] S. M. Bellovin and M. Merritt, “Limitations of the Kerberos authentication system,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 20, pp. 119–132, 1990.

[40] Y. Y. Du, H. Y. Ning, P. Yang, and Y. X. Cui, “Improvement of kerberos protocol based on dynamic password and "one-time public key,” *2014 10th Int. Conf. Nat. Comput. ICNC 2014*, pp. 1020–1025, 2014.

[41] A. Armando, R. Carbone, L. Compagna, J. Cuellar, and L. Tobarra, “Formal Analysis of SAML 2.0 Web Browser Single Sign-on: Breaking the SAML-based Single Sign-on for Google Apps,” *Proc. 6th ACM Work. Form. Methods Secur. Eng.*, pp. 1–10, 2008.

[42] P. Sharma and V. K. Sihag, “Hybrid Single Sign-On Protocol for Lightweight Devices,” *2016 IEEE 6th Int. Conf. Adv. Comput.*, pp. 679–684, 2016.

[43] J. Somorovsky and A. Mayer, “On Breaking SAML: Be Whoever You Want to Be.,” *USENIX Secur. …*, p. 16, 2012.

[44] Shibboleth Consortium, “Shibboleth.” [Online]. Available: https://shibboleth.net/. [Accessed: 28-May-2017].

[45] R. L. B. Morgan, S. Cantor, S. Carmody, W. Hoehn, and K. Klingenstein, “Federated Security: The Shibboleth Approach,” *Educ. Q.*, vol. 27, no. 4, pp. 12–17, 2004.

[46] D. Recordon and D. Reed, “OpenID 2.0: A Platform for User-Centric Identity Management,” *Proc. Second ACM Work. Digit. identity Manag.*, pp. 11–16, 2006.

[47] OpenID Foundation, “OpenID Facts.” [Online]. Available: http://openid.net/connect/faq/. [Accessed: 21-Mar-2017].

[48] OpenID Foundation, “Welcome to OpenID.” [Online]. Available: http://openid.net/connect/.

[Accessed: 21-Mar-2017].

[49] A. Blazquez, V. Tsiatsis, and K. Vandikas, "Performance evaluation of OpenID connect for an IoT information marketplace," *IEEE Veh. Technol. Conf.*, vol. 2015, 2015.

[50] OpenID Foundation, "OpenID Connect Core 1.0." [Online]. Available: http://openid.net/specs/openid-connect-core-1_0.html. [Accessed: 21-Mar-2017].

[51] T. Jager and J. Somorovsky, "How to break XML encryption," *Proc. 18th ACM Conf. Comput. Commun. Secur. - CCS '11*, p. 413, 2011.

[52] A. R. Hevner, S. T. March, J. Park, and S. Ram, "DESIGN SCIENCE IN INFORMATION SYSTEMS RESEARCH," *MIS Q.*, vol. 28, no. 1, pp. 75–105, 2004.

[53] M. K. Sein, O. Henfridsson, S. Purao, M. Rossi, and R. I. Lindgren, "Action design research," *MIS Q.*, vol. 30, no. 3, pp. 611–642, 2011.

[54] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, Dec. 2007.

[55] M. Denscombe, *Case Studies*, 5th ed. McGraw-Hill Education, 2014.

[56] M. Denscombe, "Surveys," in *The good research guide: for small-scale social research projects*, 5th ed., McGraw-Hill Education, 2014, pp. 7–31.

[57] M. Denscombe, *Documents*, 5th ed. McGraw-Hill Education, 2014.

[58] G. Kotonya and I. Sommerville, "Requirements Elicitation and Analysis," in *Requirements engineering: processes and techniques*, 1st ed., Wiley Publishing, 1998, pp. 53–85.

[59] K. Cameron, "The Laws of Identity," *Microsoft Corporation*, 2005. [Online]. Available: https://msdn.microsoft.com/en-us/library/ms996456.aspx. [Accessed: 05-Mar-2017].

[60] N. Naik and P. Jenkins, "An Analysis of Open Standard Identity Protocols in Cloud Computing Security Paradigm," *Proc. - 2016 IEEE 14th Int. Conf. Dependable, Auton. Secur. Comput. DASC 2016, 2016 IEEE 14th Int. Conf. Pervasive Intell. Comput. PICom 2016, 2016 IEEE 2nd Int. Conf. Big Data*, pp. 428–431, 2016.

[61] G. Kotonya and I. Sommerville, "Requirements Validation," in *Requirements engineering: processes and techniques*, 1st ed., Wiley Publishing, 1998, pp. 87–111.

[62] G. Kotonya and I. Sommerville, "Non-functional Requirements," in *Requirements engineering: processes and techniques*, 1st ed., Wiley Publishing, 1998, pp. 187–213.

[63] Microsoft, "Visual Studio IDE," 2017. [Online]. Available: https://www.visualstudio.com/vs/. [Accessed: 13-May-2017].

[64] J. Venable, J. Pries-Heje, and R. Baskerville, "FEDS: a Framework for Evaluation in Design Science Research," *Eur. J. Inf. Syst.*, vol. 25, no. 1, pp. 77–89, 2016.

[65] D. Roth, R. Anderson, and S. Luttin, "Introduction to ASP.NET Core," 2016. [Online]. Available: https://docs.microsoft.com/en-us/aspnet/core/. [Accessed: 13-May-2017].

[66] B. Allen and D. Baier, "Welcome to IdentityServer4," 2016. [Online]. Available:

http://docs.identityserver.io/en/release/index.html. [Accessed: 13-May-2017].

[67] N. Blumhardt, “Serilog,” 2016. [Online]. Available: https://serilog.net/. [Accessed: 13-May-2017].

[68] Microsoft, “SQL Server,” 2016. [Online]. Available: https://www.microsoft.com/en-us/sql-server/sql-server-editions-express. [Accessed: 13-May-2017].

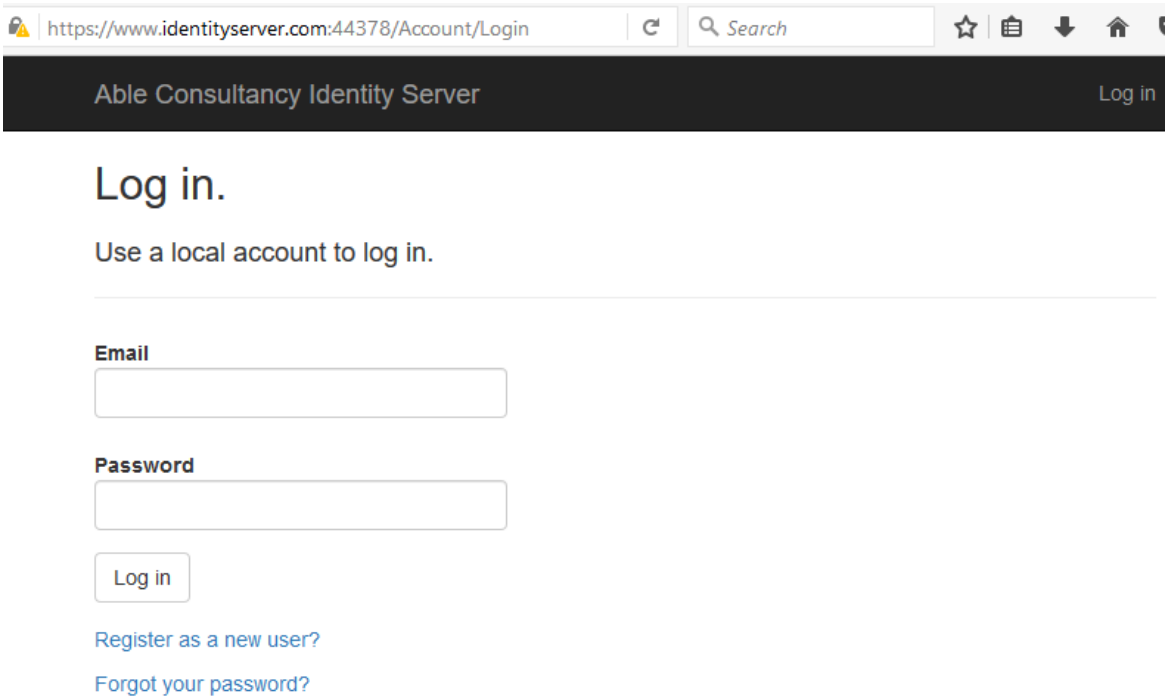
[69] Microsoft, “Entity Framework,” 2016. [Online]. Available: https://msdn.microsoft.com/en-us/library/aa937723(v=vs.113).aspx. [Accessed: 13-May-2017].

[70] Seq, “Seq,” 2016. [Online]. Available: https://getseq.net/. [Accessed: 13-May-2017].

13. Appendix A

13.1. Able Consultancy Identity Server

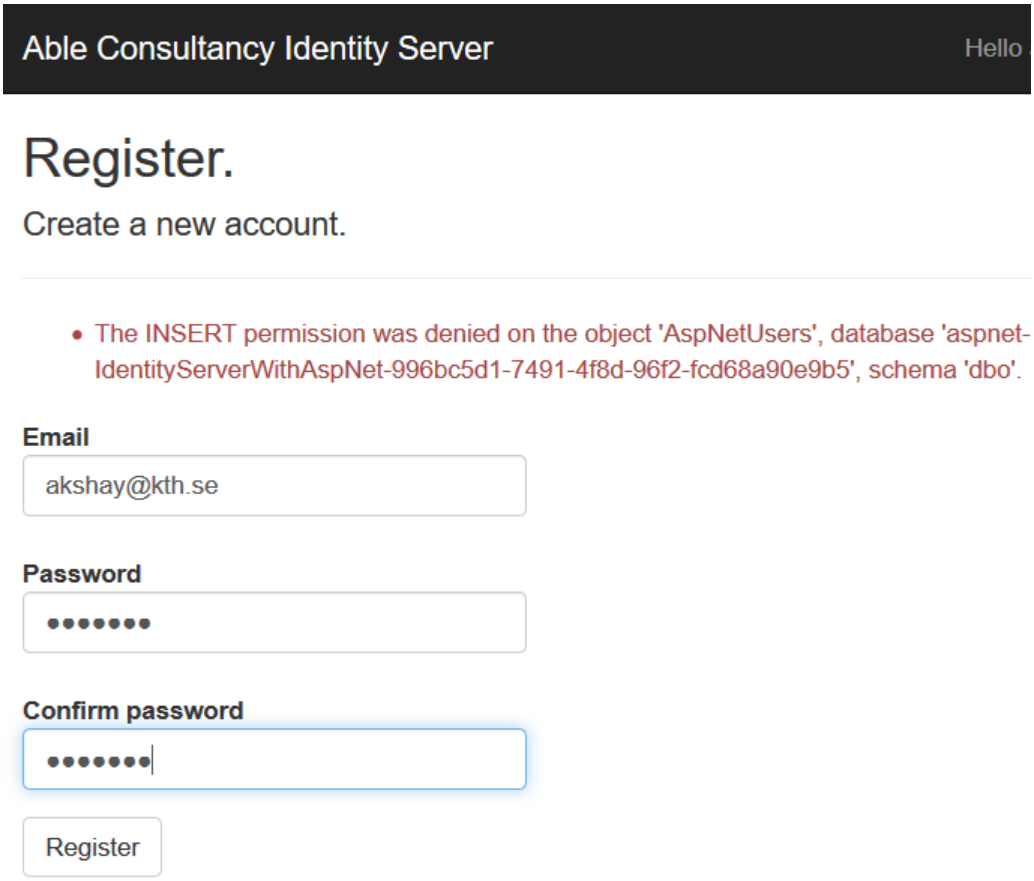
Login Page



The identity server provides a login screen to the users. The only functionality it performs is to allow users to log in and subsequently share user information with the collaborating service providers. Identity Server has read-only access to user information, and therefore no user data changes can be done from here. Identity Server is only available with HTTP over SSL and therefore all communication from the server to service providers are encrypted over the network.

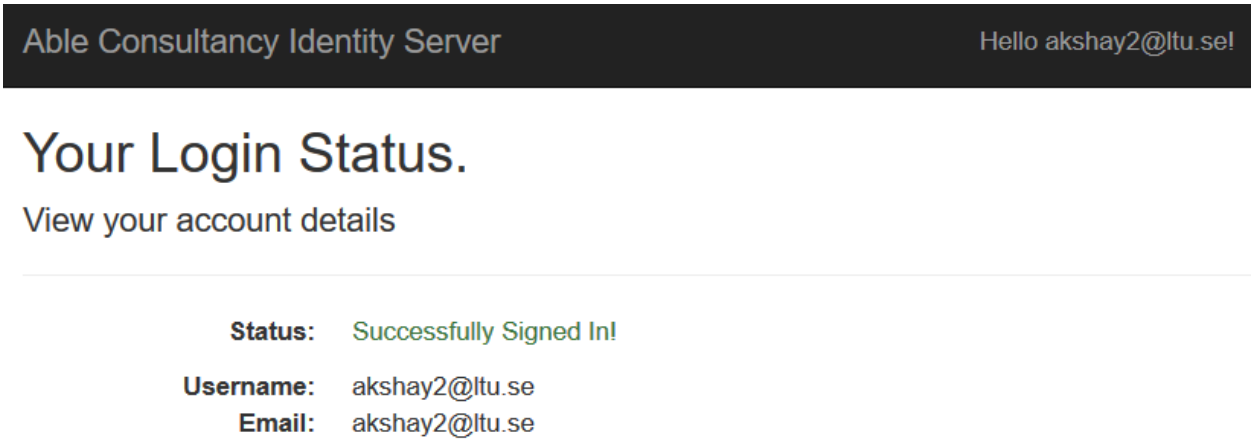
The two links – “Register as a new user?” and “Forgot your password?” are there to demonstrate that no user data changes are possible. An attempt to register new user fails as shown in the following screenshot

Register Page



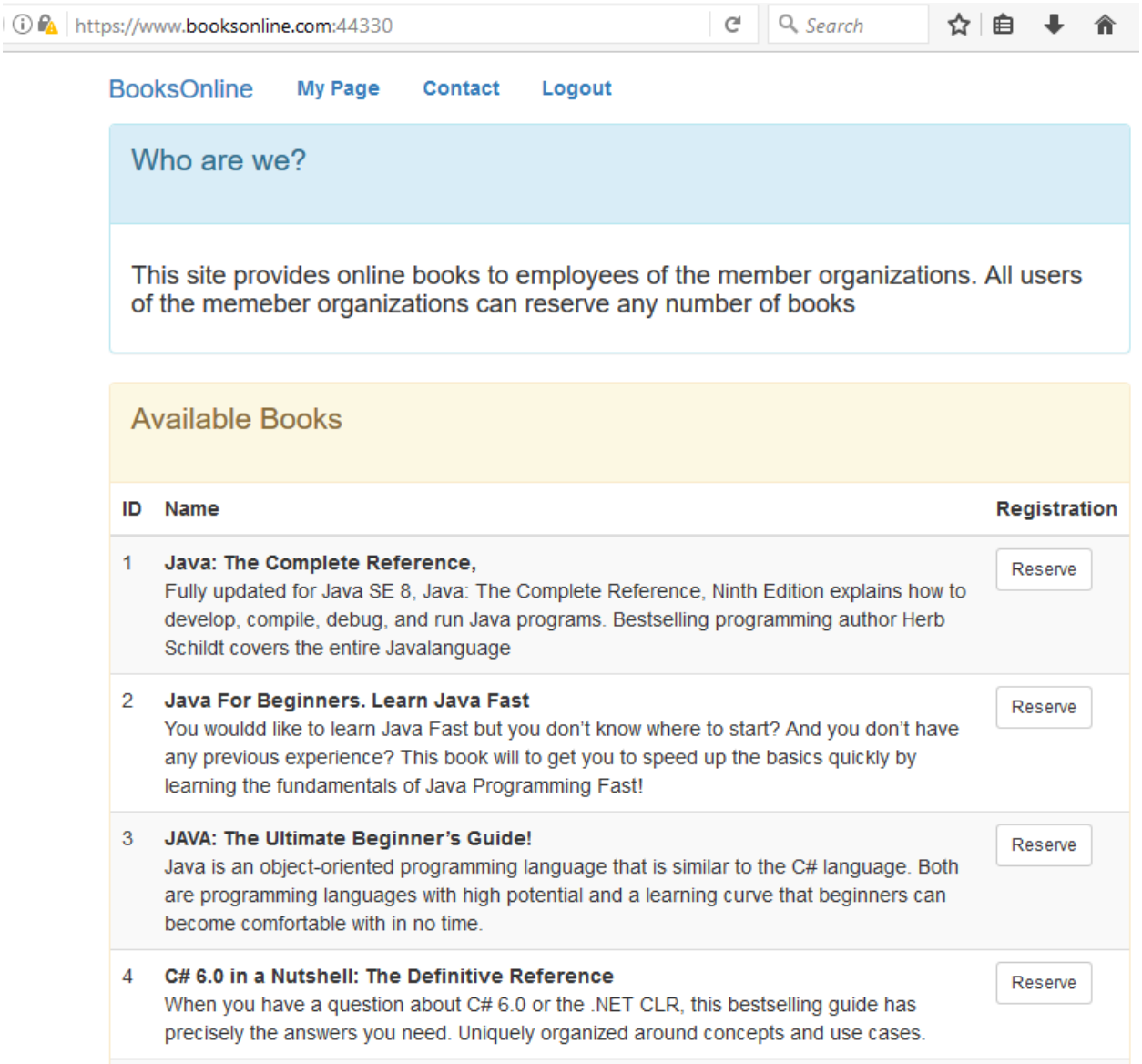
Status Page

Once the user the logged in, the user can check the login status and user information stored on the identity server



13.2. Books Online Website

Home Page



BooksOnline website is a service provider that provides access to e-books to its members. The service is only available with HTTP over SSL, and therefore all communications from the BooksOnline website are encrypted over the network. The navigation bar at the top contains four links –

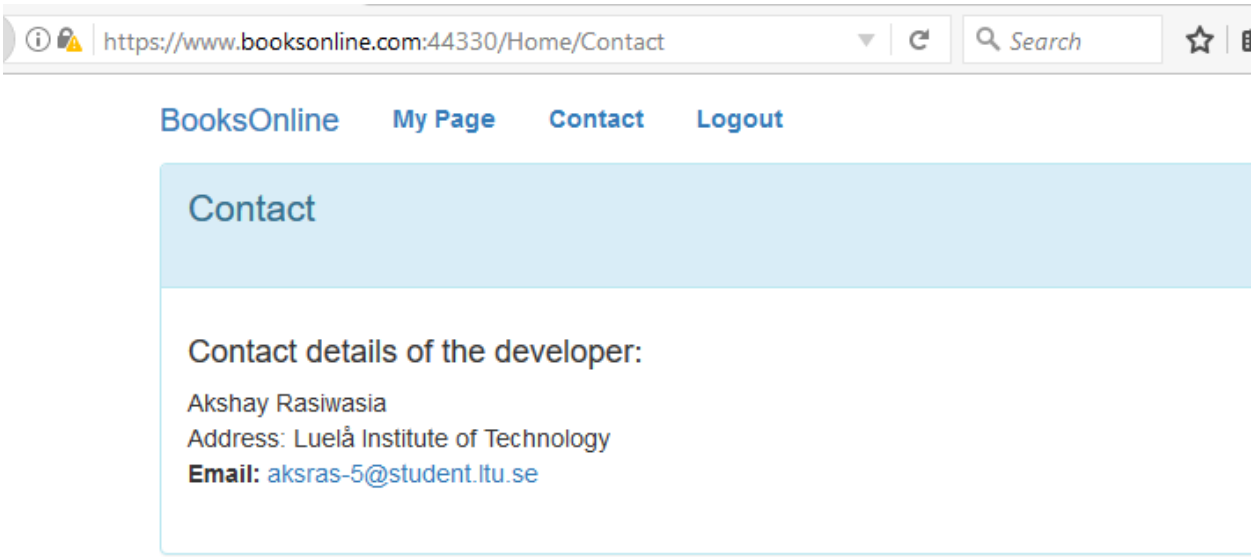
- BooksOnline: takes the user to the website's home page
- My Page: takes the user to its profile and reading list
- Contact: takes the user to the service provider's contact details
- Logout: Logs the user out

The home page of the website presents information about the service in the "Who are we?" panel. In the next panel "Available Books," the website lists the books available to read along with some description of

the book. On the right side of each book is the button to reserve the book so that the user can gain access and add it to their reading list.

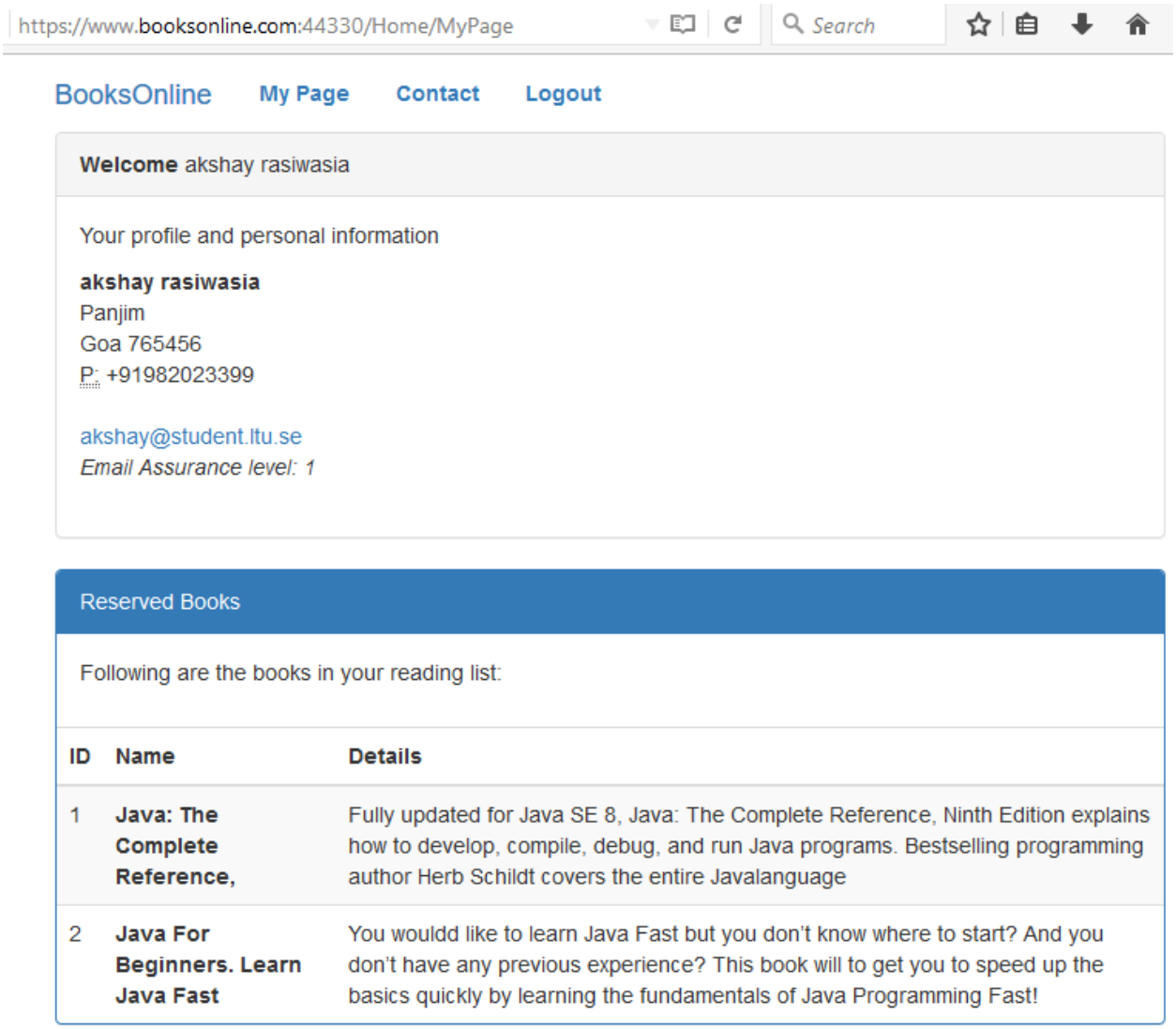
Contact Page

On click of the 'Contact' link, the user is taken to the 'Contact' page



My Page

On click of the 'My Page,' the user is taken to its personal profile and reading list. The user must be logged in to view the 'My Page.' The user information displayed is fetched from the identity server; none of the user's information is stored with the service provider. BooksOnline only has the information about the user's reading list, unique-id, and email address. For the purpose of demonstration, 'My Page' also displays the user claims and security tokens as received from the identity server.



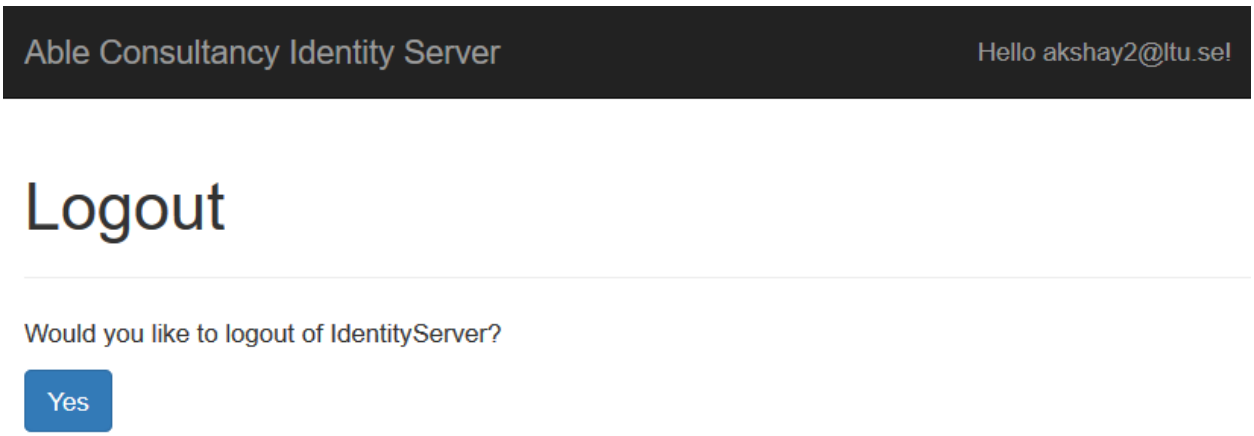
An example of user claims (information) received from the identity server is:

Claim Type	Claim Value
nbf	1495900652
exp	1495900952
iss	https://www.identityserver.com:44378
aud	BooksOnline
nonce	636314973807714184.MTc4NzQ5ZGltOGQyNy00YjkxLWE2OWEtNWUwYzk3ZTZkOTE3YWVmY
iat	1495900652
c_hash	wE1d4GL2PXN3llwQPh1_sQ
sid	beb935124f0f71aae20931b3c263c6a3
sub	94b05e05-9f60-4367-974b-2952811d65fd
auth_time	1495889067
idp	local
amr	pwd
name	akshay2@ltu.se
website	www.akshay.com
given_name	akshay
family_name	rasiwasia
preferred_username	aksras-5
email	akshay@student.ltu.se
email_verified	True
gender	male
birthdate	19830130
phone_number	+91982023399
phone_number_verified	False
email_assurance_level	1
address	{"locality": "Panjim", "region": "Goa", "postal_code": "765456"}

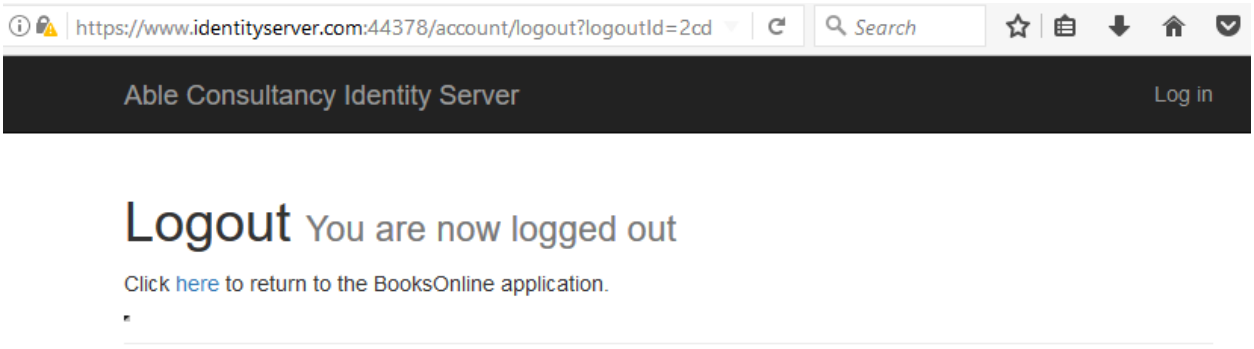
There are three security tokens involved to maintain the single-sign-on session. The access token is used to fetch user information from the identity server; refresh token is used to refresh the access token when it expires, and id token is the unique token assigned to the session between the identity server and the service provider. As long as the id token is valid, the service provider can use the access token to fetch information from the identity server or use the refresh token to renew the access token. Once the id token expires, the user has to provide consent again to the service provider. The various token expiry timeout is configurable. An example of the security tokens is the following screenshot:

Logout

On click of the 'Log out,' the user is redirected to the identity server to log out



Once the user clicks the 'Yes' button, the user is logged out of the identity server and then gets the option to return to the service provider's website.

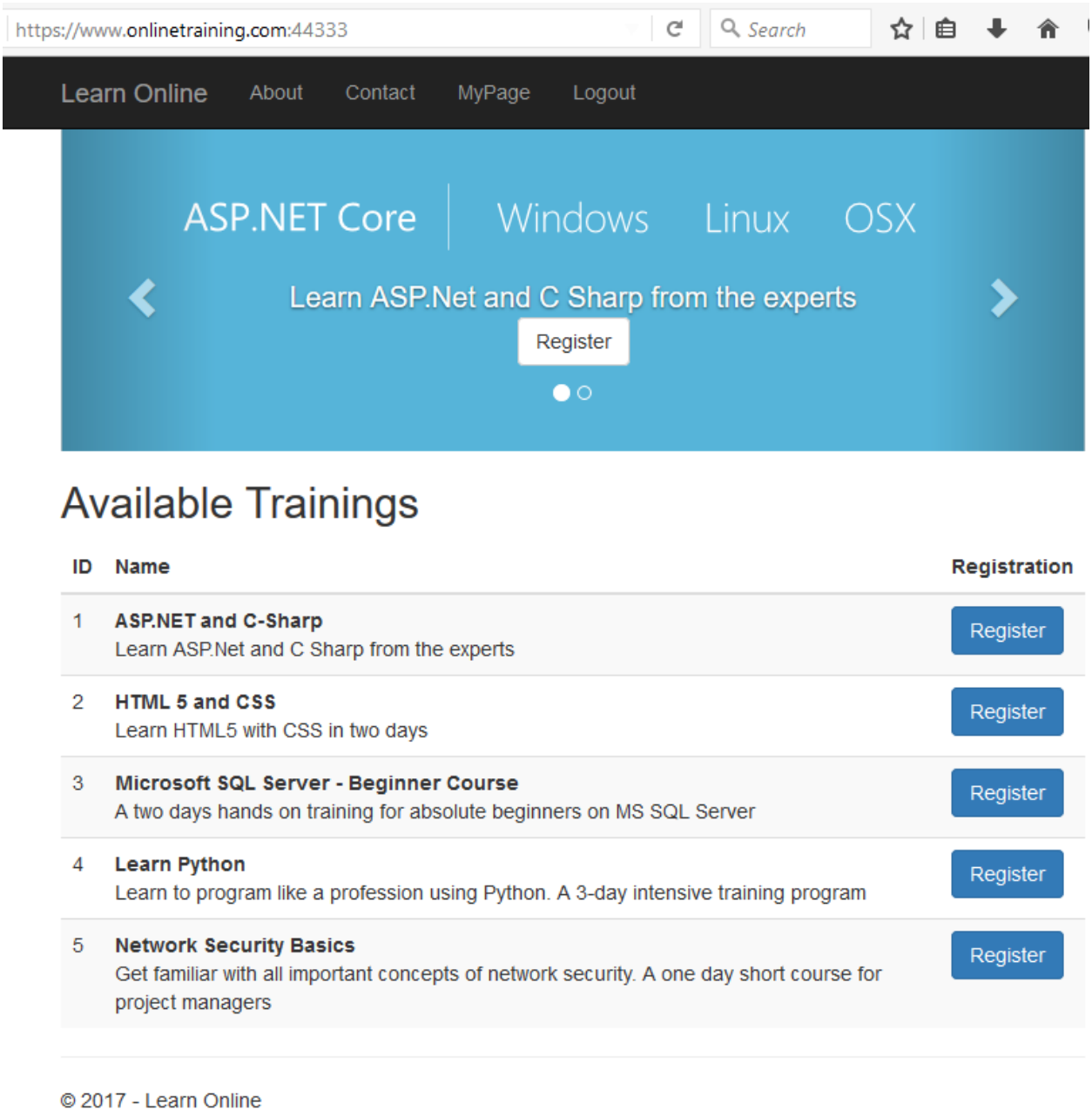


On clicking 'here,' the user is redirected to the BooksOnline home page.

13.3. Learn Online Website

'Learn Online' is another service provider that has collaboration with the Able Consultancy Identity Provider. All communication from the website happens only with HTTP over SSL, and therefore each message from the site is encrypted over the network.

Home Page



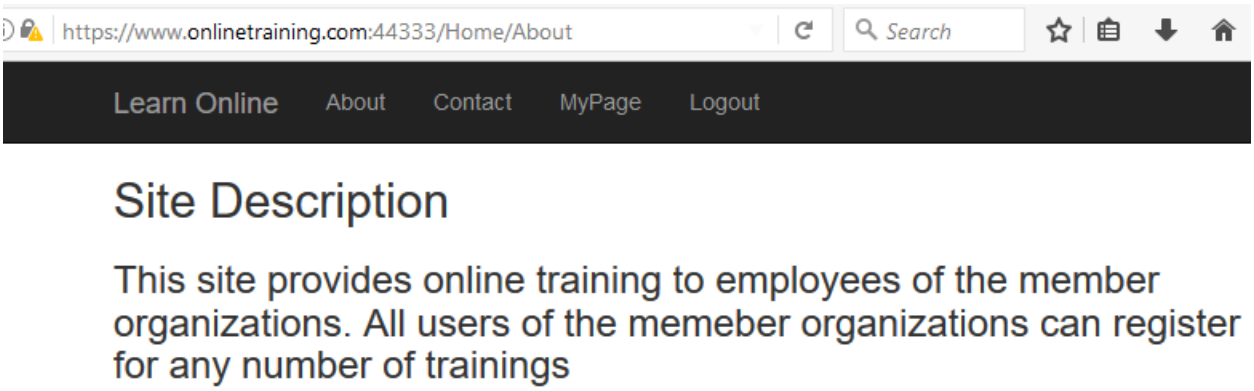
The home page of the website has a black navigation bar on top that has five links:

- Learn Online: takes the user to the website's home page
- About: Displays information about the services

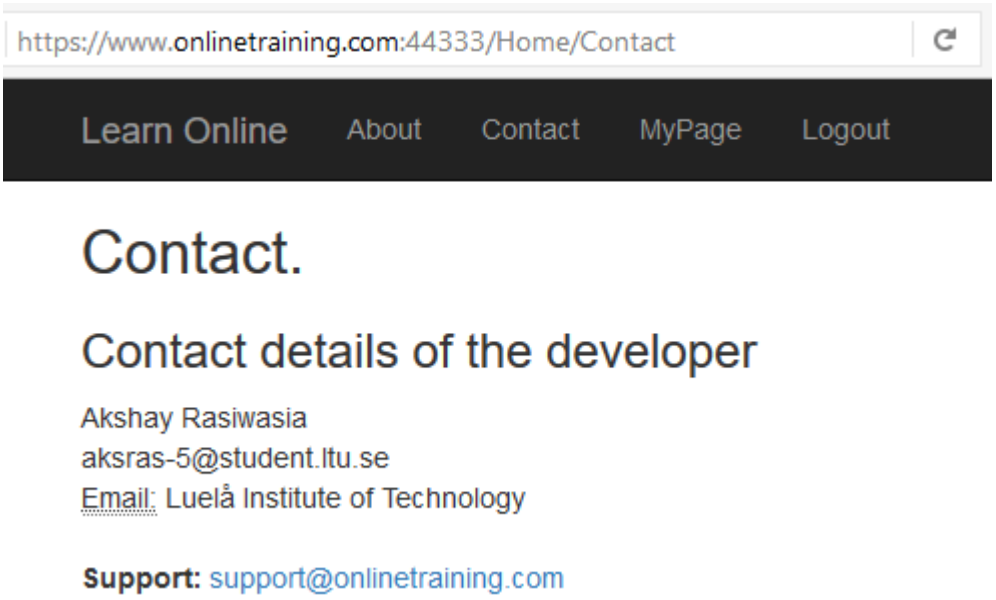
- Contact: takes the user to the service provider's contact details
- My Page: takes the user to its personal profile and registered training list
- Logout: Logs the user out

The home page also displays a list of available training that the logged in members can register to by clicking the 'Register' button on the right-hand side of each training description.

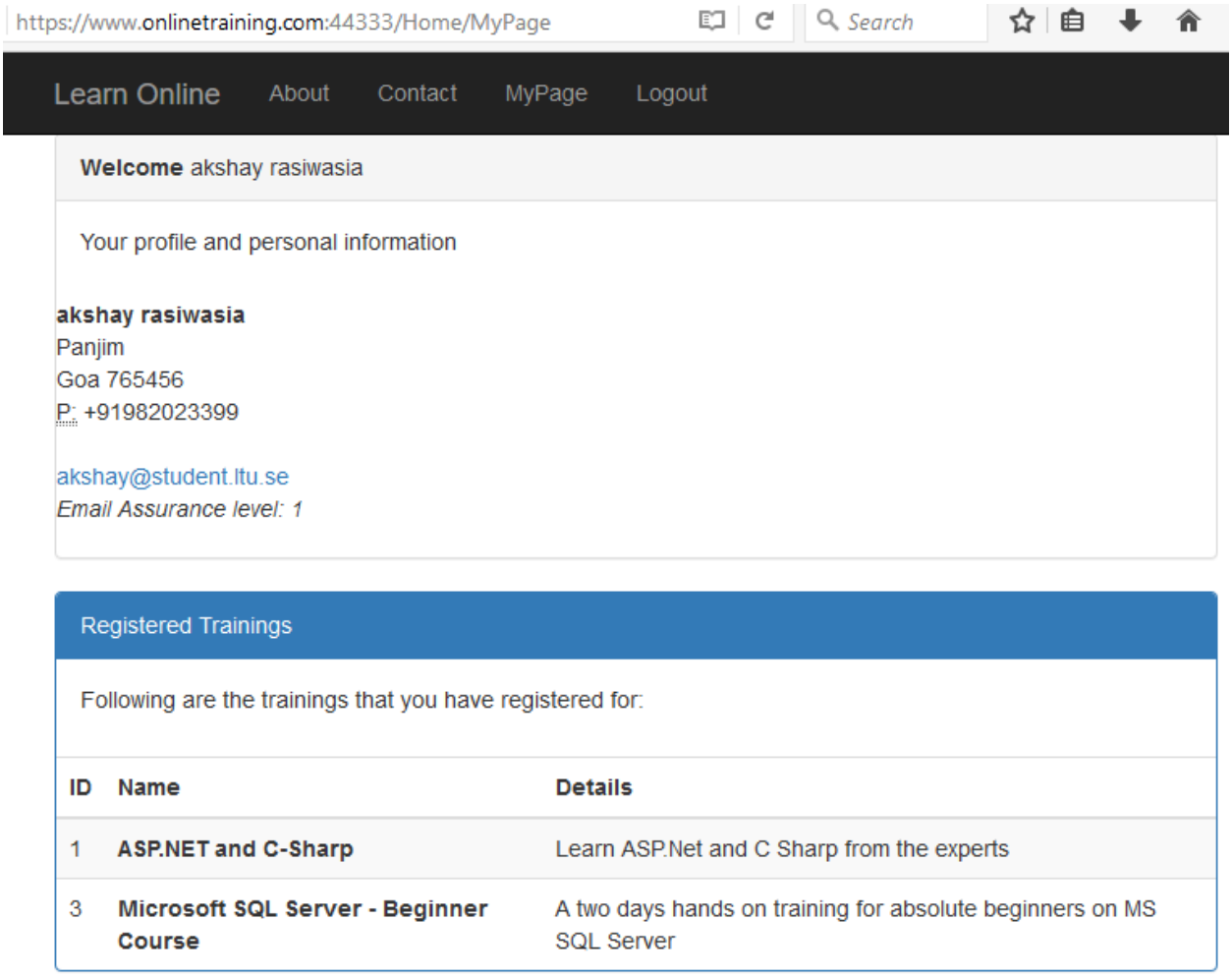
About Page



Contact Page

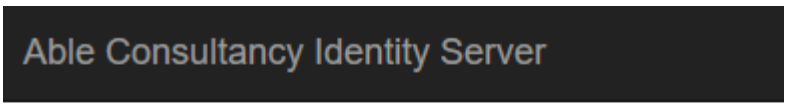


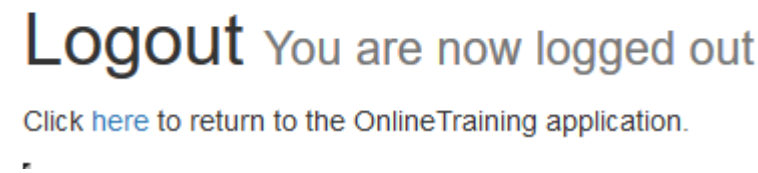
My Page



Logout Page

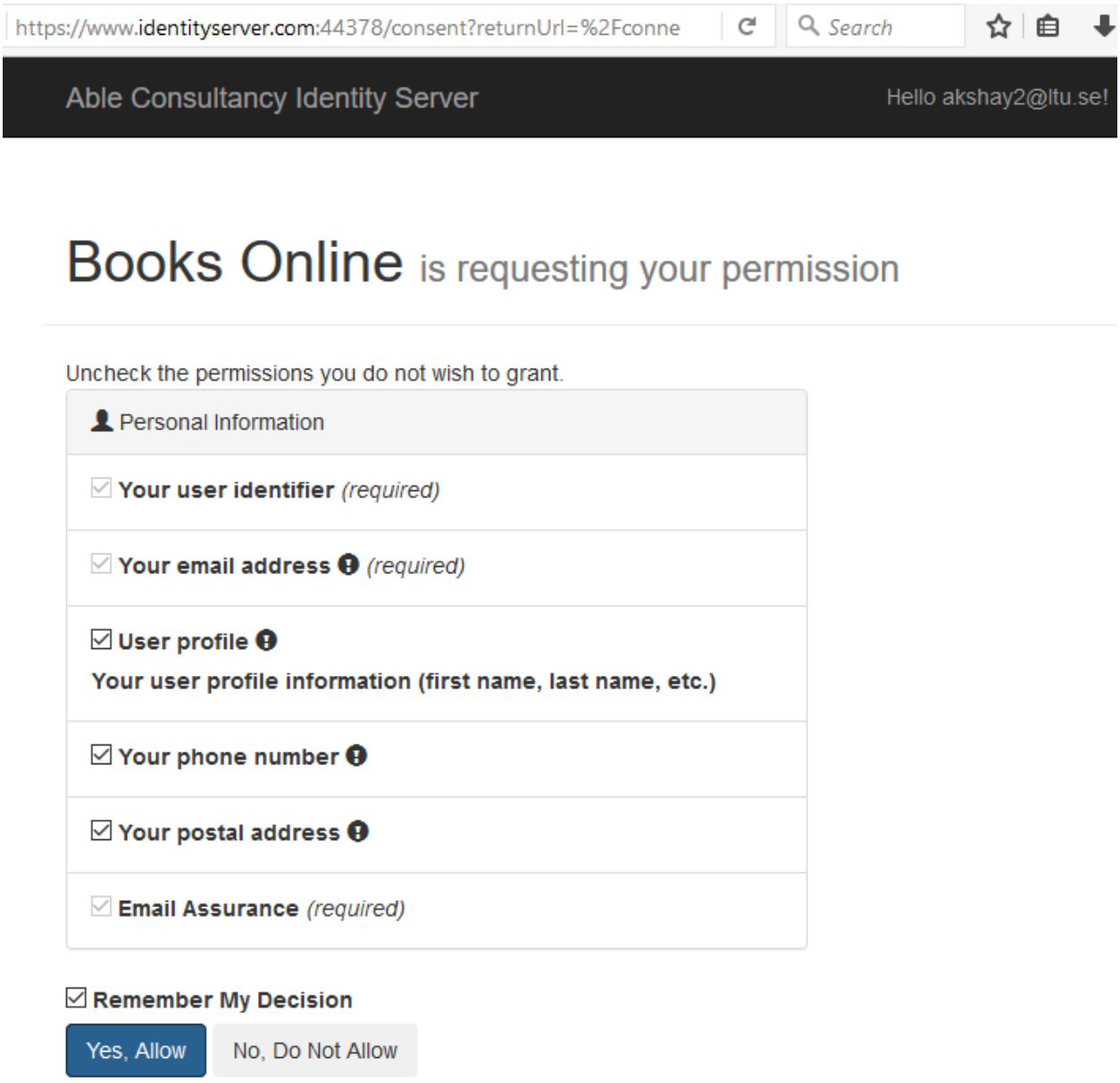
On click of 'Logout,' the user is redirected to the identity server to log out; and then given an option to navigate back to the 'Learn Online' website.



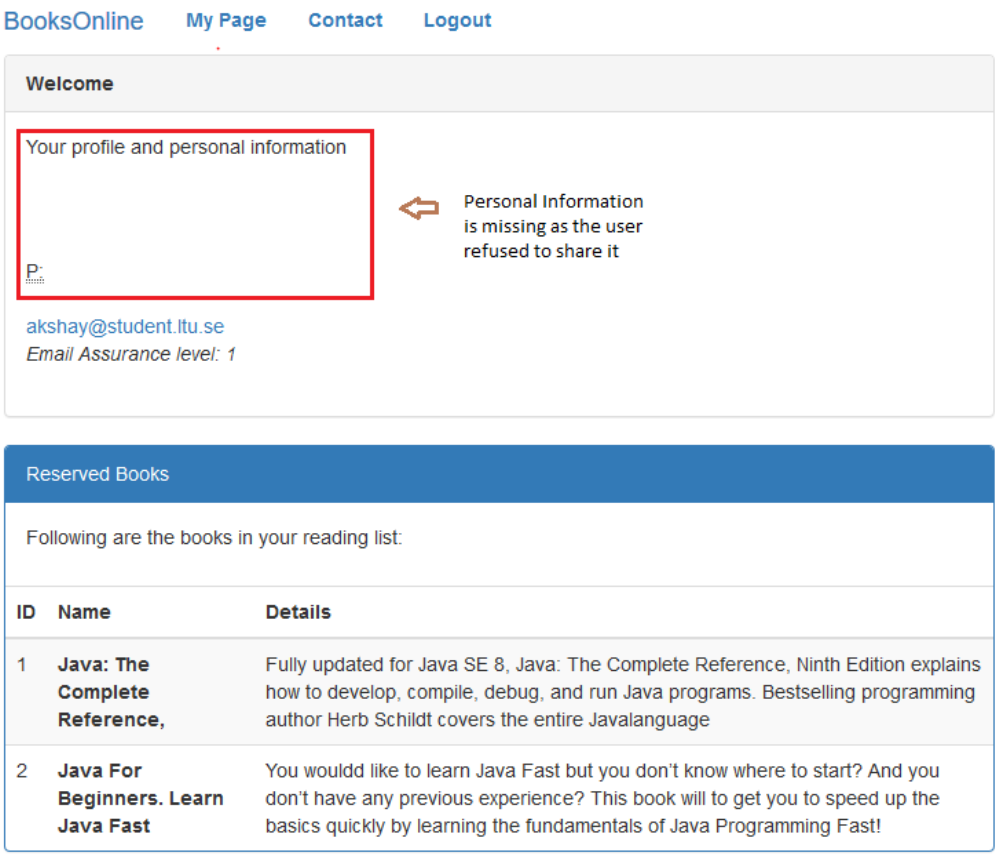


13.4. Consent Page

Consent page at the identity provider gives the users an option to see the information would be shared with the given service provider, the mandatory information that the service provider needs, the optional information, and to select which optional information the user wants to share.

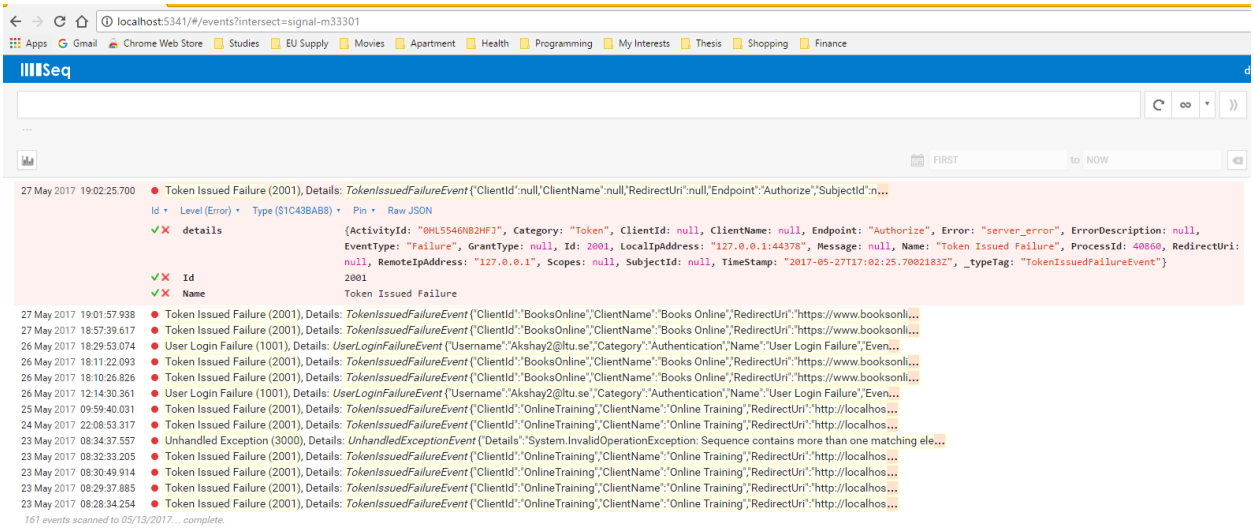


In this screenshot, the service provider 'BooksOnline' is requesting the permission to access user's personal information. The consent screen clearly indicates the mandatory and optional information. For example, if the user unchecks all the optional information i.e. postal address, phone number and personal profile and then gives permission to the service provider 'BooksOnline,' the service provider allows access, but certain personalization features would be missing at the 'BooksOnline' website.



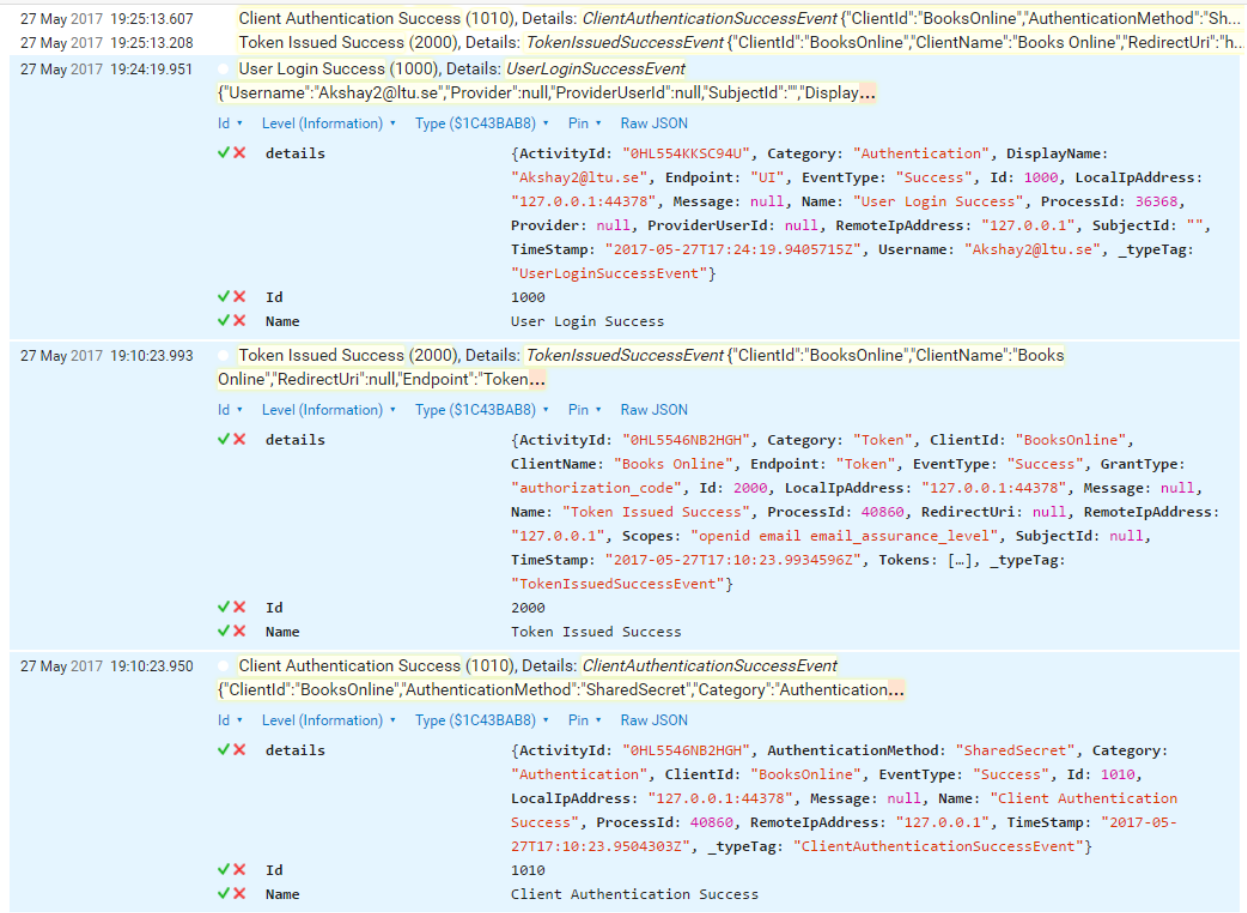
13.5. Logging

The identity server maintained detailed logs of all the important events in the SQL database. The error logs were read from the database and presented in a user-friendly way for diagnosis. An example of the error logs at the identity server



The details of one of the error

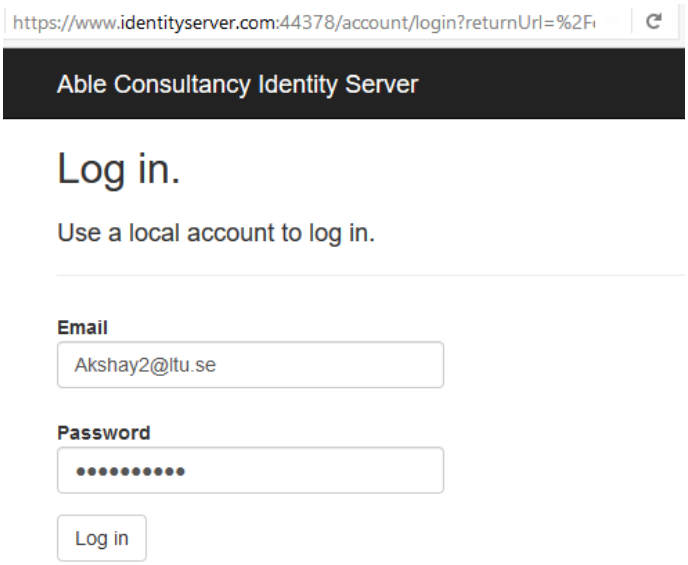
A screenshot of the log of important events such as client authentication success, token issue success, and user login success.



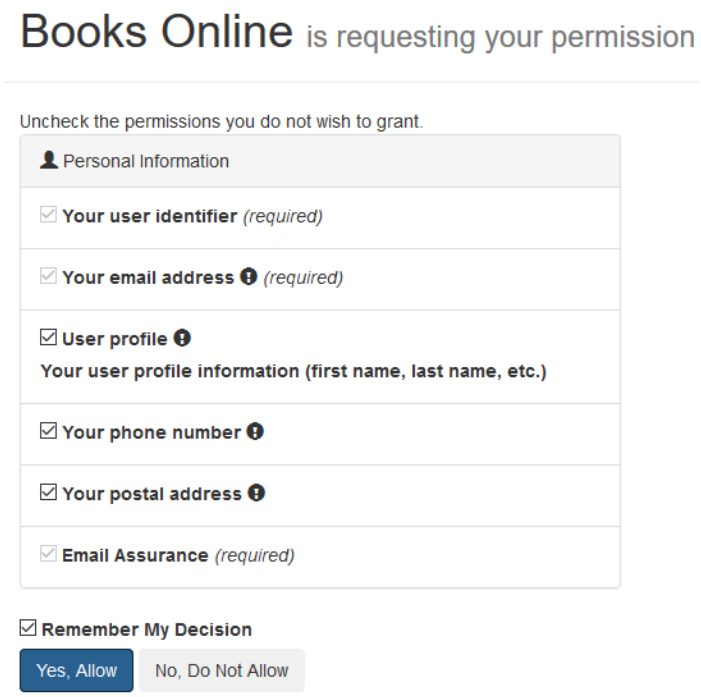
13.6. Single Sign On

This demonstration describes the steps that occur when the user tries to access a protected page on the 'BooksOnline' website.

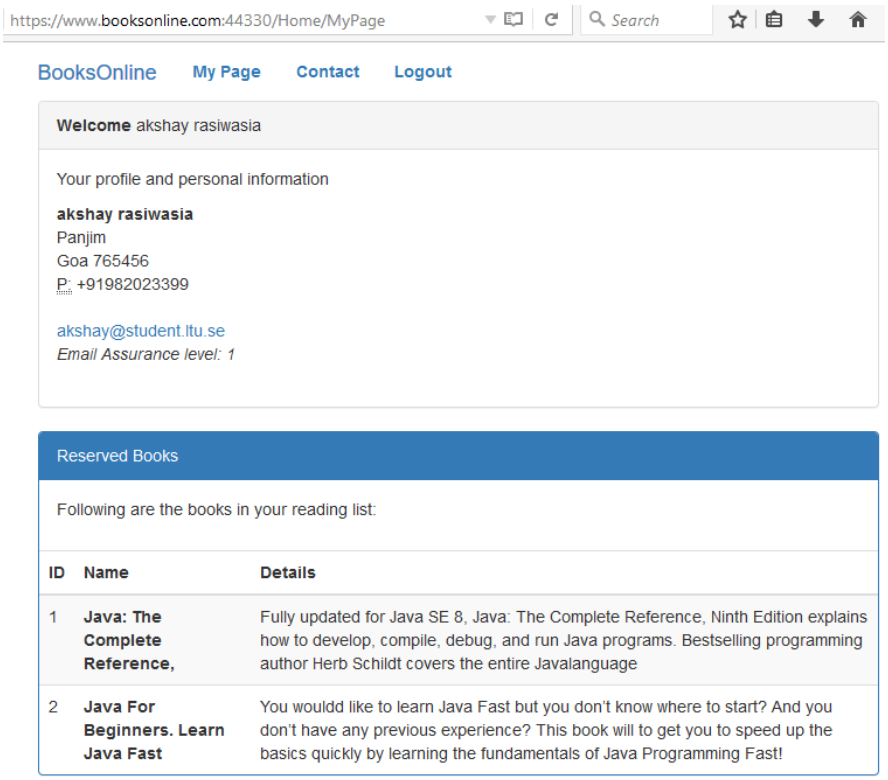
1. The user is redirected to the identity server to log in



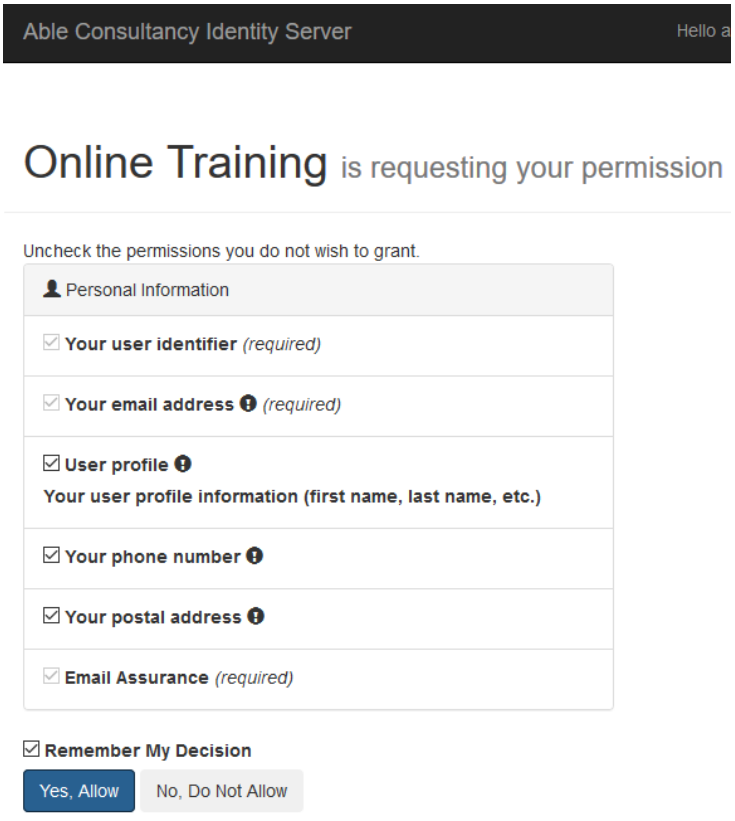
2. On successful login, the user is asked to give consent to sharing of user information with BooksOnline



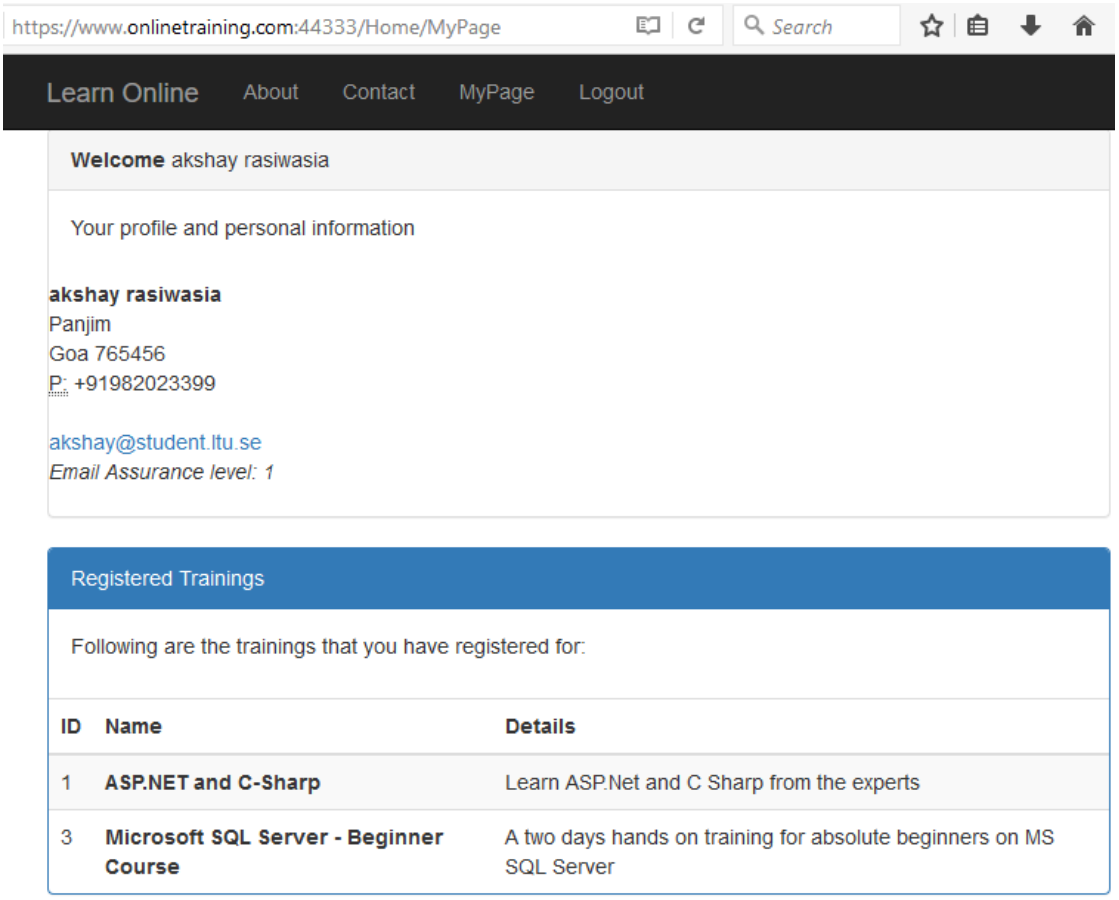
3. Once the user gives consent, the user is redirected back to BooksOnline website and can access his/her 'My Page.'



4. The same user now tries to access the protected 'My Page' at Learn Online website. This time the user is not asked to log in again but sent directly to the consent page to give approval to information sharing with Learn Online website



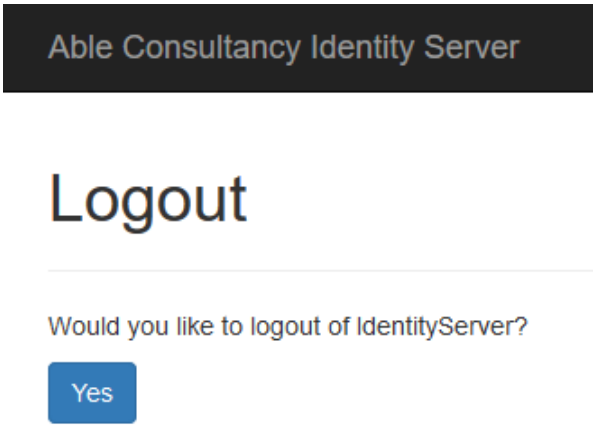
5. Once the user gives consent, the user is redirected back to Learn Online website and can access his/her 'My Page.'



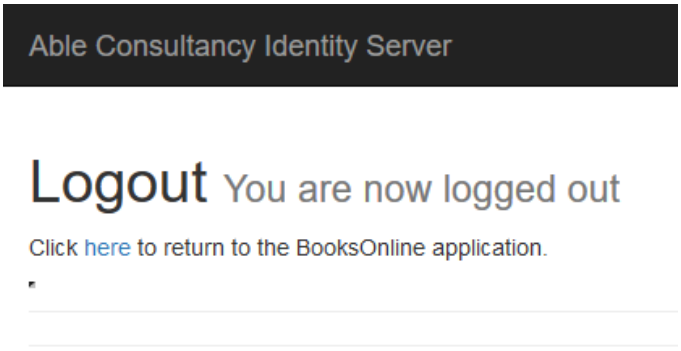
13.7. Single Log Out

This demonstration describes the steps that occur when user, the user is currently logged into the identity server and has access to both the service provider's protected resources i.e. user's profile page on both BooksOnline and Learn Online, but then decides to log out from the identity server and consequently is also logged out from both the service provider's website.

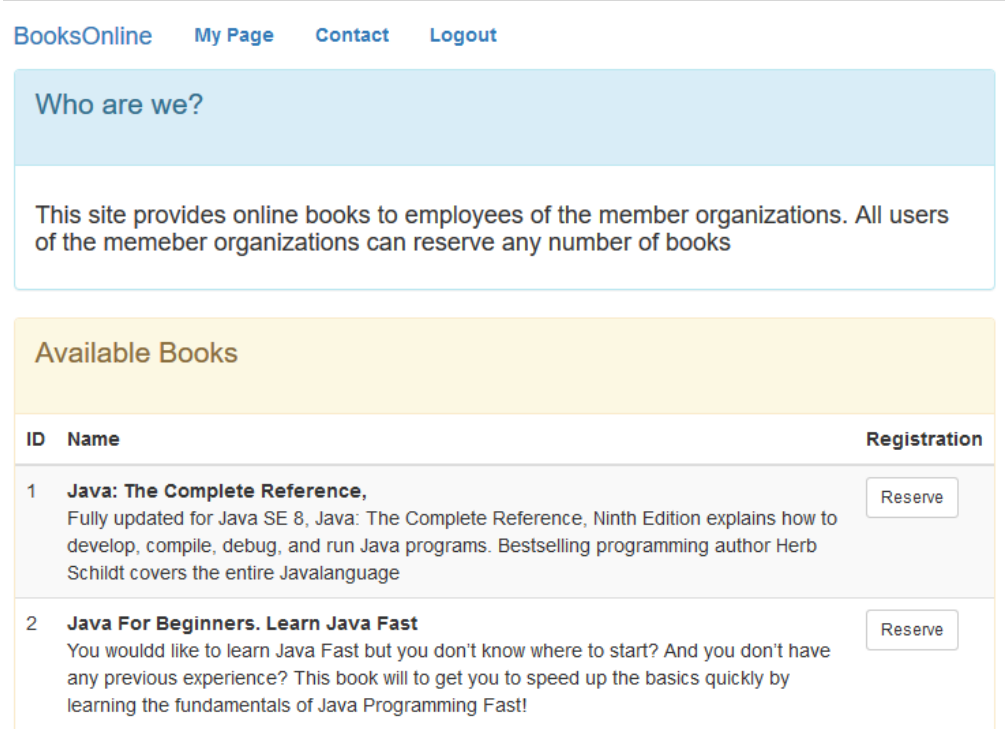
1. User clicks 'Log Out' on Books Online website. User is redirected to Identity Server to log out



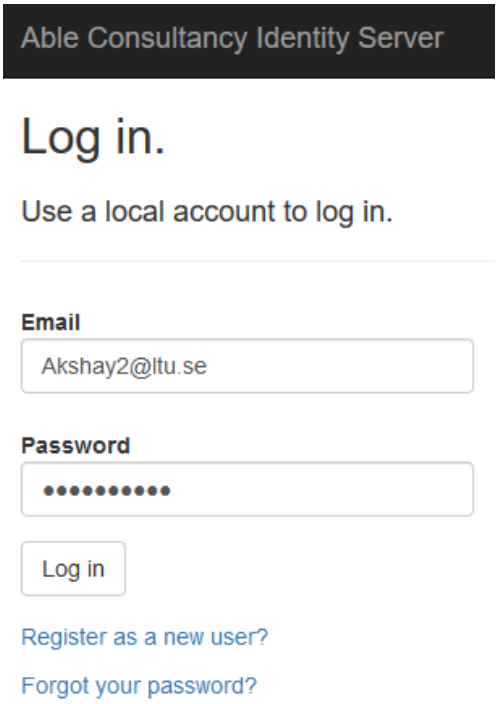
2. User confirms log out and hence logged out



3. User now re-navigates to the Books Online home page



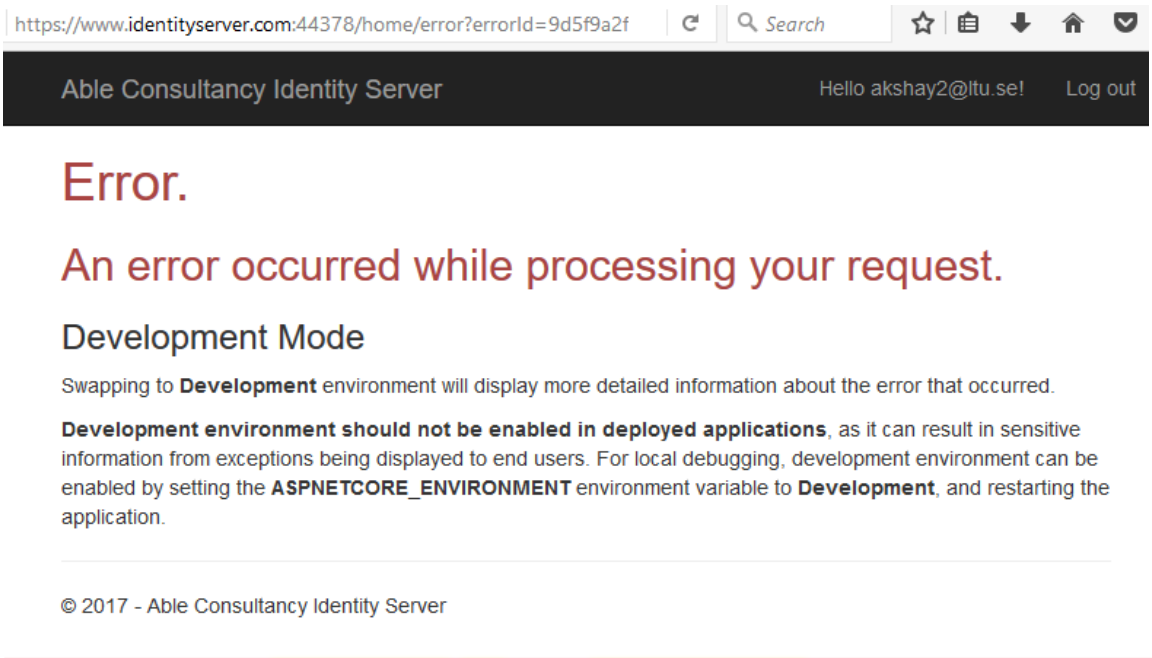
4. The user now tries to refresh the 'My Page' on the Learn Online website. The website denies access and redirects the user to log in again



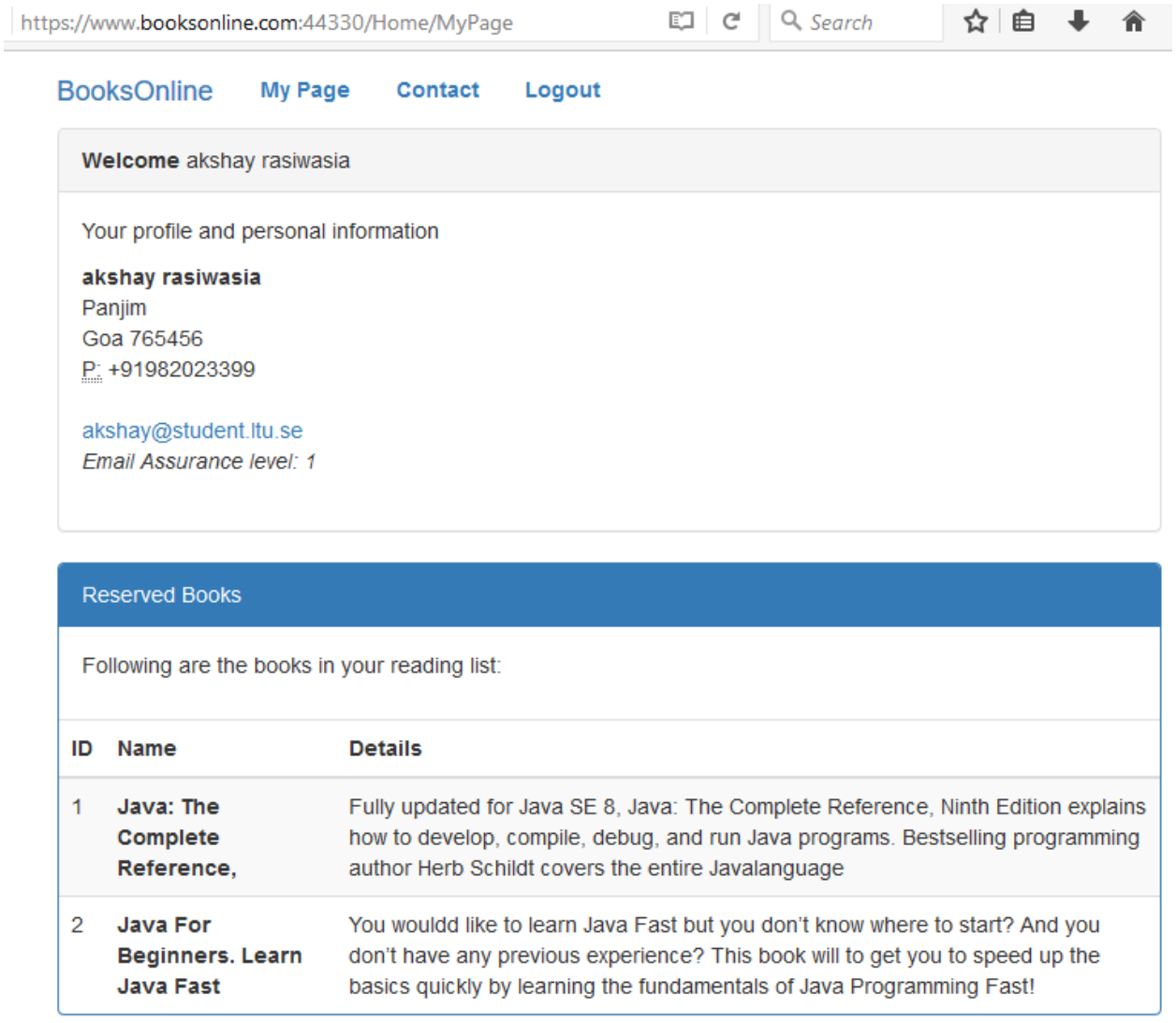
13.8. Disassociation of 'Learn Online' Service Provider

This demonstration shows how a service provider is removed from the federation.

1. Able Consultancy has terminated the agreement with the service provider Learn Online. The identity server removes the configuration associated with Learn Online. However, Learn Online has not removed the configuration.
2. A user from Able Consultancy tries to log in to Learn Online using the credentials from Able Consultancy identity server.
3. The identity server refuses the connection and an error is thrown.



4. The user now tries to access BooksOnline, and it is successfully able to do so because the relationship between Able Consultancy and BooksOnline is still intact.


https://www.booksonline.com:44330/Home/MyPage
Search
BooksOnline
My Page
Contact
Logout
Welcome akshay rasiwasia
Your profile and personal information
akshay rasiwasia
Panjim
Goa 765456
P: +91982023399
akshay@student.ltu.se
Email Assurance level: 1
Reserved Books
Following are the books in your reading list:
ID
Name
Details
1
Java: The Complete Reference,
Fully updated for Java SE 8, Java: The Complete Reference, Ninth Edition explains how to develop, compile, debug, and run Java programs. Bestselling programming author Herb Schildt covers the entire Javalanguage
2
Java For Beginners. Learn Java Fast
You wouldd like to learn Java Fast but you don't know where to start? And you don't have any previous experience? This book will to get you to speed up the basics quickly by learning the fundamentals of Java Programming Fast!

14. Appendix B

14.1. Feedback - Management Perspective

Framework Evaluation Form

Please fill in the feedback form after you have evaluated the implementation framework. Please note that all the fields are **mandatory**.

Evaluator Details	
Name	Anette Tavaststjerna
Email	anette.tavaststjerna@eu-supply.com
Date of Evauation	6/22/2017

Evaluation Criterion	Score	Comments
Usability - *the extent to which the framework is practically usable in real scenarios by the development team*	5- Excellent	Easy step-by-step list and scenarios. At EU-Supply we would be able to use it for coming SSO integrations, when we are using our technical platform
Efficiency - *the extent to which the framework would help to accomplish the SSO implementation in an optimal way without the loss of too much time and resources*	5- Excellent	Assuming some knowledge of the technical framework of SSO
Completeness - *the extent to which the framework covers the essential requirements for SSO implementation*	4- Very Good	Testing and verification part of the implementation is missing. Something more about maintaining the SSO once after it has been completed.
Reliability - *the confidence with which the framework can be trusted upon to achieve SSO implementation*	5- Excellent	
Testability - *the extent to which the guidelines given in the framework are testable to validate the implementation*	5- Excellent	
Comprehensibility - *the extent to which the framework is readable, intelligible without extra effort or ambiguity*	5- Excellent	A overview or schematic image to help understanding the access flow would be even more helpful clear to understand

14.2. Feedback – Technical Perspective

Framework Evaluation Form

Please fill in the feedback form after you have evaluated the implementation framework. Please note that all the fields are **mandatory**.

Evaluator Details	
Name	Abhiroop Gupta
Email	abhiroop.gupta@eu-supply.com
Date of Evauation	6/25/2017

Evaluation Criterion	**Score**	**Comments**
Usability - *the extent to which the framework is practically usable in real scenarios by the development team*	4- Very Good	Some very good guidelines are provided for taking into consideration while implementing a solution. However going a bit depper in technical analysis of minimum security requirements in establishing trust between RP and IdP such as on encryption/signing strategies and choices of algorithms, certificates types and key sizes would have been valuable to the dvelopment team.
Efficiency - *the extent to which the framework would help to accomplish the SSO implementation in an optimal way without the loss of too much time and resources*	4- Very Good	As described in the comment above the guidelines provides a understanding of the necessary requirements, however technical nitty gritties and challenges to be considered would have made the framework more efficient while implementating a solution.
Completeness - *the extent to which the framework covers the essential requirements for SSO implementation*	5- Excellent	All essential aspects of single sign on has been well covered in the framework. These guidelines are flexible and broadly applicable in most of the scenarios where a federated identity management is deemed.
Reliability - *the confidence with which the framework can be trusted upon to achieve SSO implementation*	4- Very Good	The described framework is quite reliable in achieving SSO implementation. Providing studies on existing IODC implementation and also guidelines on when and how much to use and integrate any existing solution such as IdentityServer2 to implement enterprise FIM would have increased the reliability.
Testability - *the extent to which the guidelines given in the framework are testable to validate the implementation*	5- Excellent	The scenarios of federated identity and single sign on are very testable in real scenarios sing the guidelines described in the work.
Comprehensibility - *the extent to which the framework is readable, intelligible without extra effort or ambiguity*	5- Excellent	The work described is very comprehensive and easy to understand.